

学校代码：10285
学 号：20204227053



硕士学位论文

(学术学位)



面向长文本的机器阅读理解研究

Reasearch on Long Text-oriented Machine Reading Comprehension

研究生姓名 董梦星
指导教师姓名 洪宇
专业名称 计算机科学与技术
研究方向 自然语言处理
所在院部 计算机科学与技术学院
论文提交日期 2023 年 4 月

苏州大学学位论文独创性声明

本人郑重声明：所提交的学位论文是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不含其他个人或集体已经发表或撰写过的研究成果，也不含为获得苏州大学或其它教育机构的学位证书而使用过的材料。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人承担本声明的法律责任。

论文作者签名：_____日期：_____

苏州大学学位论文使用授权声明

本人完全了解苏州大学关于收集、保存和使用学位论文的规定，即：学位论文著作权归属苏州大学。本学位论文电子文档的内容和纸质论文的内容相一致。苏州大学有权向国家图书馆、中国社科院文献信息情报中心、中国科学技术信息研究所（含万方数据电子出版社）、中国学术期刊（光盘版）电子杂志社送交本学位论文的复印件和电子文档，允许论文被查阅和借阅，可以采用影印、缩印或其他复制手段保存和汇编学位论文，可以将学位论文的全部或部分内容编入有关数据库进行检索。

涉密论文

本学位论文属 在_____年____月解密后适用本规定。

非涉密论文

论文作者签名: _____日期: _____

导师签名: _____日期: _____

面向长文本的机器阅读理解研究

摘要

本文旨在探讨面向长文本的机器阅读理解技术，即让计算机程序能够自动阅读长篇文本信息，并以与人类类似的方式理解长文本，随后回答相关问题的能力。这项技术是自然语言处理领域中一个重要的研究方向，并且已经在许多应用场景中发挥了重要作用，例如智能问答、文本摘要、机器翻译和知识图谱构建等。

随着互联网技术的迅速发展，人们面临的文本信息越来越多，包括新闻报道、学术论文等。为了更快速、更准确地获取需要的信息，长文本机器阅读理解技术应运而生，并且逐渐成为研究的热点。长文本阅读理解技术涉及多个子任务，如文本分段、关键信息提取、逻辑推理等。为了完成这些任务，需要采用多种自然语言处理技术，如滑动窗口、序列标注等。然而，长文本阅读理解也面临一些挑战。首先，长文本中存在大量跨句子和跨段落的逻辑关系，例如条件关系、递进关系等，这些关系需要进行跨句子和跨段落的推理才能正确理解文本意义。其次，阅读和理解长文本需要消耗大量的计算资源和时间，因此如何在保证精度的前提下提高效率也是一个难点。

首先，基于目前领域内最为重要的新闻长文本语料 NewsQA，本文提出了一种基于检索器和阅读器二阶段架构的方法。该方法基于预训练语言模型，分别实现问题与文本片段的相关性检索，以及进行序列标注，从而抽取答案文本片段。

第二，现有的长文本阅读理解架构通常只采用检索的方式来获取关键信息。本文另辟蹊径，通过在多跳阅读理解数据集 MuSiQe 上实验，提出了一种生成式的方法，将多跳问题转化为多个单跳子问题，并依次借助阅读理解模型，提取每个子问题的关键文本段落以及答案片段。

最后，针对长文本多项选择语料 QuALITY 中，文本与多个备选答案都存在关联的情形下，本文利用问题与备选答案指向的稠密检索器，检索出关键信息。同时，本文采用对比学习，以及改良过后的样本间自注意力机制，对备选答案进行更准确的语义表示，以解决备选答案之间的密切联系与区别。

本文的研究目的在于探讨长文本机器阅读理解的方法，通过提取关键信息、备选答案交互以及多跳问题分解等多个角度，本文提出了三种解决方案。通过在多个开源

数据集，如 NewsQA、QuALITY、MuSiQue 上进行实验，本文的方法取得了显著的评测指标提升，说明在长文本阅读理解方面拥有一定的实用价值。

关键词：机器阅读理解，长文本，稠密检索器，问题分解

作者：董梦星

指导老师：洪宇

Reasearch on Long Text-oriented Machine Reading Comprehension

Abstract

This study aims to explore machine reading comprehension technology for long texts, which enables computer programs to automatically read long text information and understand it in a way similar to humans, and then answer related questions. This technology is an important research direction in the field of natural language processing and has played an important role in many application scenarios, such as intelligent Q&A, text summarization, machine translation, and knowledge graph construction.

With the rapid development of Internet technology, people are facing more and more text information, including news reports and academic papers. In order to obtain the required information more quickly and accurately, long-text machine reading comprehension technology has emerged and gradually become a research hotspot. Long-text reading comprehension technology involves multiple sub-tasks such as text segmentation, key information extraction, and logical reasoning. To complete these tasks, various natural language processing technologies such as sliding windows and sequence labeling need to be used. However, long-text reading comprehension also faces some challenges. First of all, there are a large number of logical relationships across sentences and paragraphs in long texts, such as conditional relationships and progressive relationships. These relationships require cross-sentence and cross-paragraph reasoning to correctly understand the meaning of the text. Secondly, reading and understanding long texts requires a lot of computing resources and time consumption. Therefore, how to improve efficiency while ensuring accuracy is also a difficult point.

First of all, based on the most important news long-text corpus NewsQA in the current field, this article proposes a two-stage architecture based on retriever and reader. This method is based on pre-trained language models to realize relevance retrieval between questions and text fragments, as well as sequence labeling to extract

answer text fragments.

Second, existing long-text reading comprehension architectures usually only use retrieval methods to obtain key information. This article takes a different approach by experimenting with multi-hop reading comprehension dataset MuSiQUE to propose a generative method that converts multi-hop questions into multiple single-hop sub-questions and extracts key text paragraphs and answer fragments for each sub-question with the help of reading comprehension models in turn.

Finally, for the long-text multiple-choice corpus QuALITY where there is a correlation between the text and multiple candidate answers, this article uses a dense retriever pointed by questions and candidate answers to retrieve key information. At the same time, this article uses contrastive learning and an improved sample self-attention mechanism to more accurately represent the semantics of candidate answers in order to solve the close relationship and differences between candidate answers.

The purpose of this study is to explore methods for machine reading comprehension of long texts. Through extracting key information, interaction between candidate answers, and multi-hop question decomposition from multiple perspectives, this article proposes three solutions. By conducting experiments on multiple open-source datasets such as NewsQA, QuALITY, MuSiQue etc., this article's methods have achieved significant evaluation index improvements which indicates that it has certain practical value in long-text reading comprehension.

Key words: Machine Reading Comprehension, Long Text, Dense Passage Retrieval, Question Decomposition

Written by Mengxing Dong

Supervised by Yu Hong

目 录

第一章 绪论	1
1.1 研究背景和意义	1
1.2 国内外研究现状	2
1.2.1 机器阅读理解	2
1.2.2 长文本机器阅读理解	4
1.2.3 开放域问答	5
1.3 关键问题和研究难点	6
1.4 研究内容与组织结构	7
1.4.1 研究内容	7
1.4.2 组织结构	9
第二章 任务定义及评价方法	10
2.1 任务定义	10
2.2 语料资源	13
2.2.1 长文本数据集 NewsQA	13
2.2.2 多跳数据集 MuSiQue	14
2.2.3 多项选择数据集 QuALITY	15
2.3 性能评价指标	16
2.4 本章小结	19
第三章 基于检索器和阅读器架构的长文本阅读理解	20
3.1 引言	20
3.2 基于检索器和阅读器架构的长文本阅读理解	21
3.2.1 分段方法	21
3.2.2 基于预训练语言模型的段落检索器	22
3.2.3 融合方法	23
3.2.4 基于预训练语言模型的阅读理解模型	23
3.3 实验及结果分析	25
3.3.1 实验设置	25
3.3.2 实验结果和分析	26
3.4 本章小结	29

第四章 基于问题分解的多跳长文本阅读理解	31
4.1 引言	31
4.2 基于问题分解的多跳长文本阅读理解	33
4.2.1 总体架构	33
4.2.2 基于序列到序列生成的问题分解模块	34
4.2.3 基于预训练语言模型的阅读理解模型	35
4.2.4 采用分步执行的问题分解技术	36
4.3 实验及结果分析	37
4.3.1 实验设置	37
4.3.2 实验结果和分析	38
4.4 本章小结	40
第五章 基于对比学习的多项选择长文本阅读理解	42
5.1 引言	42
5.2 基于对比学习的多项选择长文本阅读理解	44
5.2.1 总体架构	44
5.2.2 基于 DPR 的检索器	45
5.2.3 样本内的自注意力机制	46
5.2.4 面向备选答案交互的对比学习方法	47
5.3 实验及结果分析	49
5.3.1 实验设置	49
5.3.2 实验结果和分析	51
5.3.3 消融实验	51
5.4 本章小结	54
第六章 总结与展望	55
6.1 工作总结	55
6.2 工作展望	56
参考文献	58
攻读学位期间的成果	66
致谢	67

第一章 绪论

本章旨在探讨长文本机器阅读理解的研究背景和意义，并对国内外研究现状的几个主要方面进行介绍，包括机器阅读理解、长文本机器阅读理解和开放域问答。在此基础上，本章分析了关键科学问题和研究难点。针对这些关键科学问题，本章详细介绍了本文的研究内容和组织结构。

1.1 研究背景和意义

机器阅读理解（Machine Reading Comprehension，简称 MRC）旨在教会机器在理解给定自然语言文本的基础上回答与之相关的问题，一直是自然语言处理（Natural Language Processing，简称 NLP）领域极具挑战性的前沿研究之一。

早期的机器阅读理解主要依赖于浅层的自然语言处理技术^[1-3]，如关键词提取^[4]、句法分析^[5]等，而这些技术忽略了句子之间的逻辑关系以及文本的语境信息。这些技术可以用于对文本进行初步的处理和分析。

随着深度学习技术的发展，机器阅读理解的研究逐渐转向基于神经网络的方法^[6]。利用深度学习技术，可以构建端到端的模型，从而更好地捕捉句子之间的关系和语境信息，提高长文本机器阅读理解的准确率。

同时，计算机可以帮助人们自动化地理解和处理大量的文本信息，从而节省时间和精力。这项技术也可以应用于很多场景，如自动化摘要^[7]、文本分类^[8]、信息检索^[9]、知识图谱^[10]等任务，也可以在教育、医疗、金融、智能客服^[11]等领域使用，使人们能够更加高效的获取和利用信息。

随着数字化信息的快速发展，人们需要从各种各样的长文本中获取信息。这包括但不限于新闻报道、学术论文、用户手册以及社交媒体。但是，长文本中通常涵盖了丰富的信息和细节，这增加了人类理解和提取信息的难度。因此，长文本机器阅读理解成为一项重要的研究任务，旨在通过计算机技术提高长文本处理的效率和准确性。

长文本机器阅读理解的研究与人工智能技术的发展密切相关。随着深度学习技术在自然语言处理领域的广泛应用和不断发展，机器阅读理解的研究也取得了显著的进展。深度学习技术为机器阅读理解任务提供了强有力的支持，使得研究人员可以通过大规模数据训练更加精准和高效的机器阅读理解模型。这些模型可以有效地帮

助人们处理长文本中的信息，从而提高信息处理的效率和准确性。

因此，长文本机器阅读理解的研究背景涉及到多个方面，包括人们对于高效阅读的需求、处理复杂语义的挑战以及深度学习技术的快速发展等。这些方面共同推动着长文本机器阅读理解的不断发展和进步，使得机器能够更好地理解和处理长文本中的信息。随着这些领域的不断发展和完善，长文本机器阅读理解技术将在未来得到更加广泛的应用，并且在各个领域中发挥着越来越重要的作用。

1.2 国内外研究现状

本节将从机器阅读理解、长文本机器阅读理解以及开放域问答三个研究视角出发，全面介绍国内外关于面向长文本机器阅读理解任务的相关研究工作。

1.2.1 机器阅读理解

机器阅读理解^[6]是自然语言处理领域的研究热点，近年来在国内外都得到了广泛的关注和研究。

随着各种深度学习技术的引入，机器阅读理解在近年来得到了越来越多的关注和发展。早在 2015 年，斯坦福大学推出了首个由自然语言问题构成的大规模机器阅读理解数据集 SQuAD^[12]；SQuAD 数据集的目标是让机器阅读理解模型能够回答自然语言问题，并从给定的篇章中正确地提取出答案。继 SQuAD 数据集之后，研究者们陆续提出了更多具有挑战性的大规模机器阅读理解数据集。为了提升模型回答的能力，SQuAD2.0^[13] 横空出世；相比于 SQuAD，增加了一些负样本，这些负样本包括了那些在篇章中无法找到答案的问题；因此，SQuAD2.0 不仅要求模型回答问题，还需要模型能够判断出问题是否有答案。NewsQA^[14] 也是一个类似于 SQuAD 的大规模机器阅读理解数据集，专注于新闻文章的阅读理解，其中包括来自 CNN 和 Daily Mail 等新闻网站的超过 10,000 篇新闻文章。此外，多项选择式机器阅读理解数据集 RACE^[15]，类似于中国学生参加的中考和高考英语，收录了来自高中和大学英语考试中的阅读理解题目；RACE 数据集中的问题包含多种类型，涵盖了不同难度级别，既有简单的词汇理解，也有复杂的推理和推断。基于多跳推理的大规模机器阅读理解数据集 HotpotQA^[16]，要求模型搜索多个文档的信息，回答一系列问题；由于涉及多个文档信息，HotpotQA 需要模型进行多文档联合推理。

与此同时，伴随着大规模机器阅读理解数据集的问世，基于神经网络的机器阅读

理解模型也相继涌现。基于神经网络的机器阅读理解模型大多采用嵌入层，编码层，交互层和输出层等四层架构。BiDAF^[17] 最早实现了双向注意力流机制，将问题与文本之间的语义关联性建模为一个注意力矩阵；并且通过引入字符级别的编码器，增强了模型对于语言的理解能力和语义建模能力。基于知识图谱的机器阅读理解模型 Reasonet^[18]，通过将知识图谱中的实体和关系引入到机器阅读理解模型中，来提高模型的推理能力和文本理解能力。基于卷积神经网络的 QANet^[19]，使用自注意力机制来学习文章和问题之间的交互表示，捕捉文章中与问题相关的信息，从而更有效的处理长文本和多文档场景。这些模型大多基于深度学习技术，通过对大规模数据的学习来提高阅读理解的能力。此外，基于多层 LSTM 的模型 Match-LSTM^[20]，可以同时对上下文和问题进行建模，并使用注意力机制来对匹配信息进行加权。在 Transformer 架构发布之后，研究人员陆续提出了各种预训练语言模型，例如 BERT^[21]、RoBERTa^[22]、DeBERTa^[23] 等，并在各种机器阅读理解任务上取得了成功。这些预训练语言模型极大地推进了自然语言处理领域的发展，它们的成功很大程度上归功于它们能够通过对大规模语料库的无监督预训练学习到文本的丰富表示。因此，本文提出的面向长文本的机器阅读理解方法均基于预训练语言模型进行研究与开发。

与此同时，国内研究者在机器阅读理解领域也取得了一定的进展。一些国内高校和科研机构也相继推出了机器阅读理解数据集和模型。清华大学自然语言处理与社会人文计算实验室提出的中文机器阅读理解数据集 CMRC^[3]，涵盖了新闻、百科、论坛等不同类型的文本，同时考虑了答案的多样性和篇章的连贯性。百度提出的中文机器阅读理解数据集 DuReader^[24]，囊括了答案抽取问题，以及选择题；该数据集在答案抽取和篇章连贯性方面的研究具有重要意义。华为也针对汉语语言提出了 CLUER-MRC^[25] 等四个相关的中文机器阅读理解数据集。

除了数据集的构建外，中文机器阅读理解方面的研究还包括了一系列的方法探索和技术创新。Wentong Chen 等人^[26] 提出了一种利用预训练的语言模型来提高中文机器阅读理解性能的方法，同时考虑了人类在理解文本时会关联的一些外部相关知识；该工作在 BERT 模型的基础上，引入了一个知识库模块，用于从给定的问题和上下文中抽取相关知识，并将其融合到语言模型中。Xue 等人^[27] 提出了一种基于多头注意力机制的机器阅读理解模型，能有效的捕捉中文文本和问题之间的关系。

1.2.2 长文本机器阅读理解

长文本机器阅读理解是在机器阅读理解的基础上，把输入文本换成了更长的形式。长文本阅读理解需要更强的对上下文和语境的理解能力，并且需要解决长文本中的信息丰富性、篇章结构复杂等挑战。如今，国内外研究者们都在积极探索这个问题，并取得了一些进展。

TriviaQA^[28] 是一个长文本机器阅读理解数据集，其问题和文本来自于维基百科、Web 网页等多个知识来源，文本长度有很大的变化范围，短至几百个单词，长可达几千甚至上万个单词；该数据集的发布推动了长文本机器阅读理解任务的研究，并成为该领域研究的重要基准数据集之一。NarrativeQA^[29] 是一个包含大量小说和电影剧本的数据集，用于自然语言处理和长文本机器阅读理解研究。

关于长文本的处理方法，滑动窗口^[30] 机制是一种常用的技术，它将长文本分割成多个较小的片段，以便于模型处理。Danqi Chen 等人^[31] 最早使用滑动窗口技术将长篇新闻文章分成多个段落，然后使用卷积神经网络模型进行阅读理解。随后，滑动窗口技术被广泛应用于长文本机器阅读理解任务中。例如，Shuohang Wang 等人^[20] 将长文本切分成多个窗口，并使用基于 LSTM 和注意力机制的模型来处理每个窗口，最终预测答案。Wang Wenhui 等人^[32] 也采用滑动窗口技术来处理长文本，然后使用自匹配注意力网络^[33]（Self-matching Networks）来捕捉文本之间的交互信息。除了这种方法，Facebook AI Research 在 2018 年提出了稠密向量检索^[34]（Dense Passage Retrieval，简称 DPR），它是一种从大规模文本集合中检索出最相关的文本段落的检索技术；DPR 利用双塔结构将文本表示为向量，并通过检索候选文本库的方式来处理长文本机器阅读理解问题。此后，DPR 被广泛应用于 TriviaQA，Natural Questions 等多个数据集，并取得了很好的效果。此外，Transformer-XL^[35] 在基于 Transformer^[36] 语言模型的基础上，引入了递归机制和相对位置编码来解决长文本的问题；递归机制允许模型在处理长序列时保留之前的状态信息；相对位置编码技术允许模型在处理长文本时捕捉到相对位置的信息。国内研究者们也在长文本机器阅读理解领域进行了很多探索和实践。清华大学和阿里巴巴联合提出了一种基于认知理论的框架 CogLTX^[37]，可以将 BERT 等预训练语言模型应用到长文本上；CogLTX 通过训练一个判断模型来识别长文本中的关键句子，并将其串接进行推理，并通过排练和衰减实现多步骤推理。

1.2.3 开放域问答

开放域问答^[38]（Open Domain Question Answering）是一种自然语言处理任务，旨在从大规模的自由文本中回答人类提出的自然语言问题。它通常包含信息检索和阅读理解两个子任务。由于其自由文本的语言表达方式和信息组织方式多样，开放域问答技术通常要求模型具有对多样化文本的理解、推理和组织能力。开放域问答技术通常可以用于搜索引擎当中，如今很多国内外学者都在研究这一技术的应用与创新。

Natural Questions^[39]（NQ）是由 Google 推出的一个开放域问答数据集，它是一个基于真实用户提问和真实网页文本的数据集；NQ 信息检索任务要求系统仅通过问题，从真实网页中选择一个最相关的文档。SearchQA^[40] 是一个基于搜索引擎的开放域问答数据集，其中由超过 140,000 个人工制作的问题来自于人们在真实的搜索引擎中提出的查询；该数据集的特点是问题的多样性和复杂性，其中包括一些需要深入理解文化、历史和常识的问题。

Dheeraj Rajagopal 等人^[41] 提出了一种 Pipeline 的方法，来解决开放域问答任务中的挑战。具体来说，该方法首先使用 TF-IDF^[42] 等技术对文本进行编码，然后使用知识库中的实体和属性对编码进行扩展，以便更好地捕获问题和答案之间的语义关系；接下来，该方法使用文本和知识库中的信息进行匹配和排序，并使用基于规则和基于机器学习的技术来生成答案。另一种称为“两阶段方法”的流程是，首先使用检索器对大量给定的文档进行筛选，从中提取与问题相关的部分文本片段，然后从这些文本片段中抽取出答案。Danqi Chen 等人^[43] 提出了一种新的方法，使用预训练语言模型来代替传统的阅读理解模型，对 Wikipedia 的文章进行阅读和理解，从中抽取问题相关的段落和答案。部分科研工作采用了端到端学习的方式，将检索器和阅读器模型同时进行训练，从而实现更加紧密的模型集成和协同工作。其中一个典型例子是 ORQA^[44]，该方法通过联合训练检索器和阅读器，实现了对候选文档的精细筛选和对答案的精准抽取，从而提高了整个 Pipeline 的效率和准确性。也有一些大模型，甚至可以不进行微调，而直接预测答案。Adam Roberts 等人^[45] 探讨了神经语言模型在无结构文本上训练时能够隐式地存储和检索知识的能力，并提出了一种评估方法来测量模型参数中包含的知识量。

1.3 关键问题和研究难点

机器阅读理解可以应用于多种自然语言处理任务，其具有重要的理论意义和广阔的应用前景。如今，以预训练语言模型为主的神经机器阅读理解技术得到了快速发展，但机器阅读理解模型在长文本问答场景下仍有巨大的提升空间。

长文本机器阅读理解的关键问题和研究难点主要包括以下几个方面：

(1) 问题理解

多文档机器阅读理解^[46] 是长文本机器阅读理解的一种形式，其与传统的阅读理解任务相比，需要更深入的理解和推理。该任务中问题的多跳形式使得问题的语法结构和包含的内容信息非常复杂，这使得神经网络模型在理解问题时面临许多挑战。例如，模型需要理解问题中的关键信息和上下文，并能够识别问题中不同子问题之间的联系和顺序，以便逐步构建问题的完整解答。问题理解的难点在于需要将自然语言转换为计算机可以理解的形式，同时还需要在不丢失关键信息的前提下对问题进行简化，以提高模型的理解能力。

为了解决这个问题，研究者们提出了各种不同的方法，包括基于图神经网络^[47] 的方法、基于知识库^[48] 的方法、基于语义^[49] 的方法和基于转换^[50] 的方法等。这些方法都旨在提高模型对问题理解的能力，以实现更加准确和智能的多文档机器阅读理解。

(2) 文档选择

在长文本机器阅读理解任务中，有时需要从大量的文档或段落中选择与问题相关的文档进行阅读理解。如果直接将所有文档都送入神经网络模型进行处理，会导致计算代价过大，同时降低模型的准确性和鲁棒性。因此，文档选择成为了长文本机器阅读理解任务中的一个关键问题，其目标是从候选文档集中快速且准确地筛选出与问题相关的文档，以减少计算代价和提高精度。

文档选择的难点在于如何有效地区分候选文档中与问题相关的内容和无关的内容。这需要模型能够理解问题的意义和背景，并能够识别出文档中与问题相关的段落和句子。现有的文档选择方法包括基于特征^[51] 的方法、基于深度学习^[52] 的方法和基于知识库^[48] 的方法等。这些方法尝试从不同的角度对文档进行建模，以便更好地捕捉与问题相关的信息。此外，研究者们还尝试了多阶段的文档选择^[53] 方法，通过逐步缩小候选文档集的规模来提高文档选择的准确性和效率。

(3) 答案抽取

机器阅读理解中的答案抽取是指如何从选出的相关段落中准确地抽取出符合问题要求的答案，以满足用户需求和提高质量。这个过程不仅涉及到答案的位置定位，还需要对答案进行语义理解和答案类型判断等^[54]，以满足用户需求并提高阅读理解质量。

然而，答案抽取过程中存在一些难点。首先，答案往往需要从复杂的文本结构中抽取，涉及到不同语言学层面的分析和理解，如词汇、句法和语义等^[55-57]。这需要模型具备深厚的自然语言处理能力，能够理解文本中的逻辑关系和上下文信息，同时能够处理诸如命名实体识别和语义角色标注等任务^[58,59]。其次，文本中可能存在歧义和多义性，同一问题可能有多个可能的答案，因此模型需要具备有效的答案排序和评估机制^[60]。在答案抽取的过程中，模型还需要考虑多个相关段落中的答案，并进行合并和消歧。最后，还需要考虑多种类型的问题，如开放式问题和封闭式问题等。不同类型的问题对于答案抽取的难度和挑战也不同，因此需要针对不同类型的问题选择合适的答案抽取方法^[61]。

总之，长文本机器阅读理解是自然语言处理领域的一个重要研究方向，其涉及到多个复杂的子任务，如问题理解、文档选择、答案抽取等，每个子任务都存在着一些独特的难点和挑战。解决这些难点不仅可以提高机器阅读理解的准确度和效率，还可以促进自然语言处理领域的发展。

1.4 研究内容与组织结构

针对以上总结出的关键问题与研究难点，本节首先引出本文的主要研究内容，随后给出文章的组织结构。

1.4.1 研究内容

本文分别针对段落选择、多跳问题分解、答案选项融合等三个关键问题进行研究，对阅读理解模型进行改进。针对上述关键问题，本文提出了一种面向长文本的机器阅读理解研究方法，其框架图如图 1-1 所示，具体分为以下三个方面：

(1) 基于检索器和阅读器段架构的长文本阅读理解研究

在长文本阅读理解任务中，传统的滑动窗口机制限制于每次只能处理 512 个词，

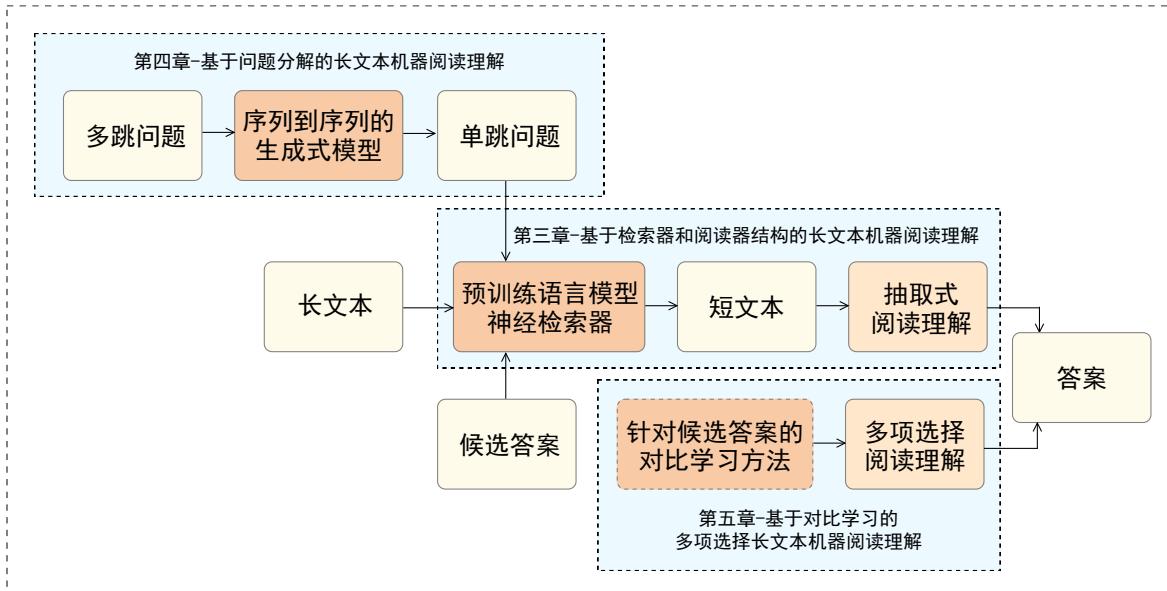


图 1-1 面向长文本的机器阅读理解研究框架图

难以建立长距离依赖，并且在截断部分时会丢失信息。因此，本文提出了一种基于二阶段架构的解决方案，该方案首先选出证据，然后抽取答案片段。在第一阶段中，检索器为每个文本片段打分，依据可回答性标签来对文本片段进行排序。第二阶段中的阅读器从高置信文本片段堆中抽取答案。这种方式可以在提取精炼信息的前提下进行答案抽取，缩小答案搜索范围，提高答案抽取的准确性。

(2) 基于问题分解的多跳长文本阅读理解研究

在多跳阅读理解任务中，由于问题过于复杂，且传统的机器阅读理解模型缺乏多跳推理能力，因此本研究提出了两种解决方案来将多跳问题分解为单跳问题。第一种方法是使用序列到序列的生成式模型^[62,63] 将多跳问题转化为多个单跳问题，并用检索模型和阅读理解模型分别筛选最佳文档，抽取最佳答案。第二种方法是在每次生成当前问题时，利用前驱问题和前驱答案来引导生成过程，并通过检索模型和阅读理解模型抽取单跳问题的答案，直至遇到带有结束标志的单跳问题。这两种方法可以有效地解决多文档阅读理解任务中的问题复杂性和推理能力缺失问题。

(3) 基于对比学习的多项选择长文本阅读理解研究

在多项选择阅读理解任务中，由于部分干扰选项与正确选项在字面表述上过于相似，同时各个选项在编码阶段缺乏交互，因此存在着正确率偏低的问题。为此，本文提出采用对比学习方法^[64]，增强选项文本的编码表示能力，以提高选项的区分度；同时利用样本内自注意力交互机制^[36]，建立多个选项之间的联系，从而使各个选项

之间的编码能够相互影响，提高答案的准确性。

1.4.2 组织结构

本文共分为六个章节，论文组织结构和各个章节的主要内容如下：

第一章绪论。本章主要介绍长文本阅读理解研究的背景和意义，通过对国内外研究现状的分析，总结目前长文本阅读理解存在的问题和研究难点。同时，本章还针对论文的研究内容和组织结构进行介绍。

第二章任务定义和评价方法。本章首先阐述了面向长文本的机器阅读理解任务的定义，并从样本分布和语言风格等角度对三个不同数据集进行了详细的实验语料资源分析，以此为基础对长文本机器阅读理解进行研究。最后，本章对长文本阅读理解任务的性能评价指标进行了探讨。

第三章基于检索器和阅读器架构的长文本阅读理解研究。本章针对长文本阅读理解领域，提出了一种基于检索器-阅读器二阶段架构的方法。该方法首先通过检索器，利用问题与文本片段之间的相关性，筛选出最有可能包含答案的一些文本片段。接着，通过阅读器，采用当前比较先进的预训练语言模型，从这些精简后的文本片段中抽取最终的答案片段。

第四章基于问题分解的多跳长文本阅读理解研究。本章针对多跳阅读理解领域，提出了一种利用问题分解技术来简化问题的方法。该方法通过使用序列到序列的生成式模型，将复杂的多跳问题分解为多个单跳问题。然后，利用检索模型和阅读理解模型分别筛选出支持文档，并从中抽取最佳答案。

第五章基于对比学习的多项选择长文本阅读理解研究。本章针对多项选择长文本阅读理解领域，提出了一种以对比学习为主的研究方法。该方法首先通过稠密向量检索的方式，从长文本中筛选出与问题和选项相关的文本片段。接着，在阅读器中引入了样本内自注意力机制，以增强多个选项之间的交互，并应用对比学习方法来提高文本的表示能力。

第六章总结和展望。本章对全文工作进行全面的总结，并对未来的研究方向进行展望。

第二章 任务定义及评价方法

机器阅读理解是指让计算机能够像人类一样理解和回答自然语言文本中的问题，它是自然语言处理技术中的一个重要分支。机器阅读理解是指让机器模拟人类阅读理解的能力，即让机器能够自动地从文本中抽取出问题的答案。它是自然语言处理中的一个重要任务。几年来，将预训练语言模型应用于机器阅读理解任务中，已经取得了很好的效果。

长文本机器阅读理解是在机器阅读理解的基础上进一步发展，以解决输入长文本篇幅的问题。这项技术在处理大量长文本时具有重要价值，能够帮助机器有效提取和分析文本中的信息和关系，从而节省人工阅读和分析的时间和成本。随着数字化时代和互联网的兴起，人们面临的信息量越来越大，长文本机器阅读理解技术可以帮助人类更好地理解和利用这些信息。此外，该技术还可以整合更多信息和背景知识，以提高智能问答和文本自动生成等应用的准确性和效率。然而，长文本机器阅读理解也面临着挑战，例如长文本中存在大量的自然语言表达的多样性和歧义，需要机器具备处理这些多义性和歧义性的能力，以正确理解文本的含义。此外，长文本还涉及到大量上下文信息和推理过程，需要机器具备对上下文信息的理解和推理能力。

本文旨在解决长文本阅读理解领域的共性问题和各个细分领域的难点，基于不同领域的长文本数据集，提出了三种不同的方法。本章将详细介绍面向长文本的机器阅读理解的任务定义，实验过程中所采用的语料，以及该任务的相关评价指标。

2.1 任务定义

长文本机器阅读理解的任务定义等同于常规的机器阅读理解，可以表述为：给定一个篇幅较长的文本文档 D (Document) 和一个问题 Q (Question)，目标是输出与该问题 Q 相关的文本片段或答案 A (Answer)。假设文本文档 D 表示为一个由 n 个单词组成的序列 $D = [w_1, w_2, \dots, w_n]$ 。问题 Q 表示为一个由 m 个单词组成的序列 $Q = [q_1, q_2, \dots, q_m]$ ，那么机器阅读理解的任务可以数学化的表示为：

$$A = f(D, Q)$$

公式中的 f 是一个需要让机器去学习的预测函数，通过向函数 f 提供文本文档 D 和问题 Q ，预测出答案 A 。根据答案风格和表现方式的不同，长文本机器阅读理解任务可以归纳为以下三种类型¹：

(1) 抽取式 (Extractive)

抽取式阅读理解任务是指给定一篇文本和一个问题，模型需要从给定的文本段落中“抽取”出正确的答案片段来回答问题。抽取式阅读理解实际上是一个分类问题，它要求机器预测答案的开始位置 pos_{start} 与结束位置 pos_{end} 。最终答案可以表示为 $D[pos_{start} : pos_{end} + 1]$ 。如表 2-1 中的抽取式样例所示，列举了 NewsQA 数据集中的一一个样例。对于问题“*When was Pandher sentenced to death?*”，答案“*February*”对应文档中的一个连续的自然文本片段。

(2) 生成式 (Generative)

生成式阅读理解要求机器基于给定的文本和问题，生成一个合适的答案，该答案不局限于文章中存在的词语，而是自由生成的。这种任务更适合实际生活场景，但是由于生成的句子无法做准确评估，因此一直无法成为业界的主流数据集。如表 2-1 中的生成式样例所示，列举了 NarrativeQA 数据集中的一条数据。对于问题“*Why does Dorothy want to go home?*”，答案“*Because she misses her family and friends.*”可以不是文档中的一个连续的文本片段，而是一个自由阐述的文本观点。

(3) 多项选择 (Multiple Choice)

多项选择阅读理解要求机器基于给定的文章和问题，从多个备选答案中选择一个最有可能是正确答案的选项。这种任务类似于英语考试中的阅读理解选择题。当前处理这类问题的模型基本都采用 $<$ 问题，文章，答案 $>$ 三元组的框架。如表 2-1 中的多项选择样例所示，列举了 QuALITY 数据集中的一一个样例。对于问题“*Why did Roylott want to kill Helen?*”和四个候选答案，机器需要从这四个答案中选出最合适的一个。

¹实际上，机器阅读理解任务按照答案类型可以分为抽取式、生成式、完形填空和多项选择四种，本文的实验数据采用了 NewsQA，MuSiQue 和 QuALITY，分别为抽取式和多项选择类型的答案，并涉及了生成式的方法。因此，本文主要考虑这三种类型的机器阅读理解任务。

表 2-1 长文本阅读理解样例

(1) 抽取式样例

文章:

(...) A high court in northern India on Friday acquitted a wealthy businessman facing the death sentence for the killing of a teen in a case dubbed “the house of horrors.” Moninder Singh Pandher was sentenced to death by a lower court in February. (...)

< 译文: (...) 上周五, 印度北部的一家高级法院宣布无罪释放了一位被指控在一起被称为“恐怖之家”的案件中杀害一名青少年的富商, 他曾被下级法院判处死刑。Moninder Singh Pandher 在二月份被下级法院判处死刑。(...) >

文本长度: 644

问题:

When was Pandher sentenced to death?

< 译文: Pandher 何时被判死刑? >

答案:

February

< 译文: 二月 >

(2) 生成式样例

文章:

(...) She took off her old leather shoes and tried on the silver ones, which fitted her as well as if they had been made for her. Finally she picked up her basket. “Come along, Toto” she said. “We will go to the Emerald City and ask the Great Oz how to get back to Kansas again.” (...)

< 译文: (...) 她脱下旧皮鞋试穿上那双银鞋, 它们非常适合她, 好像是为她量身定制的一样。最后她拿起篮子, “走吧, 托托,”她说, “我们去祖母城问伟大的奥兹怎样才能回到堪萨斯。” (...) >

文本长度: 32,655

问题:

Why does Dorothy want to go home?

< 译文: 为什么 Dorothy 想要回家? >

答案:

Because she misses her family and friends.

< 译文: 因为她想念她的家人和朋友。>

(3) 多项选择样例

文章:

(...) “The Adventure of the Speckled Band” is one of the 56 short Sherlock Holmes stories written by Sir Arthur Conan Doyle. It was first published in 1892. The story tells how Sherlock Holmes and Dr. Watson investigate the mysterious death of Julia Stoner, the sister of their client Helen Stoner, at Stoke Moran Manor. (...)

< 译文: (...) 《斑点带的冒险》是由亚瑟·柯南·道尔爵士所写的 56 个短篇福尔摩斯侦探小说之一, 首次出版于 1892 年。故事讲述了福尔摩斯和华生博士调查客户海伦·斯通纳的姐姐朱莉娅·斯通纳在斯托克·莫兰庄园的神秘死亡事件。(...) >

文本长度: 6,633

问题:

Why did Roylott want to kill Helen?

< 译文: (...) 为什么 Roylott 想杀死 Helen? (...) >

选项:

A. Because he wanted to inherit their mother’s fortune

< 译文: 因为他想要继承他们母亲的财产。>

B. Because he wanted to marry Helen;

< 译文: 因为他想和 Helen 结婚。>

C. Because he wanted to sell Helen to a circus;

< 译文: 因为他想把 Helen 卖到马戏团。>

D. Because he wanted to use Helen for experiments.

< 译文: 因为他想用 Helen 做实验。>

答案: A

2.2 语料资源

本文在新闻长文本阅读理解数据集 NewsQA, 多跳多文档阅读理解数据集 MuSiQue, 以及多项选择长文本阅读理解数据集 QuALITY 三个公开语料上进行相关实验。本节将分别介绍这三个数据集的详细信息。

2.2.1 长文本数据集 NewsQA

NewsQA 是由纽约大学、卡耐基梅隆大学和麻省理工学院联合推出的一个大规模新闻阅读理解数据集。该数据集覆盖了新闻报道中的多个主题和事件，涉及政治、经济、文化等各个领域，旨在为机器阅读理解任务提供具有挑战性的现实世界应用场景。

(1) 样本分布

NewsQA 数据集包括超过 10,000 篇新闻文章，以及超过 100,000 个与这些文章相关的问答对。其训练/开发/测试集的分布如 2-2 所示。此外，表格中还统计了与文本长度相关的数据指标。表格中的 TPP 表示每篇文章中的 token 数量 (tokens per passage)，PPP 表示每篇文章中的段落数量 (paragraphs per passage)。从这些数据指标可以看出，大多数文本长度超出了 512 这个长度限制。实际上，NewsQA 中的文本还包含了复杂的语句和语法结构。在后续方法中，本文也着重考虑到处理大量数据和复杂数据结构的问题。另外，至少有一半的文本，其段落数量达到了 18。这对于段落检索也造成了一定的挑战。

表 2-2 NewsQA 数据统计

	问题数量	TPP (中位数)	TPP (最大值)	PPP (中位数)	PPP (最大值)
训练集	90k	774	3.1k	18	87
开发集	5k	734	2.3k	18	63
测试集	5k	707	2.3k	17	54

(2) 文本风格

NewsQA 数据集的特点在于其问题和答案是由人工构造的，并且问题涉及多种类型，如实体识别、原因分析、情感分析、时间和日期等多个方面，从而涵盖了各种常见问题类型。其次，NewsQA 中的问题往往需要依赖于上下文来进行回答，而不是简单地基于问题本身。这意味着处理 NewsQA 数据需要考虑到上下文的语义信息。

另外，NewsQA 中的文本是由不同的新闻文章组成的，这些文章的句子结构和语法风格可能会有所不同。因此，在处理 NewsQA 数据时需要考虑到句子结构和语法的多样性。最后，NewsQA 数据集中的问题还包含了人类提问时可能存在的模糊性、歧义性和主观性等因素，因此对机器阅读理解模型的能力提出了更高的要求。

NewsQA 数据集已成为机器阅读理解任务中的重要标准基准数据集之一，被广泛应用于学术界和工业界的相关研究工作中。

2.2.2 多跳数据集 MuSiQue

MuSiQue 是一个多跳阅读理解数据集，它是专门为多跳阅读理解任务而设计的。该数据集通过其他机器阅读理解数据集，使用自下而上的方法进行构建，该方法系统地选择相互关联的可组合成对单跳问题，其中一个推理步骤主要依赖于来自另一个步骤的信息。这种方法使作者能够探索广阔的问题空间，并添加严格的过滤器以及其他针对关联推理的机制。它可以对构造过程和由此产生的 k-hop 问题的属性进行精细控制。每个问题都需要回答一个自然语言问题，其中问题的答案可能需要跨越多个句子。为了回答这些问题，需要进行多次推理和跳转，涉及多个句子和段落。

(1) 样本分布

MuSiQue 数据集总共包含超过 4,000 个多句子文档，以及超过 10,000 个与这些文档相关的问答对。训练/开发集的分布如 2-3 所示。表格中的后两列是与文本长度相关的统计数据。其中，对于超过 99% 以上的问题来说，提供的文档数为 20 个；另外不到 1% 的问题提供少于 20 个文档。TPD 表示每个 MuSiQue 文档中的平均 token 数量 (tokens per document)。从这些数据指标可以看出，针对每个提问，需要阅读的 token 数量是非常多的，平均可以达到 2,000 以上，这意味着相比于其他的阅读理解数据结构，MuSiQue 涵盖了更多的语义信息和上下文信息。

表 2-3 MuSiQue 数据统计

	问题数量	包含 20 个文档的问题数量	占比	TPD
训练集	19,938	19,917	99.89%	112
开发集	2,417	2,401	99.34%	109

更重要的是，MuSiQue 包含了大量的 2-4 跳的问题，如表 2-4 所示，展示了一些典型的问题。在后续方法中，本文着重考虑如何将多跳问题分解为一些简单的单跳问

题，从而进行下一步工作。

表 2-4 MuSiQue 问题跳数统计

类别	占比	示例
2 跳	72.1%	Who succeeded the first President of Namibia?
3 跳	22.0%	What currency is used where Billy Giles died?
4 跳	5.9%	When did Napoleon occupy the city where the mother of the woman who brought Louis XVI style to the court died?

(2) 文本风格

MuSiQue 包含 2-4 跳的问题，每个问题都需要对多个句子进行理解和推断。与其他机器阅读理解数据集相比，MuSiQue 数据集中的问题具有较高的复杂度和挑战性，因为它需要对故事中的多个句子进行理解、推理和跳转。此外，该数据集还包含各种类型的问题，包括原因、解释、转折和逻辑推理等。MuSiQue 数据集的发布可以促进研究人员在多跳阅读理解任务上进行深入研究和评估。

2.2.3 多项选择数据集 QuALITY

QuALITY 数据集是由纽约大学的研究者创建的。研究者采集了来自于维基百科、英文小说和新闻文章，来构成阅读文本。这些文本涵盖了多种主题，如历史、科学、文学、政治等。QuALITY 由人工编写问题，每个问题有四个选项，其中一个是正确答案。

QuALITY 数据集的难点和挑战在于段落的长度和复杂性，以及问题的多样性和深度。这个数据集有助于提高长文本理解的能力，对于一些需要处理长篇文章或书籍的应用场景很有价值。

(1) 样本分布

QuALITY 数据集收录了 381 个左右的长篇文档，这些文章的平均长度在 4,700 个英文单词左右。具体的训练/开发/测试集的数据分布如 2-5 所示。同时，QuALITY 中包含了一部分困难问题，这些困难问题由数据标注者标出他们认为难以回答的问题。

(2) 文本风格

相比抽取式阅读理解数据集，多项选择数据集 QuALITY 更注重逻辑推理能力。

表 2-5 QuALITY 数据统计

	文章数量	问题数量	困难问题数量	困难问题占比
训练集	150	2,523	1,251	49.5%
开发集	115	2,086	1,065	51.1%
测试集	116	2,128	1,044	49.1%
全部	381	6,737	3,360	49.9%

数据集中提问的风格各不相同，有些是事实性的询问，有些是推理性的判断，有些是意见性的评价。表 2-6 中随机抽取了 500 条数据，统计其推理类型。

表 2-6 MuSiQue 数据集 500 条样例数据的推理类型统计

推理类型	困难数量	简单数量	类型占比
描述	89	77	33.2%
为什么/理由	73	83	31.2%
象征意义/解释	76	63	27.8%
其他	107	111	43.6%

2.3 性能评价指标

对于抽取式阅读理解数据集 NewsQA 和 MuSiQue，本文采用精准匹配 EM 和调和匹配 F1 来衡量模型预测答案文本片段的准确性；同时，针对 MuSiQue 中需要生成问题的阶段，本文也采用了 BLEU、METEOR 和 ROUGE 等指标来评估生成的可靠程度。对于多项选择数据集 QuALITY，本文采用准确率 Acc 来评估模型在多个备选答案中选择正确答案的能力。

针对抽取式阅读理解数据集 NewsQA 和 MuSiQue，本文选择了 EM 和 F1 这两个指标来衡量模型预测答案文本片段的准确性。此外，针对 MuSiQue 中需要生成问题的阶段，本文还使用了 BLEU、METEOR 和 ROUGE 等指标来评估生成的可靠程度。对于多项选择数据集 QuALITY，本文采用了准确率 Acc 来评估模型在多个备选答案中选择正确答案的能力。以上指标的选择充分考虑了不同任务的特点，以确保模型的性能能够得到全面的评估。

(1) EM

EM (Exact Match) 指标用于衡量模型的预测答案是否与真实答案完全匹配。当

模型的预测答案与真实答案完全一致时，EM 值为 1；反之，如果预测答案与真实答案不完全匹配，EM 值为 0。EM 指标的优点是简单直观，能够快速评估模型的准确性，但缺点是对于稍有不同的答案就会评估为错误，对模型的容错性要求较高。

(2) F1

F1 是一种综合考虑模型精精确率（Precision，P）和召回率（Recall，R）的指标，通常用于评估二分类或多分类任务中的模型性能。在抽取式阅读理解任务中，F1 被用来衡量模型预测答案文本片段的准确性。它是精确率和召回率的加权平均值，用公式表述为：

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

其中， P 是指模型正确预测的 token 数量与总预测 token 数量的比率，即 $TP/(TP+FP)$ ； R 是指模型正确预测的 token 数量与真实 token 数量的比率，即 $TP/(TP+FN)$ 。其中，TP 表示真正例，即模型正确预测为正例的 token 数量；FP 表示假正例，即模型错误地将负例预测为正例的 token 数量；FN 表示假负例，即模型错误地将正例预测为负例的 token 数量。F1 的取值范围为 0 到 1，越接近 1 表示模型的性能越好。并且 F1 更注重模型的综合表现

(3) BLEU BLEU (Bilingual Evaluation Understudy) 是一种用来评估机器翻译质量以及其他生成任务的指标，它通过计算候选文本和参考文本之间的 n 元语法匹配度来衡量它们的相似度。

BLEU 的计算公式可以表述如下：

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

其中，BP 是惩罚因子，用来防止过短的候选文本获得高分。它定义为：

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$$

公式中，c 是候选文本的长度，r 是参考文本的有效长度（即与 c 最接近的长度）；N 是最大的 n 元语法长度，通常取 4；wn 是权重系数，通常取 $1/N$ ；pn 是 n 元语

法精确度，即候选文本中与参考文本匹配的 n 元语法数量除以候选文本中所有 n 元语法数量。

BLEU 指标的取值范围为 0 到 1，值越高表示机器翻译的结果与参考结果越接近。

(4) METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) 是一种基于单词精确度和召回率的调和平均数的指标，同时考虑了词干、同义词和词序的匹配，常用于评估机器翻译系统的生成结果。

METEOR 的计算公式如下：

$$METEOR = \frac{10PR}{R + 9P}(1 - Penalty)$$

其中，P 是单词精确度，即候选文本中与参考文本匹配的单词数量除以候选文本中所有单词数量；R 是单词召回率，即候选文本中与参考文本匹配的单词数量除以参考文本中所有单词数量；Penalty 是惩罚因子，用来降低过多切分或乱序的候选文本得分。它定义为：

$$Penalty = 0.5\left(\frac{Chunks}{Matches}\right)^3$$

公式中的 Chunks 是候选文本中与参考文本匹配的连续单词块数量；Matches 是候选文本中与参考文本匹配的总单词数量。

METEOR 的取值范围是 0 到 1 之间

(5) ROUGE

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) 是一种用于评估自然语言生成系统输出的指标，它是以召回率为导向的评估指标，旨在衡量系统生成的文本与参考答案之间的重叠程度。

ROUGE 指标分为多个变种，本文用到的是 ROUGE-N，它测量系统生成文本与参考答案中 N 个连续词汇的重叠程度。其计算公式为：

$$ROUGE_N = \frac{\sum_{r \in \text{Reference}} \sum_{n \in n\text{-grams}} Count_n(r) \cdot \min Count_n(r), Count_n(\text{System})}{\sum_{r \in \text{Reference}} \sum_{n \in n\text{-grams}} Count_n(r)}$$

其中， n 表示 n -gram 的长度， r 表示参考答案，System 表示自动生成的文本。 $Count_n(r)$ 表示在参考答案中 n -gram n 在 r 中出现的次数， $\min Count_n(r), Count_n(\text{System})$ 表示参考答案和自动生成的文本中 n -gram n 的最小出现次数。 $\sum_{n \in n\text{-grams}}$ 表示对所有长度为 n 的 n -gram 求和， $\sum_{r \in \text{Reference}}$ 表示对所有参考答案求和。

(6) ACC

在多项选择阅读理解任务中，ACC (Accuracy) 是最常用的评价指标之一，它衡量模型对问题的正确率。

准确率是最常见的分类评价指标，其计算公式为模型分类正确的样本数量除以总样本数量。其计算公式如下：

$$ACC = \frac{TP + TN}{P + N}$$

其中， P 表示正例样本数量， N 表示负例样本数量， TP 和 TN 分别表示模型预测正确的正负例样本数量。

2.4 本章小结

本章首先对面向长文本的机器阅读理解任务进行了明确定义；接着，本章从三个数据集的样本分布、语言风格等多方面对数据集进行了分析，指出了不同数据集的特点和难点，为后续的研究和模型选择提供了参考；最后，本文介绍了几个关键的评价指标，这些指标可以用于评估模型的性能和指导模型的优化和改进。

第三章 基于检索器和阅读器架构的长文本阅读理解

长文本机器阅读理解是指在给定一段长文本的情况下，让模型回答特定问题。虽然基于 Transformer 的模型已经取得了很好的成果，但是由于时间开销的问题，大多数模型并不擅长处理长序列。一般来说，滑动窗口是其中一个合适的解决方案。这种方法将文章等分成多个片段，针对每个文本片段独立预测答案，而不考虑文本片段的上下文关系。然而，这种方法缺乏上下文之间的远距离依赖，这会严重损害模型性能。为了解决这个问题，本文提出了一个专门针对长文本阅读理解的两阶段方法 ThinkTwice。

ThinkTwice 解决长文本阅读理解的过程主要分为两个步骤。首先，检索出最终答案最有可能位于的若干个文本片段；然后，从这些精简后的文本片段中抽取最终的答案片段。本章在 NewsQA 数据集上进行了实验。实验结果表明，ThinkTwice 可以从长文本中捕获到最具有信息含量的文本片段。同时，ThinkTwice 与现有的基线模型相比，获得了相当大的性能提升。

3.1 引言

机器阅读理解技术^[6] 旨在针对给定文本，教机器学习回答问题。这项技术一直是自然语言处理领域的研究热点之一。预训练语言模型采用多层 Transformer 架构和自注意力机制^[36]，已经取得了显著的成果。

尽管现有的机器阅读理解系统（以及其他自然语言处理系统）在短文本领域中取得了成功，但由于预训练语言模型所能容纳的文本长度限制¹，这些系统仍然不善于有效地处理长序列。同时，如果仅仅是增加输入长度，模型的复杂度 ($O(n^2)$) 也将呈现平方级的增长，这会导致维度爆炸现象的出现。

在处理长文本时，最直观的方法是截断^[12,65] 和滑动窗口^[30]。截断方法将长文本截断为模型所能接受的长度，而滑动窗口方法将文章划分为若干个固定长度的片段，并对每个片段预测答案。然而，这两种方法都存在问题，因为它们舍弃了部分文本，或者丢弃了关键的上下文信息。这些问题都是由于时间和空间的高复杂度带来的。因此，另一类研究方法主张简化 Transformer 架构^[66-68]。然而，这些方法由于自身存在

¹例如，BERT 的最大位置词嵌入长度为 512。

的问题，在现实世界中很少被应用。

本章受到人类阅读行为的启发，提出了一种名为 ThinkTwice 的二阶段方法，旨在解决长文本阅读理解中的挑战。该方法主张将长文本压缩为短文本，以模拟人类在阅读长文本时的选择性阅读行为。具体而言，当人们面对一篇长文本时，他们会无意识地选择与给定问题相关的文本片段，并将这些信息整理到工作记忆^[69] 中，以推理出答案。基于这一人类行为，ThinkTwice 采用了检索器和阅读器两个模块，分别用于过滤和压缩大量文本信息以及实现问答功能。此外，ThinkTwice 还使用了分段模块对长文本进行分段，以及融合模块来整合检索得到的关键信息，以提高阅读理解的准确性和效率。

本章在 NewsQA 数据集^[14] 上对提出的方法 ThinkTwice 进行了评估和验证。该数据集的文本通常比较长，并且是新闻文本。实验结果表明，相较于一些基线模型^[21,70,71]，ThinkTwice 实现了重要的提升。特别的，该方法通过在第一阶段检索出一些具有信息量的段落，从而实现了可观的性能提升，这极大的提升了第二阶段推理的准确性。

本章的贡献主要如下：

- 在长文本阅读理解领域，本章提出了一种全新的方法 ThinkTwice，该方法将长文本压缩为短文本片段，以取代先前直接处理长文本的方法。
- 实验结果表明，对于长文本阅读理解数据集 NewsQA^[14]，ThinkTwice 方法在四个主要的预训练语言模型^[21,22,70,72] 中均取得了可观的性能提升。

3.2 基于检索器和阅读器架构的长文本阅读理解

图 3-1 展示了 ThinkTwice 的架构，它由四个基本模块组成：1) 一个分段器，将给定文章切割为更短的文本片段；2) 一个检索器，筛选出与给定问题最相关的一些文本片段；3) 一个融合器，将筛选出的文本片段根据原始顺序进行整合；4) 一个阅读器，阅读给定问题和融合后的文本片段，从而预测出最终答案。

3.2.1 分段方法

长文本阅读理解的主要挑战在于如何在给定文章中的大量知识片段中准确地定位到最重要的信息。这些文章的长度通常超过现有神经网络模型所能容纳的最大长

度（例如 512 个 token）。为了应对这个问题，我们在 NewsQA 数据集中为每篇文章的结尾添加了一个分割标记。通过这种方式，可以将输入文本 P 分割成文本片段 P_1, P_2, \dots, P_n ，其中每个文本片段的长度限制在 60-80 个 token 左右。这种形式化的分割方法可以有效地提高模型的处理效率和准确性。

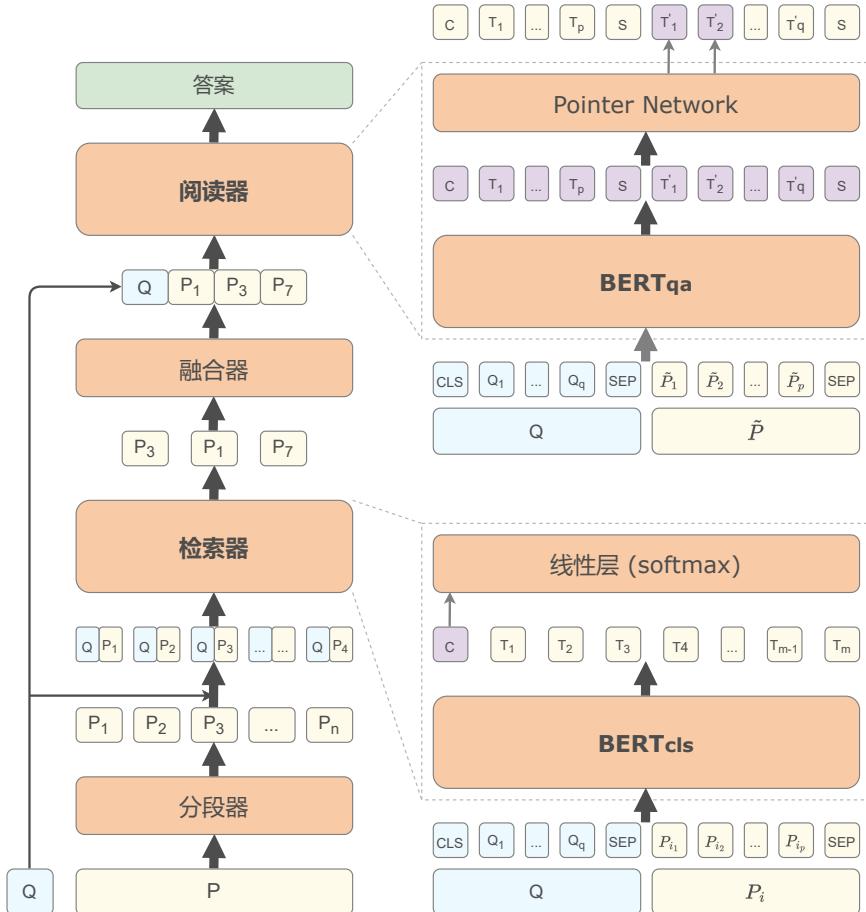


图 3-1 ThinkTwice 的基本架构

3.2.2 基于预训练语言模型的段落检索器

当人们需要从大量文本中寻找答案时，通常会选择保留与当前问题最相关的文本片段，并过滤掉琐碎的信息。受到这种人类行为的启发，本文提出了一个基于预训练语言模型的检索器，来选择最有可能回答问题的最重要的文本片段。

具体而言，检索器部分首先将问题 Q 和每个文本片段 $\{P_i\}_{i=1}^n$ 组合成多个序列 $\{x_i\}_{i=1}^n$ ，其中 $x_i = [CLS]Q[SEP]P_i[SEP]$ ²。然后，检索器的编码器部分，也就是预训

²[CLS] 和 [SPE] 是特殊 token。前者对于编码后的输入序列，理论上可以表示整体信息；后者主要用于分隔输入序列

练习语言模型，用于将输入 x_i 编码成整个序列的上下文词嵌入表示 H_i :

$$H_i = BERT_{cls}(x_i) \quad (3.1)$$

[CLS] 的隐向量表示 H_i^{cls} 代表了整个序列的总体表示。然后，一层线性网络和 Softmax 层用于得到该序列的分类概率 \hat{y}_c :

$$\hat{y}_c = \text{Softmax}(\text{Linear}(H_i^{cls})) \quad (3.2)$$

其中 \hat{y}_c 表示当前文本片段 P_i 是否包含可以回答给定问题的有效信息^[73]。在训练阶段，当 P_i 包含标注答案时，“有/无答案”标签 $y_{c(i)}$ 设为 1，否则设为 0。

交叉熵损失函数用于计算真实答案 y_c 和预测概率 \hat{y}_c 之间的损失:

$$\mathcal{L}_{retriever} = -\frac{1}{n} \sum_{i=1}^n [y_{c(i)} \log \hat{y}_{c(i)} + (1 - y_{c(i)}) \log(1 - \hat{y}_{c(i)})] \quad (3.3)$$

3.2.3 融合方法

上一小节利用检索器得到了输出概率值 $\{\hat{y}_{c(i)}\}_{i=1}^n$ ，该值可以量化每个文本片段 $\{P_i\}_{i=1}^n$ 回答给定问题 Q 的可能性大小。接下来，根据 \hat{y} 的值，选择与问题 Q 最相关的 k 个文本片段，并舍弃其他片段。被选中的文本片段按照它们在原文中的顺序合并成单个序列，以确保整个组合文本的语义连贯性和上下文连续性。例如，当 $k = 3$ 且分数最高的 3 个文本片段分别是 P_3 , P_1 和 P_7 时，融合器需要根据它们在原文中的相对顺序将它们拼接在一起。对于可能出现的特别长的文本，融合器可以直接对其进行截断。因此，融合器的输出可以表示为图 3-1 中的 $\tilde{P} = < P_1, P_3, P_7 >$ 。

3.2.4 基于预训练语言模型的阅读理解模型

通过以上模块，已经从原始的长文本 P 中提取了一段短文本 \tilde{P} 。本节中的阅读器模块首先将问题 Q 和上一节得到的 \tilde{P} 拼接成一个单独的序列 $z = [\text{CLS}]Q[\text{SEP}]\tilde{P}[\text{SEP}]$ ，随后使用另一个预训练语言模型 BERT^[21] 作为阅读器的编码器，将输入 z 映射到一个上下文隐向量序列。下一步，指针网略将问题相关的文本表示的答案片段的起始和

结束位置进行解码：

$$\hat{y}_s, \hat{y}_e = PN(BERT_{qa}(z)) \quad (3.4)$$

其中 \hat{y}_s 和 \hat{y}_e 分别表示由阅读器解码的预测答案的起始和结束位置的概率。

在训练过程中，阅读器中使用了交叉熵损失来计算起始和结束位置的损失：

$$\mathcal{L}_{reader} = \frac{1}{2} CrossEntropy(\hat{y}_s, y_s) + \frac{1}{2} CrossEntropy(\hat{y}_e, y_e) \quad (3.5)$$

其中， y_s 和 y_e 分别表示开始和结束位置的标签。如果当前段落给出的信息不能回答问题， y_s 和 y_e 都设为 0，也就是指向 [LCS] token。在预测阶段，阅读器首先计算“有答案”分数 $score_{has}$ 和“无答案”分数 $score_{null}$ ，各自用 \hat{y}_s 和 \hat{y}_e 来表示：

$$score_{has} = \max_{1 \leq i \leq j < L} (\hat{y}_s^{(i)} + \hat{y}_e^{(j)}) \quad (3.6)$$

$$score_{null} = \hat{y}_s^{(0)} + \hat{y}_e^{(0)} \quad (3.7)$$

其中 i 和 j 表示在整个序列长度 L 内的答案位置，并且由于起始位置一定在结束位置之前，有 i 一定限制为小于 j 。另外，鉴于序列首部的 [CLS] 不表示 \tilde{P} 中的任何词， $\hat{y}_s^{(0)}$ 和 $\hat{y}_e^{(0)}$ 表示整个文本没有答案的概率。

接下来，通过计算 $score_{has}$ 和 $score_{null}$ 之间的距离来计算两者之间的距离分数 $score_{dist}$ ，以此作为“有/无答案”的依据：

$$score_{dist} = score_{null} - score_{has} \quad (3.8)$$

$$s, e = \operatorname{argmax}_{1 \leq i \leq j < L} (\hat{y}_s^{(i)} + \hat{y}_e^{(j)}) \quad (3.9)$$

其中有一个阈值 δ ，如果 $score_{dist}$ 小于 δ ，阅读器输出的 s 和 e 就代表了答案的起始和结束位置，否则阅读器会将该问题视为一个不可回答问题。

3.3 实验及结果分析

3.3.1 实验设置

本章在一个具有挑战性的长篇文本阅读理解数据集 NewsQA 上进行了实验。该数据集包含来自 CNN 的 13,000 篇新闻文章，以及 120,000 条人工生成的问题答案对。在排除了 20,000 条标注者认为没有意义的低质量问题后，本章对剩余的数据进行了实验分析。因此，NewsQA 的训练/开发/测试集分别包含 90,000/5,000/5,000 条问题答案对。此外，本章还具体统计了每篇文章的 token 数量（即 tokens per passage, TPP）和每篇文章的段落数量（即 paragraphs per passage, PPP）。在训练/开发/测试集上，TPP 的中位数分别为 774/734/707，最大值分别为 3,100/2,300/2,300。而 PPP 在三个集上的中位数为 18/18/17，最大值为 87/63/54。

针对 NewsQA 数据集，存在一些长文本模型。

- Match-LSTM^[74]。该模型使用了两个单向的 LSTM 来编码问题和文章。
- BiDAF^[75]。BiDAF 的核心思想是其中的双流注意力层，双流注意力分别计算上下文之于问题的注意力以及问题之于上下文的注意力。
- AMANDA^[76]。AMANDA 针对答案抽取提出了一种端到端的专注于问题的多因子注意力网络。多因子注意力编码聚合了位于多个句子中有意义的事实。
- DecaProp^[71]。该模型把自注意力网络整合进 RNN 模^[77]型。
- Longformer^[66]。该预训练语言模型使用了稀疏注意力矩阵，来解决序列长度的限制。
- CogLTX^[78]。CogLTX 架构通过判别多个句子间的相关性来识别重要句子。

本章采用 EM 和 F1 这两个官方指标来衡量长文本机器阅读理解的性能。其中，EM 指标用于衡量预测答案与真实答案完全匹配的比例，而 F1 指标则用于衡量预测答案与真实答案在 token 级的平均重叠程度。

为了验证两阶段阅读策略的有效性，本章使用了部分预训练语言模型，并采用将文章划分成多个段落的滑动窗口机制来处理长文本阅读理解问题。所采用的预训练语言模型包括 BERT^[21]、RoBERTa^[22]、ALBERT^[72] 和 SpanBERT^[70]，这些模型均基于 Transformer^[36] 架构，且使用 PyTorch 实现。在训练阶段，基础模型的学习率设为 2e-5，而大模型的学习率则设为 2e-6。此外，热身比例设为 0.1，L2 权重衰减设

为 0.01。批量尺寸在基础模型中为 8，在大模型中为 1。对于检索器，轮次数量在基础模型中设为 1，在大模型中设为 2；而阅读器中这个值保持为 3。在分词方面，本章采用 wordpieces^[79] 进行分词，并将第一阶段的最大长度设为 256，第二阶段的最大长度设为 512。第一阶段中进行了大量实验，用来选择 k 值³。

表 3-1 长文本阅读理解模型在 NewsQA 上的表现，以及 CoLISA 和对应的预训练语言模型在 NewsQA 上的表现

模型	开发集		测试集	
	F1	EM	F1	EM
Match-LSTM ^[74]	49.6	34.4	50.0	34.9
BiDAF ^[75]	-	-	52.3	37.1
AMANDA ^[76]	63.3	48.8	63.7	48.4
DecaProp ^[71]	65.7	52.5	66.3	53.1
Longformer-base ^[66]	68.1	58.3	68.1	58.1
CogLTX ^[78]	-	-	70.1	55.2
BERT-base ^[21]	65.6	56.3	65.4	55.2
+ ThinkTwice(descending order)	66.6	57.8	65.8	55.6
+ ThinkTwice(ours)	68.5	58.8	68.6	57.7
RoBERTa-base ^[22]	63.7	53.5	63.2	53.1
+ ThinkTwice(ours)	67.7	58.6	67.7	58.4
ALBERT-base ^[72]	68.1	58.2	68.0	58.0
+ ThinkTwice(ours)	68.7	59.1	68.6	58.8
SpanBERT-base ^[70]	67.7	57.1	67.5	56.2
+ ThinkTwice(ours)	69.9	59.8	69.7	59.4
BERT-large	68.9	59.2	68.8	58.6
+ ThinkTwice(ours)	70.1	59.5	69.8	59.4
SpanBERT-large	71.2	61.8	70.9	59.8
+ ThinkTwice(ours)	<u>72.1</u>	<u>62.2</u>	<u>71.5</u>	<u>61.0</u>

3.3.2 实验结果和分析

与现有模型相比。表 3-1 呈现了 ThinkTwice 模型与现有模型的比较结果。该表说明，本章提出的模型在 NewsQA 数据集上表现卓越，超越了现有的长文本阅读理解模型。具体而言，在 EM 指标上，本模型相比现有模型提升了 5.8 个百分点，在

³ k 值是一个超参数，表示检索器筛选出的最好的 k 个段落。

F1 指标上提升了 1.4 个百分点。此外，实验结果也同样表明，实现 ThinkTwice 策略可以提高所有预训练阅读理解模型的性能。例如，在 BERT-base 模型上，F1 值提高了 3.2 个百分点，在 RoBERTa-base 模型上，F1 值提高了 4.5 个百分点。然而，在 ALBERT-base 模型上的提升并不显著。这可能有两个原因：首先，ALBERT 模型中的句子顺序预测（sentence-order prediction, SOP）预训练任务已经解决了句子内部连贯性的问题；其次，ALBERT 模型中实现了跨层参数共享机制，导致即使实现了新的策略，参数变化也非常微小。在其他模型上实验性能的大幅提升表明，ThinkTwice 策略中的检索器精准地抽取了最重要的段落，从而将长文本压缩成合适长度的短文本，同时最大程度地保留了原始文本中最重要的信息。

此外，为了研究不同融合方式对最终性能的影响，本章还尝试在 BERT-base 模型上采用两种不同的融合策略来合并筛选出的文本片段。其中一种方法是将检索器提取出的文本片段按照与给定问题的相关性进行降序排列并进行合并；另一种方法是按照先前所述的方法，以原文的顺序进行合并。从表格 3-1 中的第 8 行和第 9 行可以发现，原始的融合方式要显著优于降序的融合方式（提升 2.8），尽管采用降序方式排列的模型也超出了基线模型（0.4）。这个对比实验表明，扰乱序列顺序可能会导致上下文信息的丢失。

段落检索。图 3-2 展示了检索器的不同参数配置与 ThinkTwice 模型之间的关系。对于检索器，评估指标 Hits@ k （前 k 个最准确的）衡量了检索器筛选出的最佳 k 个段落是否包含真实答案。对于 ThinkTwice 模型，F1 评估 NewsQA 数据集上阅读理解模型的最终性能。红色曲线表明， k 值越大，Hits@ k 准确率就越高。同时，当 k 大于 3 时，检索器的该项指标也超过了 90%。此外，当 k 等于 5 时，ThinkTwice 模型（绿色曲线）实现了最佳性能（68.6）。这表明，当 k 值较小时，检索器无法召回足够多的备选段落；当 k 大于 5 时，更多的备选段落需要搜索更大的文本范围，从而导致性能下降。因此，ThinkTwice 在所有实验中将超参数 k 设为 5。本节还将 ThinkTwice 模型与相应的 BERT-base 阅读理解模型（蓝色曲线）进行了比较。结果显示，当 k 设为 2 到 9 之间时，ThinkTwice 模型的表现优于 BERT-base，这验证了 ThinkTwice 两阶段策略的有效性。

文本长度的作用。为了验证 ThinkTwice 在长文本领域的作用，本节将对多个不同长度的文本进行测试，并将测试结果与 BERT-base 和 Longformer-base 等阅读理解模型进行比较。值得注意的是，这里 ThinkTwice 模型的阅读器应用了 BERT-base。图 3-3 列出了实验结果。可以看到，在较短的文本长度范围内（[0,512] 和 [512,1024]），Long-

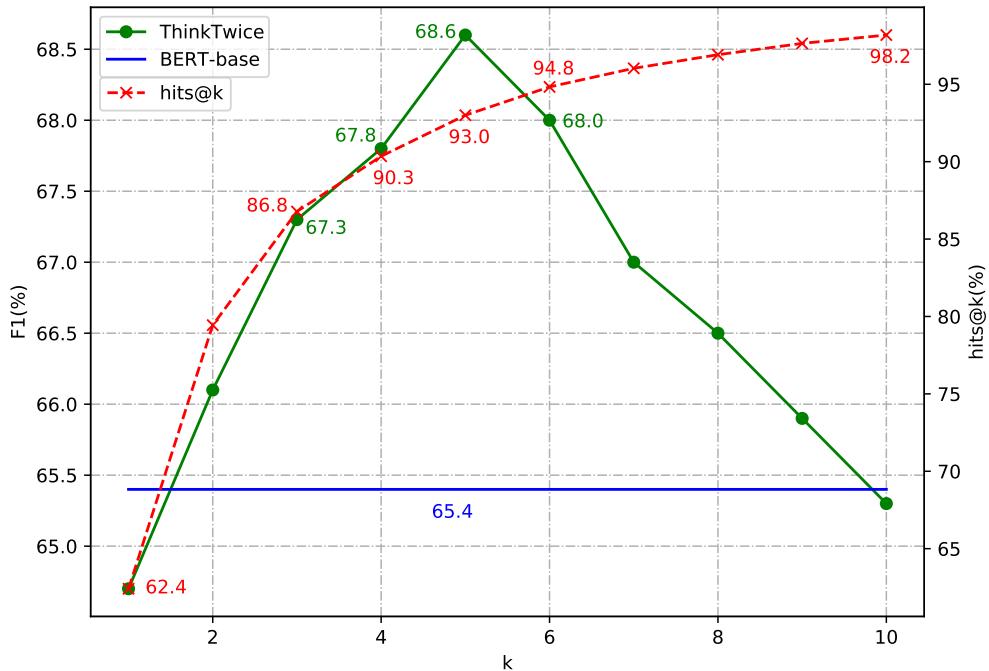


图 3-2 在不同 k 值情况下，检索器和 ThinkTwice 模型（基于 BERT-base）的性能关联

former 实现了最好的实验结果。其中的原因是 Longformer-base 继承了 RoBERTa 的预训练参数权重，而这些参数已经在阅读理解任务中表现得很好。然而，在较长的文档长度范围内 $((1024, 1536], (1536, +\infty))$ ，本章提出的 ThinkTwice 模型显著优于其他模型。这也证实了 ThinkTwice 能够准确地定位到包含答案的文本片段。此外，在较长文档的实验结果中，BERT-base 也要优于 Longformer-base，这表明滑动窗口机制 (BERT-base) 相较于直接的长文本输入 (Longformer-base)，也有一定优势。最后可以发现，随着文档长度的增加，ThinkTwice 模型的性能是最稳定的，特别是当文本长度大于 512 的时候。

本章节还进行了一个案例分析，以进一步比较 NewsQA 上 ThinkTwice 模型的预测结果与其他模型之间的差异。研究结果表明，ThinkTwice 模型的预测答案与真实答案更加接近。

为了验证模型在处理特别长的文本上的性能，本章节选取了两篇长度超过 2,000 词的文章作为样例，如表 3-2 所示。在例子 1 中，模型需要回答奥巴马说了什么，ThinkTwice 准确地定位到包含最终答案的第一段段落，并给出了奥巴马说出的合适内容。然而，BERT 和 ALBERT 却没有像预期那样表现出色，它们分别抽取了其

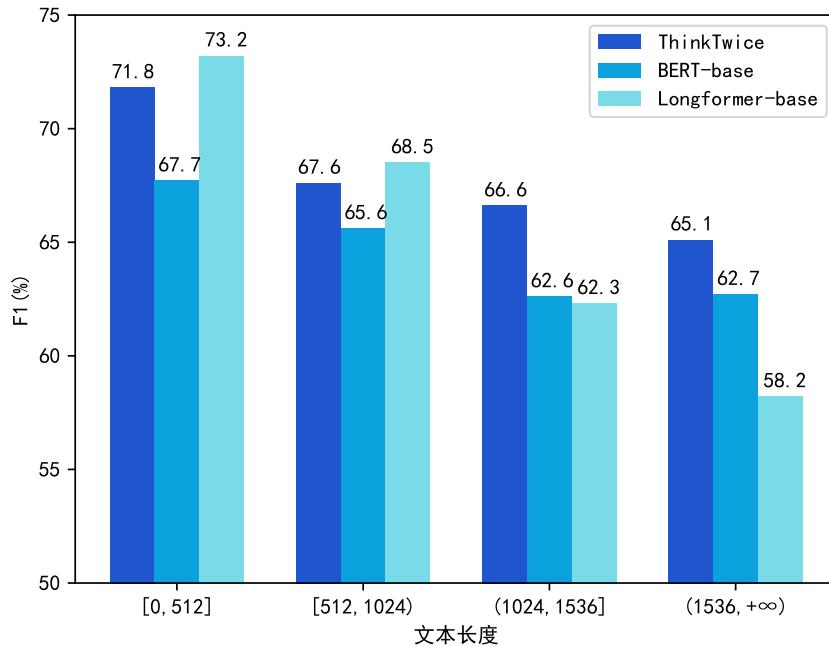


图 3-3 不同文本长度下，ThinkTwice，Bert-base 和 Longformer-base 的性能。四个区间内，样本的数量分别为 1,611, 2,074, 1,089 和 210

他段落的句子。在例子 2 中，ThinkTwice 模型同样准确地定位到了正确的段落，而 BERT 和 Longformer 的表现则不尽如人意。

3.4 本章小结

本章提出了一种二阶段方法，用于在长文本阅读理解任务中解决预训练语言模型中的长度限制问题。ThinkTwice 模型通过将长文本压缩为较短的文本形式，并准确定位答案位置来实现这一目标。实验结果和分析表明，该方法在长文本任务上具有有效性。然而，该方法存在一个潜在的缺陷，即由检索器压缩得到的短文本可能因缺乏先行词而导致不连贯。因此，未来的研究将集中于通过指代消解或位置词嵌入等方式来解决这个问题。

表 3-2 NewsQA 数据集上两个关于三个基线模型和 CoLISA 模型的预测的例子

例子 1
文章:
(CNN) – President Barack Obama spoke with Egypt's president moments after Hosni Mubarak addressed his country, telling the Egyptian that he must make good on his promises and avoid a violent response to the thousands of protesters in the streets. (...)
< 译文: (...) 巴拉克·奥巴马总统在穆巴拉克发表讲话后不久与埃及总统进行了交谈, 告诉埃及总统他必须履行承诺, 避免对街头上万名抗议者采取暴力行动。(...) >
Token 数量: 2,036
问题:
What did Obama say to Mubarak?
< 译文: 奥巴马对穆巴拉克说了什么? >
答案:
he must make good on his promises
< 译文: 他必须遵守诺言 >
CoLISA 预测:
he must make good on his promises and avoid a violent response (True)
< 译文: 他必须遵守诺言并避免采取暴力行动 >
BERT 预测:
I just spoke to him after his speech (False) < 译文: 我在他演讲结束以后跟他讲过 >
ALBERT 预测:
It is very important that people have mechanisms in order (False)
< 译文: 人们拥有机制非常重要 >
Longformer 预测:
he must make good on his promises and avoid a violent response (True)
< 译文: 他必须遵守诺言并避免采取暴力行动 >
例子 2
文章:
(...) Tucked away in the verdant hills west of St. Andrews, Kingarrock Hickory Golf Course (greens fee, \$40 for nine holes and \$55 for 18) is a nine-hole, 2,022-yard country estate course that is played exclusively with antiquated equipment. (...)
< 译文: (...) 坐落在圣安德鲁斯西部郁郁葱葱的山丘上, Kingarrock Hickory 高尔夫球场 (九洞的场地费为 40 美元, 18 洞的场地费为 55 美元) 是一个九洞、2022 码的乡村庄园球场, 球场上全部使用古董设备打球。(...) >
Token 数量: 2,288
问题:
What is Kingarrock Hickory?
< 译文: Kingarrock Hickory 是什么? >
答案:
is a nine-hole, 2,022-yard country estate course that is played exclusively with antiquated equipment
< 译文: 是一个九洞、2022 码的乡村庄园球场, 球场上全部使用古董设备打球 >
CoLISA 预测:
a nine-hole, 2,022-yard country estate course that is played exclusively with antiquated equipment (True)
< 译文: 一个九洞、2022 码的乡村庄园球场, 球场上全部使用古董设备打球 >
BERT 预测:
the kind of place that can change the way one thinks about golf (False)
< 译文: 一种可以改变人们对高尔夫球看法的地方 >
ALBERT 预测:
Golf Course (True) < 译文: 高尔夫球场 >
Longformer 预测:
Top hotel penthouses (False) < 译文: 顶级酒店顶层套房 >

第四章 基于问题分解的多跳长文本阅读理解

多跳机器阅读理解是指利用多个相关文档段落进行多次推理，以实现对复杂问题的理解和回答。与常规的单跳机器阅读理解相比，多跳机器阅读理解需要综合运用文本中的信息、常识和推理能力。通常，多跳阅读理解使用基于图神经网络的方法或基于检索的方法。其中，图卷积网络、图注意力网络和图循环网络通过邻接矩阵建立文档间多个句子或实体之间的联系。基于检索的方法类似于第三章提到的方法，利用文本匹配模型得到与问题相关的文档，然后进行阅读理解。然而，这些模型不擅长寻找支持证据，因为它们缺乏进行真正的多跳推理的能力。目前的多跳阅读理解模型往往是利用快捷方式进行求解，这意味着模型无需实际执行必要的推理步骤即可回答问题。因此，本文提出了两种相关方法 SeqComposer 和 StepComposer，用来将复杂的多跳问题分解为多个简单的单跳问题。这些单跳问题依次检索相关文档作为支持证据，并对这些文档进行阅读理解，以获取每个子问题以及最终的答案。本章在多跳阅读理解数据集 MuSiQue 上进行了实验。实验结果表明，SeqComposer 和 StepComposer 与现有的检索式基线模型相比，在支持证据 F1 的性能上取得了不错的提升。因此，这两种方法可以更好地处理复杂的多跳问题，提高多跳阅读理解的性能。

4.1 引言

多跳阅读理解^[16]（Multi-hop Reading Comprehension, MH-RC）是指需要在多个相关文档段落中进行多次推理，以实现对复杂问题的理解和回答。相较于单跳阅读理解，多跳阅读理解更接近于人类的语言推理能力，具有广泛的应用前景，但也具有极大的挑战性。图 4-1 给出了一个多跳阅读理解数据集 MuSiQue 的例子。例如，对于一个多跳问题 “Who is the president of ... ?”，需要将其分解为四个子问题，每个子问题都不存在多跳的复杂关系。对于每个单跳问题，需要在给定的文档集合中找到对应的文档，并从文档中抽取答案。在该示例中，每个子问题的答案又会重新作用于下一个问题。

在多跳推理中，系统需要利用多个文档中获取的信息进行推理，以得出最终答案。然而，多跳阅读理解不仅需要获取最终答案，还需要系统筛选出可用于回答答案

的支持文档。

表 4-1 MuSiQue 中的例子。

多跳问题：

Who is the president of the newly declared independent country, that established the Timor Leste Commission of Truth and Friendship, with the country containing the airport that includes Lion Air?

< 译文：谁是新宣布独立的成立了帝汶真相与友谊委员会，并包含了包括狮航在内的机场的国家的总统？>

问题分解

子问题 1: What airport is Lion Air part of?

< 译文：狮航隶属于哪个机场？>

答案 1: Juanda International Airport

< 译文：胡安达国际机场 >

子问题 2: #1 » country?

< 译文：答案 1 属于什么国家？>

答案 2: Indonesia

< 译文：印度尼西亚 >

子问题 3: #2 Timor Leste Commission of Truth and Friendship » country?

< 译文：印度尼西亚的帝汶真相与友谊委员会属于什么国家？>

答案 3: East Timor

< 译文：东帝汶 >

子问题 4: who is the president of newly declared independent country #3 ?

< 译文：谁是新宣布独立的国家东帝汶的总统？>

答案 4: Francisco Guterres

< 译文：弗朗西斯科·古特雷斯 >

针对多跳阅读理解，通常可以采用两种不同的方法进行处理。其中一种方法是基于图神经网络，例如 HGN^[80] (Hierarchical Graph Network) 系统。该系统使用一个层次化的图神经网络来执行多跳推理。通过构建一个层次图来聚合来自多个段落的线索，该图由不同粒度级别（问题、段落、句子、实体）的节点构成。该系统的表示形式使用预训练的 BERT 模型初始化。另一种方法是基于检索的方法，例如 SAE^[81] (Select, Answer and Explain) 系统。该系统通过使用一个可解释的模型来选择最相关的文档，然后在这些文档中执行多跳推理，以回答问题。

然而，这两种方法都存在一些问题。尽管图神经网络能够直接找到答案，但其采

用的黑盒模型无法提供充分的证据支持，因为它可以在不执行必要的推理步骤的情况下回答问题。相比之下，检索的方式则采用了长文本阅读理解的做法，但未能充分利用多跳的特点。

因此，本章使用人类回答多跳问题的方法，提出了一种基于问题分解的方法，将复杂的多跳问题分解为多个简单的单跳问题。对于这些单跳问题，我们依次从长文本中检索出相关文档作为支持证据，并对这些文档进行阅读理解，以获取每个子问题的答案和最终答案。本章在 MuSiQue 多跳阅读理解数据集上对两种问题分解方法 SeqComposer 和 StepComposer 进行了实验。实验结果显示，相较于现有的检索式基线模型，本章提出的方法在支持证据 F1 评价指标上有显著提升。

本章的主要贡献如下：

- 针对多跳阅读理解领域，本章采用问题分解的基本思路，提出了 SeqComposer 和 StepComposer 两种方法。这些方法通过将复杂的多跳问题分解为多个简单的单跳问题，更好地适应了阅读理解任务。
- 本章采用生成式模型对抽取式阅读理解任务中的问题分解部分进行评价，并评估了多跳问题的分解效果。
- 本章针对多跳阅读理解数据集 MuSiQue，展示了 SeqComposer 和 StepComposer 方法相对于检索式基线模型在支持证据 F1 上的明显提升。

4.2 基于问题分解的多跳长文本阅读理解

本节提出了一个基于问题分解来完成多跳阅读理解的模型架构 SeqComposer。本节将按照 SeqComposer 的问题分解部分，检索与阅读理解部分，以及对其进行的改善三方面进行逐一展开。

4.2.1 总体架构

图 4-1展示了本章提出的 SeqComposer 模型的结构，它由三个主要组成部分构成。首先，问题分解模块使用流行的序列到序列生成网络将复杂的多跳问题分解为多个简单的单跳问题。然后，基于神经网络的检索模型和阅读理解模型将生成模型得到的单跳问题与文档集合并，得出问题的答案，并将答案与下一个问题进行充分的结合，得到下一步骤的完整表示。最后，本文对 SeqComposer 模型进行了改进，得

到了新的 StepDecomposer 架构，该架构多次执行问题分解过程，并将生成结果传递给检索和阅读模型。

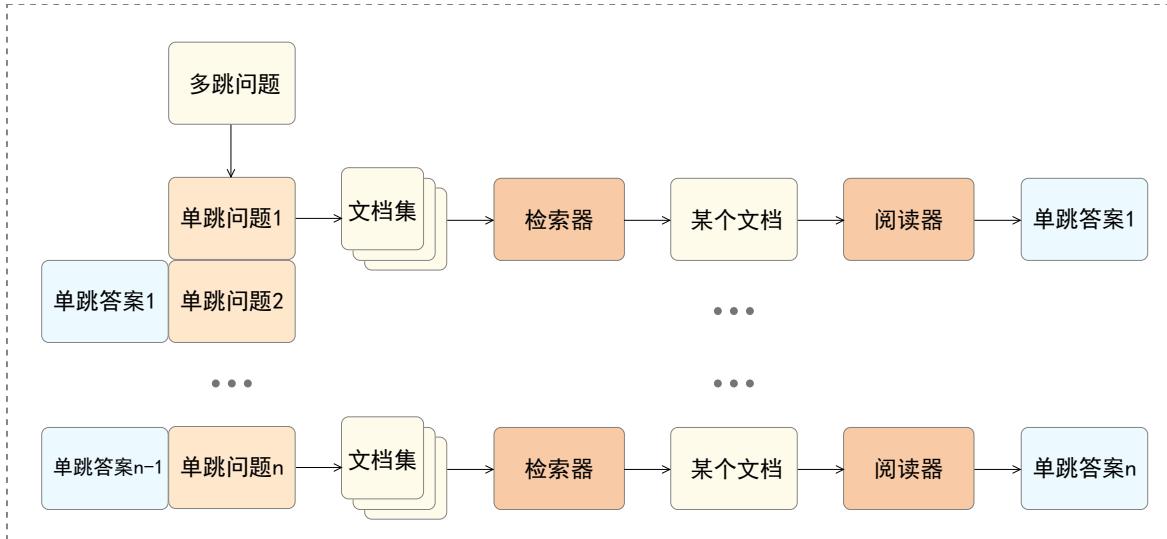


图 4-1 SeqDecomposer 的整体架构

4.2.2 基于序列到序列生成的问题分解模块

问题分解技术是一种将复杂多跳问题分解成多个简单单跳问题的方法。这种方法的优点在于简化了问题的回答过程，使得回答过程更加可控和可解释。

本文提出了 SeqDecomposer 架构，该架构采用序列到序列的生成模型，对于多跳问题 Q ，将其分解为多个单跳问题 q_1, q_2, \dots, q_n 。这个过程可以表示为：

$$q_1, q_2, \dots, q_n = GEN(Q) \quad (4.1)$$

序列到序列的生成模型通常包括 BART 和 T5 等。BART^[62] (Bidirectional and Auto-Regressive Transformer) 是 Facebook AI Research 在 2019 年提出的一种序列到序列的生成模型，具有自编码和自回归的能力，能够在不同的任务中共享参数。而 T5^[63] (Text-to-Text Transfer Transformer) 是 Google 在 2020 年提出的一种序列到序列的生成模型，可以将所有的自然语言处理任务都转化为文本到文本的转换任务，从而用同一种模型解决不同的任务。在本文中，同时使用了 BART 和 T5 来进行相关实验比较。需要注意的是，从生成第二个及以后的子问题 $q_i (i \geq 2)$ 开始，由于 q_i 可能需要前一个子问题 q_{i-1} 的答案 a_{i-1} ，因此在生成子问题 q_i 时，需要先将先前的答案 a_{i-1} 以词槽的形式填充到子问题 q_i 中。直到整个系统得到最后一个子问题 q_n 的答案 a_n

后，将停止迭代。

4.2.3 基于预训练语言模型的阅读理解模型

在多跳阅读理解中，通常会以多文档的形式提供参考文本。这种形式导致文本总长度通常会非常长。因此，与第三章类似，本章需要针对每个单跳问题进行文档检索和阅读的操作。

针对单跳问题 q_i ，需要先进行检索，从备选文档中找到相关文档。在 MuSiQue 数据集中，大部分问题都分配到 20 个文档，记为 $D = D_1, D_2, \dots, D_{20}$ ，这 20 个文档只有一个与 q_i 相关。为了进行检索，可以将子问题 q_i 与所有文档依次拼接，形成 $x_{ij} = [CLS]q_i[SEP]D_j[SEP]$ 。通过一个预训练语言模型，可以得到 x_{ij} 的整体表示 H_{ij} 。接着，使用一层全连接网络和 Softmax 层计算 H_{ij} 的分类概率值 \hat{y}_c ，表示子问题 q_i 与文档 D_j 的相关性。损失函数可以计算 \hat{y}_{ij} 与子问题 q_i 对应的正确文档标签 y_{ij} 之间的差距，以监督检索模型的学习。其中，检索出与子问题 q_i 相关的文档的交叉熵损失函数可以表示为：

$$\mathcal{L}_i^{Retr} = -y_{ij} \log \hat{y}_{ij} + (1 - y_{ij}) \log (1 - \hat{y}_{ij}) \quad (4.2)$$

在得到子问题 q_i 相关的文档 D_j 后，阅读模块需要将两者的表示拼接在一起，得到 $x_i = [CLS]q_i[SEP]D_j[SEP]$ 。然后，阅读模块中的编码器将 x_i 转化成隐向量；该隐向量经过一个指针网络，可以标识出文档中每个 token 作为答案起始位置和结束位置的概率值，分别用 \hat{y}_s 和 \hat{y}_e 来表示。训练过程中，用真实答案的起始和结束位置 y_s 和 y_e 来进行训练，其损失函数可以表示为：

$$\mathcal{L}_i^{Ans} = \frac{1}{2} \text{CrossEntropy}(\hat{y}_s, y_s) + \frac{1}{2} \text{CrossEntropy}(\hat{y}_e, y_e) \quad (4.3)$$

在推理阶段，若上述模型获取的子答案 a_i 不是最后一跳问题的答案，需将其填充至下一个单跳问题 q_{i+1} 中的槽位，以完成下一跳问题的形成。这样，下一跳问题 q_{i+1} 即成为了一个完整的自然语言问题，可以不断循环执行检索与阅读的步骤，直到获取最终答案 a_n 为止。

4.2.4 采用分步执行的问题分解技术

在前文中提到的 SeqComposer 架构中，使用了序列到序列的生成模型，将复杂的多跳问题一次性分解为了一系列单跳问题。虽然用占位符填充子问题可以使每个问题看起来具备连贯性，但本文认为在生成子问题的过程中，没有充分利用可用的答案信息。

为了改进 SeqComposer 的工作流程，本节提出了一种新的架构 StepComposer，该架构针对问题分解部分进行了修改。在处理复杂的多跳问题 Q 时，StepComposer 的方法是每次只让生成模型生成一个单跳问题 q_i ，与 SeqComposer 相比，其具体实现方式有所不同。具体实现过程如图 4-2 所示，对于问题 Q 以及已经生成的前 $k - 1$ 跳单跳问题 q_1, q_2, \dots, q_{k-1} ，这些单跳问题已经通过前述的检索和阅读理解模型得到了对应的子答案 a_1, a_2, \dots, a_{k-1} 。此时，将这些已知信息，包括原始问题 Q 、单跳问题 q_1, q_2, \dots, q_{k-1} 以及单跳答案 a_1, a_2, \dots, a_{k-1} 输入到生成式模型中，目的是生成下一跳问题 q_k ，即：

$$q_k = GEN(Q, q_1, a_1, q_2, a_2, \dots, q_{k-1}, a_{k-1}) \quad (4.4)$$

完成最后一跳问题 q_n 的生成后，需要使用检索模型和阅读理解模型得到最终答案，以完成整个流程。需要注意的是，现实应用中多跳问题的跳数是未知的。因此，本文针对 QuALITY 训练集中的最后一跳问题添加了结束标记，以便在生成式模型生成带有结束标记的问题时，能够判断此时已经到达了最后一跳问题。

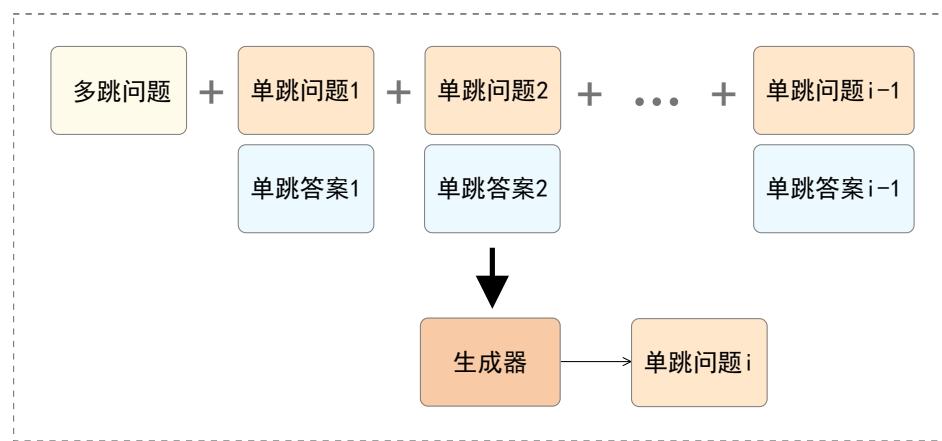


图 4-2 StepComposer 的问题分解过程

4.3 实验及结果分析

4.3.1 实验设置

本章进行了大量实验，主要在多跳多文档阅读理解数据集 MuSiQue 上进行。MuSiQue 的文章主要来源于英文维基百科，是对一些经典的阅读理解数据集（如 SQuAD 等）的重构。该数据集中对于每个问题所给出的相关文档数量要远远大于其他多跳数据集，其中 99% 以上的问题都给出了 20 个可能相关的文档，还有少部分问题是给出了 16-19 个文档。MuSiQue 包含了 2-4 跳总共约 25k 个问题，其中 2 跳问题占据了绝大多数。训练集包含了 20k 个问题。与以往的多跳阅读理解数据集不同，MuSiQue 往往需要给出正确的多跳支持证据才能找到最终答案。本文还统计了 MuSiQue 中问题的长度分布，训练集中多跳问题的 token 数量中位数为 18，经过训练好的生成模型将其分解为多个单跳问题后，拼接后的句子的 token 数量的中位数为 28。

本章采用了选择器 + 问答器架构作为基线模型。该架构首先对备选文档进行排序，筛选出 K 个最相关的文档 D_K ¹。具体而言，对于给定的问题 Q ，需要判断每个文档 D_i 与 Q 是否相关，训练时使用交叉熵损失。随后，回答器基于筛选出的文档 D_K 来预测答案及其支持证据。选择器使用了 RoBERTa-large 作为预训练语言模型，而回答器则使用了 Longformer-large 作为预训练语言模型。在筛选出 K 个相关文档后，回答器使用这些文档来生成答案和支持证据。

本章主要使用支持证据 F1 和答案 F1 来衡量多跳机器阅读理解模型的性能。

本章基于问题分解技术提出了两个模型 SeqDecomposer 和 StepDecomposer。在问题分解阶段，本研究使用 BART-large^[62] 和 T5-large^[63] 进行大部分实验。在问答阶段，本研究主要使用 ALBERT-xxlarge^[72] 和 DeBERTaV3-large 作为预训练语言模型。在训练阶段，所有模型的学习率均为 1e-5。在生成阶段，BART 模型的 batch size 为 64，epochs 为 2，T5 模型的 batch size 为 16，epochs 为 2。在问答阶段，DeBERTa 的 batch size 为 2，epochs 为 3，ALBERT 的 batch size 为 1，epochs 为 2。

¹ K 是一个超参，通常设定为 3, 5, 或 7

4.3.2 实验结果和分析

表 4-2 记录了 SeqDecomposer 的问题分解部分在三个生成指标上的表现。从表中可以看出，BART-large 由于其本身的预训练方式，非常适合做序列到序列生成的任务，因此其在问题分解任务中的表现要优于 T5。因此，本节剩余的关于问题分解的实验都会基于 BART-large 进行。

表 4-2 SeqDecomposer 的问题分解评价指标。

	BLEU	METEOR	ROUGE
BART-large	49.4	74.2	53.0
T5-large	49.0	73.2	53.4

此外，本文针对 2 跳、3 跳和 4 跳的问题，分别统计了各自的生成指标，如表 4-3 所示。不同跳数之间存在明显差异，一方面是因为更高跳数的问题分解难度更大，另一方面是对应的训练数据更少。由于 StepDecomposer 的分解方式是在得到了上一问的答案的基础上再进行生成，因此，本文采用将所有单跳问题拼接的方法，评估对应的生成指标。对比结果如表 4-4 所示。可以看出，StepDecomposer 的生成指标略低于 SeqDecomposer，原因是融合了问答模型中抽取的一些错误答案。

表 4-3 使用 BART-large 的 SeqDecomposer 在 2,3,4 跳问题分解上的评价指标

	BLEU	METEOR	ROUGE
2 跳	53.4	77.7	56.0
3 跳	47.3	72.8	51.6
4 跳	40.7	66.1	46.3

表 4-4 SeqDecomposer 和 StepDecomposer 的问题分解评价指标对比

	BLEU	METEOR	ROUGE
SeqDecomposer	49.4	74.2	53.0
StepDecomposer	48.9	71.9	55.5

接下来，本文按照与基线模型相同的方式，训练一个检索器。对于每个子问题，数据集给定了相应的唯一一个支持文档。本文的工作在于判断问题与每个文档是否对应，通过计算交叉熵损失，得到得分最高的文档，作为支持文档。其实验数据如表 4-5 所示。表格中的预训练语言模型都可以做到对文档的高质量检索，即便是参

数量更高的模型，相对于 BERT-base 也不会有明显差异。表中两列数据的差异在于，是否为文档加上标题。结果是加标题与否，对最终的性能没有明显影响。

接下来，本文将按照与基线模型相同的方式训练一个检索器。对于每个子问题，数据集给定了一个相应的唯一支持文档。本文的工作在于判断问题与每个文档是否对应，并通过计算交叉熵损失得到得分最高的文档作为支持文档。实验结果如表 4-5 所示。表格中的预训练语言模型都可以实现高质量的文档检索，即便是参数量更高的模型，与 BERT-base 相比也没有明显差异。表中两列数据的差异在于是否为文档加上标题。实验结果表明，加标题与否对最终的性能没有明显影响。

表 4-5 检索器的评价指标

	ACC（无标题）	ACC（有标题）
BERT-base	96.3	96.6
BERT-large	96.8	97.1
BERT-base	97.4	<u>97.4</u>
BERT-base	<u>97.5</u>	97.2

此外，本文需要训练一个单跳阅读器。该阅读器能够在给定单跳问题和支持文档的情况下，利用阅读理解技术从文档中提取答案。实验结果如表 4-6 所示，不同规格的模型的性能表现存在较大差异。因此，本文的主要实验采用了 DeBERTa 和 ALBERT 这两个更优的预训练语言模型。

表 4-6 阅读器的评价指标

	F1（无标题）	F1（有标题）	EM（无标题）	EM（有标题）
BERT-base	64.3	61.5	57.6	54.3
BERT-large	73.1	72.0	65.6	64.1
DeBERTaV3-large	81.9	82.5	73.5	73.8
ALBERT-xxlarge	<u>84.5</u>	<u>84.3</u>	<u>77.0</u>	<u>76.0</u>

表 4-7 展示了使用 BART-large 模型进行问题分解，结合 DeBERTaV3-large 和 ALBERT-xxlarge 两种模型进行检索与问答阶段的实验结果。实验结果表明了三个现象。首先，相对于 SeqComposer，StepComposer 在两个评价指标上都略微提升，这表明本文针对问题分解方式的改进取得了一定效果。由于 StepComposer 存在错误答案积累的情况，这也许是导致该方法并没有特别显著提升的原因。其次，在检索和问答阶段中，无论采用 DeBERTaV3 还是 ALBERT 的架构，都没有表现出明显的

性能差异。这表明，模型的整体架构起着关键作用，而不是单个环节的模型本身的能力。最后，在支持证据的 F1 指标方面，与选择器和回答器的基线模型相比，本章提出的 SeqComposer 和 StepComposer 均有显著的提升，而在答案的 F1 指标方面略有下降。其中，SeqComposer 在支持证据的 F1 方面的提升最高达 6.7，最低为 5.0，在答案的 F1 方面的下降最高达 3.7，最低为 1.5。StepComposer 在支持证据的 F1 方面的提升最高达 7.9，最低为 7.5，在答案的 F1 方面的下降最高达 2.2，最低为 1.5。这与模型的初衷相符。无论是 SeqComposer 还是 StepComposer，都希望通过将多跳问题分解为单跳问题的形式，找到每一跳问题对应的支持文档，因此它们的支持文档的 F1 值会更高。而基线模型中提出的选择器和回答器的模式偏向于直接将多跳问题和备选文档输入神经网络，从而快速获得答案，因此该方法更倾向于答案抽取。

表 4-7 SeqComposer 和 StepComposer 的实验性能

模型	支持 F1	支持 EM	答案 F1	答案 EM
基线				
RoBERTa-large & Longformer-large	75.2	-	<u>52.3</u>	-
SeqComposer 模型				
DeBERTaV3 & DeBERTaV3	80.2	49.9	49.6	39.8
DeBERTaV3 & ALBERT	80.5	50.7	48.6	38.1
ALBERT & DeBERTaV3	81.8	51.1	<u>50.8</u>	<u>41.1</u>
ALBERT & ALBERT	<u>81.9</u>	<u>51.8</u>	49.7	39.6
StepComposer 模型				
DeBERTaV3 & DeBERTaV3	82.7	56.1	50.7	<u>42.1</u>
DeBERTaV3 & ALBERT	83.0	<u>57.8</u>	<u>50.8</u>	40.3
ALBERT & DeBERTaV3	83.0	56.7	50.2	41.6
ALBERT & ALBERT	<u>83.1</u>	56.3	50.1	39.9

4.4 本章小结

本章提出了一种基于问题分解技术，用以处理多跳阅读理解任务的方法，并改善模型搜寻支持证据的能力。该方法包括 SeqComposer 和 StepComposer 两种类型，两者都将复杂的多跳问题分解为多个简单的单跳问题，然后依次完成检索和阅读理解任务。实验结果和分析表明，该方法在捕获证据文档的能力方面有显著的提升。

不过，本章还未对模型进行全方位的探索。因此，未来的研究将考虑应用大型模型中思维链的方式，对问题分解进行更细致的研究。

第五章 基于对比学习的多项选择长文本阅读理解

多项选择阅读理解是一种通过阅读并理解一篇给定文章和问题，从多个备选答案中选出最合适答案的任务。近期的研究致力于捕获文章、问题和选项三元组之间的联系，但由于文本长度过长，模型难以重点关注备选答案之间的联系，导致在面临复杂的多项选择问题时，最近的方法往往会将干扰选项误判为正确答案。为了解决这一问题，本章提出了一种基于对比学习和样本内注意力的模型 CoLISA (Contrastive Learning and In-Sample Attention)，该模型通过对对比学习和样本内注意力机制，相对精准地剔除干扰选项。特别地，CoLISA 采用对比学习来获取蕴含其他选项信息的选项表示，并应用样本内注意力机制，使得多个选项之间产生联系。实验结果表明，CoLISA 在正确与错误选项之间的联系花费了更多注意力，能够识别选项之间的差异，并在 QuALITY 数据集上实现了 SOTA (state-of-the-art) 的性能。

5.1 引言

机器阅读理解 (Machine Reading Comprehension, MRC) 是一种通过对给定文档进行推理，要求模型回答指定任务的任务。作为 MRC 任务的一种变体，多项选择阅读理解 (Multi-choice Reading Comprehension, MC-RC) 主要针对在给定文章的情况下，从多个备选答案中选出最合适的答案来回答问题。MC-RC 要求模型在阅读并理解参考文章的前提下，判断每个备选答案的正确性。

在 MC-RC 领域，现有的研究通常集中于解决文章和备选答案之间的差异问题，以解决给定问题^[82,83]。这些模型通常会对每个选项进行独立编码。然而，这种方法会限制模型的推理能力，因为对于某个问题，每个选项并不能直观地与其他选项进行交互。此外，在某些真实情况下，某些备选答案在字面上或语义上与正确答案非常相似，这使得仅仅判断单个选项的正确性变得更加困难。目前已有的方法往往无法处理这些情况。因此，本章认为需要采用更精细的方法来处理这些所谓的干扰项与正确答案之间的关系。表 5-1 具体描述了 QuALITY^[84] 中一个关于难以辨别的干扰项的实例¹。参考例子中的整篇文章，可以认为正确答案 O_2 (表中红色部分) 和干扰选项 O_1 (表中蓝色部分) 可以认定都是接近于正确答案的。为了根据给定问题选出最合适

¹QuALITY 中也存在输入文章长度超过模型限制的情况，本章也会在后续讨论这个问题。

答案，模型需要找到正确答案和干扰选项表示之间的差异。

文章：

(...) I'm sure that 'justifiable yearnings for territorial self-realization' would be more appropriate to the situation (...) (Over 4,000 words)

< 译文：(...) 我确信“对领土自我实现的正当渴望”可能更适用于这种情况 (...) >

问题：

According to Retief what would happen if the Corps did not get involved in the dispute between the Boyars and the Aga Kagans?

< 译文：如果 Corps 不介入 Boyars 和 Aga Kagans 的争端，根据 Retief 的说法会发生什么？>

选项：

O_1 : The Aga Kagans would enslave the Boyars

< 译文：Aga Kagans 会奴役 Boyars>

O_2 : The Boyars and the Aga Kagans would go to war

< 译文：Boyers 和 Aga Kagans 会引发战争 >

O_3 : The Aga Kagans would leave Flamme to find a better planet

< 译文：Aga Kagans 会离开 Flamme，找到更好的星球 >

O_4 : The Boyars would create a treaty with the Aga Kagans without the Corps' approval

< 译文：Boyers 会与 Aga Kagans 签订条约，而不需要得到 Corps 的批准 >

表 5-1 QuALITY 中的例子。

人类总是需要通过仔细比较各个选项之间的差异，来排除看上去正确的错误选项，从而回答阅读理解-多项选择问题（MC-RC 问题）^[85]。受到该流程的启发，本章提出了一个基于对比学习和样本内注意力（CoLISA）的架构，它包含两个主要的特点。首先，由于应用了两次不同的 dropout 掩码，得到两个有着轻微差异的表示。也就是说，对于同一个输入，会有两组输出值，每一组都包含了一个正确答案和若干个错误选项。CoLISA 主要通过对比学习的方式，将两个正确选项的表示拉近，然后把正确答案与干扰选项之间的表示推远；因此，该模型可以学习到更有效的文本表示。此外，自注意力机制^[36] 也应用在了特定样本中的多个选项之间，在选项之间共享彼此的信息。因此，模型借助多个备选答案之间的自注意力交互的方式，学习到了倾向于选择正确答案的能力。本章还在两个 MC-RC 数据集 QuALITY^[84] 和 RACE^[15] 上进行了实验。实验结果表明，CoLISA 的性能表现显著超过其他现有方法。

本章的研究贡献可以概括如下：

- 引入对比学习方法，以区分多项选择长文本阅读理解任务中正确答案和干扰选

项之间的差异。该方法通过赋予干扰选项更多的权重，从而使其获得更多的注意力。

- 针对一个特定的例子，将样本内注意力机制应用于多个选项之间，促进它们之间的互动。
- 本章提出的 CoLISA 方法在 QuALITY 上实现了 SOTA 的性能，并在 RACE 上相对于其他基线模型实现了显著的提升。

5.2 基于对比学习的多项选择长文本阅读理解

给定一篇参考文本，一个目标问题，还有一些备选答案，多项选择阅读理解 (multiple choice reading comprehension, MC-RC) 任务是指将其中一个选项预测为最终答案。形式上来说，将一篇文章定义为 $p = [s_1^p, s_2^p, \dots, s_n^p]$ ，其中 $s_i^p = [w_1^s, w_2^s, \dots, w_l^s]$ 表示 p 中第 i 个句子， w_j^s 表示 s_i^p 中的第 j 个词。问题可以定义为 $q = [w_1^q, w_2^q, \dots, w_l^q]$ ，其中 w_i^q 表示 q 中的第 i 个词。选项集合可以定义为 $O = [o_1, o_2, \dots, o_r]$ ，其中 $o_i = [w_1^o, w_2^o, \dots, w_k^o]$ 表示第 i 个选项， w_k^o 表示 o_i 中第 k 个词。MC-RC 的目标是最大化所预测选项的概率：

$$a = \underset{i}{\operatorname{argmax}}(\mathcal{P}(o_i|p, q)) \quad (5.1)$$

当 p 的长度超过编码器的最大输入长度时，本章的做法是通过检索与问题和备选答案相关的句子，将长文本压缩成篇幅较短的文本。较短文本表示为 $c = [s_1^c, s_2^c, \dots, s_m^c]$ ，其中 $s_i^c = [w_1^s, w_2^s, \dots, w_l^s]$ 表示 c 中的第 i 个句子， w_j^s 表示 s_i^c 中的第 i 个句子。

5.2.1 总体架构

如图 5-1 所示，本章提出了一种新颖的 MC-RC 框架，它充分运用了对比学习和样本内注意力 (Contrastive Learning and In-Sample Attention, CoLISA)，主要由两部分组成：首先是基于 DPR (Dense Passage Retrieval，稠密检索) 的检索器，在一篇很长的文章中，根据对给定问题和对应的过个备选答案之间的相关性，筛选出相关的句子，以此来根据原文的原始顺序来构建一段新的文本；另外，CoLISA 阅读器需要根据给定问题和上下文中，从若干选答案中预测出最终答案。CoLISA 阅读器由两部分组成。第一部分是样本内注意力 (In-Sample Attention, ISA) 机制，换句话说，引入一个加入了多头自注意力机制并针对长序列的网络，来增强特定样本中多个选

项之间的交互。第二部分是对比学习（Contrastive Learning, CoL）以及一个干扰因子（Distractive Factor, DiF），来表示由上下文、问题和备选答案组合而成的序列。

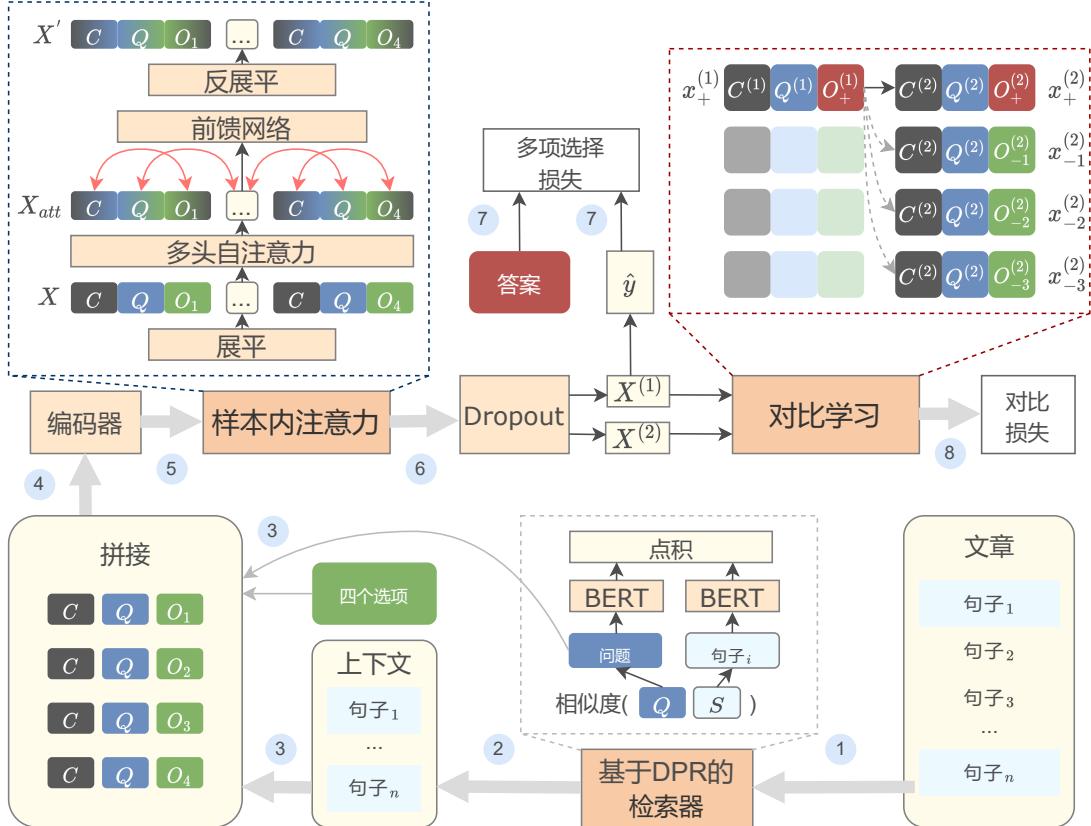


图 5-1 CoLISA 的基本架构。

5.2.2 基于 DPR 的检索器

为了从一篇长文本中筛选出相关的句子，本节首先采用基于 DPR (Dense Passage Retrieval, 稠密检索) 的句子检索器。该检索器常用于隐式语义编码^[86]。值得注意的是，DPR 提供了两个不同的编码器，用于预训练抽取不同种类的句子，本节使用相应的 DPR 编码器以确保检索到的句子具有多样性。上下文编码器 \$E_S\$ 将参考文章 \$p\$ 中的所有句子 \$s\$ 编码为 \$d\$ 维向量。类似地，查询编码器 \$E_R\$ 将问题 \$q\$ 和选项集合 \$O\$ 编码为 \$d\$ 维向量，作为两种不同的检索查询 \$r\$。\$s\$ 和 \$r\$ 的 [CLS] token 的全局表示用于计算它们的负欧几里得距离 (\$L^2\$)：

$$- L_{dist}^2(r, s) = - \|E_R(r) - E_S(s)\|^2 \quad (5.2)$$

针对选项查询，本节按照 s 和 r 之间的相关性降序排列，选择前 k 个句子，并考虑到语义连贯性的问题，筛选出这 k 个句子的前一句和后一句。然而，同一个句子可能会被多次选择，因此需要去重。最终得到 k 个唯一的句子作为参考上下文。

对于问题查询，首先选择前 n 个句子作为参考上下文²。由于来源于选项的证据相比于来源于问题的证据更合适，所以本节不采集前一句和后一句。同样，需要通过去重来确保所有抽取句子的唯一性。

在选出最合适的句子后，再根据原文顺序将这些句子进行排序，然后将它们拼接成一段参考上下文 c 。算法 1 详细阐述了抽取的过程。

算法 1 长文本抽取算法

输入: 文章 $p = [s_1, s_2, \dots, s_n]$, 问题 q , 选项 $O = [o_1, o_2, \dots, o_m]$

输出: 上下文选择编码器

```

if  $x \in p$  then
     $E_x = ContextEncoder(x)$ 
else
     $E_x = QuestionEncoder(x)$ 
end if
for  $o_i \in O, s_j \in p$  do
     $sim(o_i, s_j) = -L_{dist}^2(E_{o_i}, E_{s_j})$ 
end for
选择: 前  $k$  个相关的句子  $s_{i_1}, s_{i_2}, \dots, s_{i_k}$ 
选择: 它们的前一句和后一句  $s_{i_1}^{prev}, s_{i_1}^{next}, \dots, s_{i_k}^{prev}, s_{i_k}^{next}$ 
上下文  $\leftarrow$  被选择的句子
for  $s_j \in p$  do
     $sim(q, s_j) = -L_{dist}^2(E_q, E_{s_j})$ 
end for
选择: 前  $n$  个相关的句子  $s_{j_1}, s_{j_2}, \dots, s_{j_n}$ 
上下文  $\leftarrow$  被选择的句子
上下文. 去重 (). 排序 (). 字符串化 ()
```

5.2.3 样本内的自注意力机制

在进行编码 q 和 o_i 时，针对同一个问题，每个备选答案的编码都是相互独立的，这导致在计算注意力时可能会出现注意力缺失的问题。为了解决这个问题，本章借鉴了人类回答多项选择问题的方式，即通过同时比较多个选项来排除干扰项，最终决定选择哪个选项。基于这种思路，本章提出了一种样本内注意力（In-Sample Attention）

²考虑到预训练语言模型能容纳所抽取的句子的总长度， k 和 n 需要设定为合适的值，这里在所有实验中将 k 设为 2, n 设为 1。

机制，以增强不同选项表示之间的交互作用。

通常，每个备选答案 o_i 先和它对应的上下文 c 以及问题 q 进行拼接，构成一个三元组 $x_i = [c; q; o_i]$ 。本章通过将每个 x_i 依次喂入预训练编码器中，从而得到一系列完整的序列表示。为了建模多个选项 o_i 之间的联系，本章提出的 ISA 模块首先收集到所有与相同问题 q 对应的 x_i ，对它们进行拼接，构建一个单独的序列 $X = [x_1; x_2; \dots; x_n]$ 。然后，计算 X 的自注意力表示，来学习多个选项之间的远程依赖。具体来说，利用平凡的自注意力机制的架构，同时将序列 X 分解为三个矩阵 Q, K, V 。输出的自注意力矩阵可以计算为：

$$SA(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (5.3)$$

其中 d_k 是一个缩放因子，它表示 K 的维度，以防引起梯度消失现象^[36]。更进一步，多头注意力机制被引入到 ISA 中，从不同的向量维度来更加综合性的表示 X 。多头自注意力的流程可以定义为：

$$\text{head}_i = SA(QW_i^Q, KW_i^K, VW_i^V) \quad (5.4)$$

$$H = \text{Concat}(\text{head}_1, \dots, \text{head}_h) \quad (5.5)$$

$$MSA(Q, K, V) = HW^O \quad (5.6)$$

其中 h 表示各个平行注意力头的数量， W_i^Q, W_i^K, W_i^V, W^O 是注意力参数矩阵。这里还将多头自注意力的向量表示记为 X_{att} 。如图 5-1 中的 ISA 机制所描述的，作用在拼接后的序列 X 的注意力机制通过计算选项之间的表示 X_{att} ，实现彼此之间的交互。

此外，为了避免多头自注意力输出的塌陷^[36,87]，多头自注意力机制后面还增加了一个全连接网络：

$$X_{fnn} = FFN(X_{att}) \quad (5.7)$$

这里使用了 GeLU^[88] 函数作为激活函数。最终， X_{fnn} 需要根据多个输入三元组 x_i 的维度拆解为多个表示。输出三元组的结合记为 X' 。

5.2.4 面向备选答案交互的对比学习方法

为了推动 CoLISA 在明确区分正确答案和干扰选项表示之间的差异方面的表现，本节引入了对比学习（Contrastive Learning, CoL）模块。受通用对比学习框架^[64] 的

启发, CoL 模块旨在通过将同一样本内的所有表示均匀分布在特定的向量空间上, 促进两个正向三元组表示之间的变得更加接近。

在上一节介绍的 ISA 模块之后, 具有交互注意力的输入向量 X' 通过 dropout 层进行传递, 并进行两次相同的操作, 以针对每个输入生成两个略有差异的编码表示, 分别记为 $X^{(1)}$ 和 $X^{(2)}$ 。

之后, 先要计算 MC-RC 任务中在目标标签 y 和输出 $X^{(1)}$ 之间的交叉熵损失。具体来说, 通过一层全连接神经网络可以将 $X^{(1)}$ 转换为预测输出 \hat{y} , 它的维度与标签 y 相同。损失函数定义如下:

$$\mathcal{L}_{MC-RC} = -\frac{1}{N} \sum_{i=1}^N (y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})) \quad (5.8)$$

注意到来自同一样本的输出 $X^{(1)}$ 和 $X^{(2)}$ 均由相同数量的三元组 x_i 组成, 其中包含一个正确答案的三元组以及其余包含错误答案的三元组。因此, 对于每个输入样本的两个输出结果, 对比损失可以被定义为负对数似然函数的平均值。具体而言, 对于其中一个输出 $X^{(1)}$, 本节将包含正确答案的三元组作为锚点, 并移除包含错误选项的三元组。对于另一个输出 $X^{(2)}$, 所有三元组均被保留, 其中包含正确选项的三元组视为正例, 而其他三元组则被视为负例。每个损失项均能够区分正例和负例之间的差异。

CoL 的损失函数定义如下:

$$\mathcal{L}_{CoL} = -\frac{1}{N} \sum_{j=1}^N \log \frac{e^{\text{sim}(x_+^{(1)}, x_+^{(2)})/\tau}}{\sum_{i=1}^S e^{\text{sim}(x_+^{(1)}, x_i^{(2)})/\tau}} \quad (5.9)$$

其中 $x_+^{(1)}$ 是 $X^{(1)}$ 中包含正确答案的三元组的编码表示, 而 $x_i^{(2)}$ 是 $X^{(2)}$ 中所有三元组的表示, $X^{(2)}$ 中的 $x_+^{(2)}$ 是正例样本, τ 是可配置的超参数, 温度系数。 $\text{sim}(\cdot)$ 是一种相似度指标 (本章中所有的实验都是使用余弦相似度函数), S 代表了每个样本中三元组的数目, N 代表了 batch 的大小。聚合后的对比损失 \mathcal{L}_{CoL} 主要通过在每个 batch 中的所有样本取平均获得。

干扰因子。基于样本中不同的干扰项会对模型造成不同程度的干扰, 本节还将一个对比因子 (Distractive Factor, DiF) 引入到了 CoLISA 模型, 具体来说, 将 DiF 结合到对比学习的过程当中。QuALITY 的数据标注人员针对每个问题的所有错误选项中, 把误导他们最严重的的那个标记为强干扰项。因此, 本节中通过构建一组置信因子, 根据每个选项对应的标注分数, 来代表它们对 \mathcal{L}_{CoL} 的贡献度。本节中枚举了对

每个选项的标注票数来构建 DiF $\Theta = [\theta_1, \theta_2, \dots, \theta_n]$, 其中 n 是备选答案的数量。然后再用一个 softmax 函数来帮助缩放 Θ , 从而更明显的区分每个 θ_i 。每个 θ_i 被修正为如下的形式:

$$\theta_i = \frac{e^{\theta_i}}{\sum_i^n e^{\theta_i}} \quad (5.10)$$

幂运算可以清晰的划分 n 个系数 θ_i 。当计算对比损失的时候, θ_i 乘上了它对应选项的相似度的值, 来衡量正确答案和干扰答案的差距。 θ_i 的值越大, 对对应选项贡献的损失就越多。经过上述分析, 对比损失函数可以改进为如下:

$$\mathcal{L}_{CoL} = -\frac{1}{N} \sum_{j=1}^N \log \frac{\theta_+ e^{sim(x_+^{(1)}, x_+^{(2)})/\tau}}{\sum_{i=1}^S \theta_i e^{sim(x_+^{(1)}, x_i^{(2)})/\tau}} \quad (5.11)$$

其中 θ_+ 和 θ_i 分别对应样本中的正例以及第 i 个选项。最终的损失函数可以表示为:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{MC-RC} + (1 - \alpha) \cdot \mathcal{L}_{CoL} \quad (5.12)$$

其中 $\alpha \in [0, 1]$ 是一个平衡系数。

5.3 实验及结果分析

本节首先详细描述了实验的设置, 并在此基础上呈现了实验结果。接着, 本节对实验结果进行了进一步的对比和分析, 特别是与其他现有模型进行了比较。此外, 本节还进行了一系列消融实验, 以验证 CoLISA 模型的有效性和鲁棒性。

5.3.1 实验设置

本节采用了基准数据集 QuALITY 和 RACE, 以评估 CoLISA 模型的效果。其中, 主要实验是在 QuALITY 数据集上进行的³。同时, 本节也报告了在 RACE 数据集上的其他实验结果。

- QuALITY^[84]。这是一个多项选择阅读理解 (MC-RC) 数据集, 其每篇文章的平均长度约为 5000 个 token。该数据集的一个显著特征是一些干扰选项会对模型的认知能力造成负面影响。因此, 如果不依靠摘要或文本片段, 略读和简单

³基于 DPR 的检索器在 QuALITY 数据集中对于长文本输入的检索是可行的。此外, 最强干扰项仅在 QuALITY 数据集中有标注, 因此, 大部分实验均在该数据集上完成。

的搜索已经不足以让模型持续表现出优异的性能。该数据集的文本主要采集自科幻小说和杂志，并由标注人员进行阅读和评估。

- RACE^[15]。由于 QuALITY 只由 6,737 条源数据构成，难以全面评估实验结果，因此实验中还使用了另一个大规模 MC-RC 数据集 RACE 来验证模型的性能。RACE 从中国的中考和高考英语阅读理解试题中收集。大多数问题需要推理，并且文本涉及的领域是多样的，从新闻、故事到广告，这使得数据集更具挑战性。

本章主要研究针对 QuALITY^[84] 和 RACE^[15] 两大数据集的多项选择问答任务。为了评估模型的性能，本章采用准确率（Accuracy, ACC）作为主要的评估指标，用以衡量模型在回答问题时的正确率。由于两个数据集中存在不同难度的文本，因此本章的实验也分别考虑了 QuALITY 数据集中全部/困难子集的 acc 以及 RACE 数据集中中考/高考子集⁴的 ACC。对于基于 DPR 的检索器，本章采用了与 QuALITY 数据集中的基线模型相同的方法来减轻检索稀疏性的影响，以便进行公平比较。具体而言，本章使用了一个问题编码器对查询进行编码，以及一个上下文编码器对文章进行编码。

本节针对 CoLISA 的阅读器，采用了 DeBERTaV3-large 模型作为预训练语言模型。在 QuALITY 和 RACE 上的所有实验中，学习率均为 1e-5，热身率为 0.1。此外，Dropout 比例保持默认的 0.1，激活函数采用了 GeLU^[88]。所有实现均使用 16 的 batch 尺寸和 512 个 token 的最大长度。在 QuALITY 上进行了 20 轮的模型微调，在 RACE 上进行了 3 轮的微调。

为进行对比学习，由于实验开销的原因，本文在 DeBERTaV3-base 和 RoBERTa-base 模型上调整了温度系数 τ 。经过调参，基础模型上最优的 τ 值为 0.1，该值直接应用于大模型中。同时，交叉熵损失和对比损失分别分配一半到最终的综合损失中。此外，先前提到的干扰因子仅仅应用于 QuALITY 的实验中，因为 RACE 数据集没有对所有选项进行干扰程度标注。

所有实验都在一张 Tesla V100-32GB 的 GPU 上进行训练，Apex 中的 fp16 精度模式也用于加速训练过程。

虽然在 QuALITY 上有许多方法可以进行尝试，但本章仅选取了两个典型的基

⁴在 QuALITY 数据集中，源数据根据问题难度分为全部和困难子集，而在 RACE 数据集中，中考和高考子集代表了两个水平的入学考试试题。

线模型。

- Longformer^[66]。该预训练语言模型结合了滑动窗口局部注意力和全局注意力机制，来编码较长的文本序列。Longformer 支持最多达到 4,096 长度的输入序列。实验对比过程中采取了 Longformer 作为其中一个基线模型，因为它包含了需要回答 QuALITY 样本中大部分的文本。
- DPR & DeBERTaV3^[23,86]。该管道架构由一个检索器和一个阅读器组成。基于 DPR 的检索器用于在一篇文章中针对指定问题抽取最相关的上下文。被选取的上下文作为输入的一部分，将被喂给后续的模块。最后，一个用于 MC-RC 标准的 DeBERTa-V3 模型将会抉择出正确的答案选项。

5.3.2 实验结果和分析

本节对 CoLISA 模型与两个 QuALITY 上强力的基线模型进行比较，以及 RACE 上三个预训练语言模型进行比较进行分析。总体而言，表 5-2 所示的结果表明⁵，CoLISA 模型优于其他所有模型。其中，Longformer 模型的性能不佳，这说明 Longformer 模型在处理冗长的源文本时无法准确定位关键信息。这可能是由于长文本预训练语言模型需要更多的训练数据，而 QuALITY 提供的训练数据十分有限所致。此外，QuALITY 中文本的长度仍超过了 Longformer 模型的最大编码长度限制，这可能会导致关键信息的丢失。相比之下，DPR 和 DeBERTaV3 架构由于其自身的抽取策略，性能表现更为突出。与之前提到的两个基线模型相比，CoLISA 模型学习到了更有效的上下文表示方法，并通过引入的对比学习（CoL）策略，更加准确地识别多个备选答案之间的差异。同时，将 ISA 机制引入 DeBERTaV3 编码器之后，可以持续增强模型的性能表现。这种尝试意味着，用 ISA 风格的方式在多个选项之间进行内部交互，将进一步捕捉到所有备选答案之间的差异关系。

5.3.3 消融实验

对比学习。为了有效评估 CoL 模块对整个模型表现的影响，本节进行了一个消融实验，具体如表 5-3 中的第一列所示。首先，在抛弃 ISA 模块的情况下，可以评估

⁵本节在 QuALITY 数据集上进行了 DPR 和 DeBERTaV3 架构基线的重新实现（标有 *），表现明显优于 QuALITY 原文中的记录（53.6/47.4）。DeBERTaV3-large 的结果也是如此。而其他基线结果则来自相关研究或相关排行榜。在 RACE 数据集上的实验不需要基于 DPR 的检索器和干扰因素。在训练过程中，QuALITY 数据集中的模型是中间经过 RACE 数据集训练并在 QuALITY 上进行微调的。

表 5-2 QuALITY（全部/困难）和 RACE（中考/高考）数据集上，开发集和测试集的 ACC 实验结果

模型	QuALITY		RACE	
	开发集	测试集	开发集	测试集
Longformer-base ^[66]	38.1/32.8	39.5/35.3	-	-
DPR & DeBERTaV3-large ^[84]	56.7*/48.6*	55.4/46.1	-	-
DeBERTaV3-base ^[23]	-	-	81.1 (85.2/79.4)	79.7 (82.8/78.4)
XLNet-large ^[89]	-	-	80.1 (-/-)	81.8 (85.5/80.2)
RoBERTa-large ^[22]	-	-	- (-/-)	83.2 (86.5/81.8)
DeBERTaV3-large ^[23]	-	-	88.3* (91.4*/87.0*)	87.5* (90.5*/86.8*)
CoL (DeBERTaV3-base)	-/-	-/-	82.9 (87.3/81.0)	81.6 (85.3/80.1)
CoLISA (DeBERTaV3-base)	-/-	-/-	83.2 (86.4/81.9)	81.6 (84.6/80.4)
CoL (DeBERTaV3-large)	60.1/52.6	62.1/54.3	88.6 (91.6/87.3)	87.9 (90.8/86.9)
CoLISA (DeBERTaV3-large)	61.7/53.6	62.3/54.7	88.8 (91.1/87.8)	87.8 (90.0/87.0)

仅使用对比学习方法对模型整体性能的影响。实验结果表明，相比于基线模型（即表中前三列 DPR & DeBERTaV3-large 的架构），单独使用 CoL 组件的方法显著提升了模型性能。

表 5-3 QuALITY 数据集的开发集上，关于对比学习模块，干扰因子，样本内注意力机制的消融实验

模型	ACC	模型	ACC	模型	ACC	模型	ACC
基线	56.7	基线	56.7	基线	56.7	基线	39.6
CoL	60.1	CoL	60.1	CoL	60.1	CoL	40.8
用问题作为检索	58.9	应用干扰因子	60.9	应用自注意力	60.4	应用样本内负例	37.1
应用样本内负例	59.4	应用 KL 损失	57.0	上下文遮蔽	60.4		
				应用 transformer	61.0		
				应用两层 transformer	60.8		
				应用改动的 transformer	58.9		

对于基于 DPR 的检索器，CoL 组件将问题和多个备选答案都作为查询词，从参考文章中抽取上下文。相比之下，仅使用问题作为查询词会导致性能大幅下降。这一结果的解释是，CoL 方法主要针对于将正确答案和干扰选项进行区分，从而更好地抽取与问题/备选答案相关的证据文本。

通过使用标准的 dropout 两次，可以得到两个不同的语义文本的表示。在对比学习的过程中，每一个表示分别包含了一个正例。在收集负例的时候，先前的工作会将同一个 batch 中剩余的样本视为负例，通常称之为 batch 内的负例^[64]。本节也测试了一下 batch 内负例的方法。遵循这种方法，实验结果揭示，将构建负例的方法从样本内部改为 batch 内部，会导致性能下跌⁶。在同一个 batch 内，多个样本之间没有

⁶事实上，基础模型上的性能远远低于表格中列出的，当把相同的实验从大模型迁移到基础模型时，会表现出更

必然的联系，这就违背了将正确答案按和干扰选项推远的目标。

干扰因子。正如之前所提到的，CoLISA 中设计了一个专门针对 QuALITY 的干扰因子（Distractive Factor, DiF），以强调困惑选项的作用。表 5-3 中的第二列展示了 DiF 的实现结果。本节中将 DiF Θ 与相应选项的相似度进行相乘操作，以体现对干扰项的权重。实验结果表明，DiF 模块显著提高了 CoL 模块的性能。一个直观的解释是，DiF 迫使干扰选项对比损失贡献更多的数值，从而使模型更倾向于学习如何识别干扰选项。

基于一个假设，每个备选答案成为最终答案的概率分布遵循特定的概率分布。本节在实验中使用 KL 散度损失^[90] 替换原来的交叉熵损失，这是一种直观的做法。实验结果表明，KL 散度损失的效果与 DiF 相似。此外，交叉熵损失对于正确区分每个三元组（即上下文、问题和备选答案）的正确性至关重要。

样本内注意力。如表 5-3 中的第三列所示，本节中还利用了 ISA 机制的变体来验证它们的有效性。可以如预期中观察到，当 ISA 机制是自注意力机制时，实验性能会由于多个三元组之间存在交互而提升。然后，实验进一步尝试去遮蔽上下文 token，这就意味着只有问题与选项彼此之间可以进行注意力交互，并没有上下文信息的参与。结果性能是维持不变，这表明问题和备选答案之间的注意力交互是关键的步骤。

消融实验也进一步展示了 transformer 架构优于单独的一层自注意力层。这是因为，transformer 内部的前馈网络保留了大量的参数，来确保注意力输出的传播。另有一个有趣的现象，如果再配上额外的一层 transformer 层，会导致模型性能轻微的下降。本节认为，从 RACE 和 QuALITY 上先后微调的 checkpoint 被继续沿用，扩充了随机初始化的参数，这会加重训练负担。

此外，本节还通过移植编码器内部的注意力机制，修改了基础编码器的内部结构。整个预训练编码器由 n 层⁷组合而成，其中每一层共享完全相同的结构：一个多头自注意力子层和一个前馈神经网络子层。接下来，本节也针对多项选择的交互，尝试在两个子层之间增加了一个额外的注意力层。注意到，更低的层主要表达浅层的语义，而更高的层主要针对深层语义，实际操作中只在在顶端的 4 层补充了注意力机制。这种修改过后的 transformer 架构直觉上建模了低层次中的序列内交互，以及高层次中的多序列交互。实验结果表明，修改编码器架构并不是最好的方法，因为这么

差的性能。结果列在了表 5-3 的最后一列中。性能从 40.8 疯狂跌到 37.1。这里的基线模型是 RoBERTa-base。对于大模型，由于机器设备的限制，实验中不得不使用一个较小的 batch 尺寸。因此，在大模型上无论是 batch 内的方法还是样本内的方法，都不能展示出巨大的差异。

⁷对于基础模型， n 是 12；对于大模型， n 是 24。

做并没有完全利用预训练的成果。

5.4 本章小结

本章主要研究多项选择长文本阅读理解的学习方法和注意力机制，旨在解决备选答案中的干扰选项问题。本章提出的方法 CoLISA 通过实现多个选项之间的交互，有效地解决了正确答案和干扰选项之间对比的难题。CoLISA 在两个基准数据集上的实验表明，其性能得到了持续的提升。未来的研究方向可以探索更多与对比学习方法相关的变体。

第六章 总结与展望

6.1 工作总结

长文本机器阅读理解是指计算机能够理解长篇文本内容并回答相关问题的能力。利用机器阅读理解技术，搜索引擎能够直接返回用户提出的问题的正确答案，而不再受限于从检索召回的文档中推理出最终答案。这种技术大大提高了信息检索的有效性和用户体验。然而，传统的基于预训练语言模型的方法面临着难以处理超过一定长度文本的挑战。这会导致信息丢失和推理困难，从而引发一系列问题，例如如何将备选答案用于检索长文本，以及如何处理多跳问题等。为了解决这些问题，本文从三个角度提出了相应的解决方法：一是基于检索器-阅读器的二阶段架构，二是基于对比学习的选项交互，三是基于多文档问答中问题分解的研究方法。这些方法具有针对性，能够有效地提高长文本机器阅读理解的能力。

(1) 基于检索器-阅读器二阶段架构的长文本阅读理解研究

针对滑动窗口机制在长文本阅读理解任务中存在的局限性，即由于固定长度限制而导致的长距离依赖缺失和信息丢失问题，本文提出了一种基于二阶段架构的方法。该方法由检索器和阅读器两个模块组成，其中检索器负责对文本片段进行可回答性评分和排序，从而筛选出高置信度的证据片段，而阅读器则负责从证据片段堆中抽取答案片段。通过这种方法，不仅可以保留关键信息，而且可以缩小答案搜索空间，提高答案抽取的效率和准确性。实验结果表明，该二阶段方法有效地提升了抽取式长文本阅读理解模型的能力。与基线模型相比，该方法提升了 0.6% 的 F1 值和 1.2% 的 EM 值。

(2) 基于问题分解的多跳长文本阅读理解研究

本文针对多文档阅读理解任务中问题复杂性高以及传统机器阅读理解模型缺乏多跳推理能力的挑战，提出了两种基于问题分解的方法，将多跳问题转化为单跳问题。第一种方法采用序列到序列的生成式模型，生成一系列单跳问题，并结合检索模型和阅读理解模型从多个文档中检索和抽取答案。第二种方法则在生成当前单跳问题时，利用前驱问题和前驱答案作为额外输入来引导生成过程，并通过检索模型和阅读理解模型获取单跳答案，直到生成带有结束标志的单跳问题。在 MuSiQue 数据集上，这两种方法相较于基线模型，在证据 F1 指标上分别提升了 2.4% 和 2.3%，在答

案 F1 指标上分别提升了 1.2% 和 0.8%。

(3) 基于对比学习的多项选择长文本阅读理解研究

本文针对多项选择阅读理解任务中的干扰选项与正确选项字面相似度高以及选项编码缺乏交互的问题，提出了一种基于对比学习和自注意力交互的方法来增强选项编码表示和区分能力。该方法首先采用对比学习方法来训练一个选项编码器，使得不同选项之间的语义差异能够在编码空间中得到体现。其次，利用样本内自注意力交互机制来建立各选项之间的交互关系，从而增强选项编码的区分能力。经过实验验证，本文提出的方法相比于基线模型，在 QuALITY 数据集上全部数据集上提升了 6.9% 的 ACC 值，在困难数据集上提升了 8.6% 的 ACC 值，具有较好的效果。

6.2 工作展望

本文针对面向长文本的机器阅读理解进行研究，并有效提高了 NewsQA, MuSiQue 和 QuALITY 等数据集相应的实验性能。同时，在实验过程中，本文总结出以下几个亟待改进的地方：

(1) 长文本阅读理解中的指代消解问题

在长文本机器阅读理解任务中，由于文本长度巨大，存在大量指代消解问题。本文第三章提出了一种基于检索器-阅读器二阶段架构的方法，该方法利用了预训练语言模型的能力，建立了前后文中的共指关系。然而，对于一些难以挖掘的指代关系，本文未能深入探讨。因此，在未来的研究中，可以继续研究指代消解的方法，以进一步提高本文提出的模型的性能。

(2) 基于大模型和提示的多跳阅读理解

针对多跳阅读理解问题，本文提出了两种基于问题分解的方法，以提高模型搜索证据的能力。尽管实验过程中已经尝试了一些生成式模型，但并未对所有模型进行全面的探索。考虑到目前 ChatGPT 等大型模型的发展，这些模型或许更适合于解决问题分解任务。在未来的研究中，可以结合递归提示等提示技术，将复杂问题分解为多个简单子问题，并逐步回答，以进一步提高模型的准确性和效率。

(3) 多项选择阅读理解中的多种对比学习方法

本文的第五章节采用了对比学习方法，以建立多项选择长文本阅读理解中不同选项之间的联系。同时，本文还提出了一些衍生实验，例如构建样本内负例和增加干

扰因子等。然而，这些实验并没有涉及多种对比学习方法。因此，未来的工作可以探索将其他技术如知识图谱等加入到对比学习方法中，以及从对比损失权重以及温度系数的设置等角度，进行更全面的实验。

参 考 文 献

- [1] Liu S, Zhang X, Zhang S, Wang H, Zhang W. Neural machine reading comprehension: Methods and trends [J]. ArXiv, 2019, abs/1907.01118.
- [2] Zhang Z, Zhao H, Wang R. Machine reading comprehension: The role of contextualized language models and beyond [J]. ArXiv, 2020, abs/2005.06249.
- [3] Cui Y, Liu T, Xiao L, Chen Z, Ma W, Che W, Wang S, Hu G. A span-extraction dataset for chinese machine reading comprehension [C]//EMNLP-IJCNLP. 2019.
- [4] Li R, Zhang X, Li C, Zheng Z, Zhou Z, Geng Y. Keyword extraction method for machine reading comprehension based on natural language processing [J]. Journal of Physics: Conference Series, 2021, 1955.
- [5] Yu C, Li X. Ssag-net: Syntactic and semantic attention-guided machine reading comprehension [J]. Intelligent Automation & Soft Computing, 2022.
- [6] Hermann K M, Kočiský T, Grefenstette E, Espeholt L, Kay W, Suleyman M, Blunsom P. Teaching machines to read and comprehend [J]. arXiv preprint arXiv:1506.03340, 2015.
- [7] Dong Y. A survey on neural network-based summarization methods [J]. ArXiv, 2018, abs/1804.04589.
- [8] Liu P, Qiu X, Huang X. Recurrent neural network for text classification with multi-task learning [J]. ArXiv, 2016, abs/1605.05101.
- [9] Trabelsi M A, Chen Z, Davison B D, Heflin J. Neural ranking models for document retrieval [J]. Information Retrieval Journal, 2021, 24: 400 - 444.
- [10] Lan G, Li Y, Hu M, Sun Y, Zhang Y. Knowledge graph integrated graph neural networks for chinese medical text classification [J]. 2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 2021: 682-687.
- [11] Ebersbach S. Artificial neural networks in real life applications [C]//2016.
- [12] Rajpurkar P, Zhang J, Lopyrev K, Liang P. Squad: 100,000+ questions for machine comprehension of text [J]. arXiv preprint arXiv:1606.05250, 2016.
- [13] Rajpurkar P, Jia R, Liang P. Know what you don't know: Unanswerable questions for squad [C]//Annual Meeting of the Association for Computational Linguistics.

- 2018.
- [14] Trischler A, Wang T, Yuan X, Harris J, Sordoni A, Bachman P, Suleiman K. Newsqa: A machine comprehension dataset [J]. arXiv preprint arXiv:1611.09830, 2016.
 - [15] Lai G, Xie Q, Liu H, Yang Y, Hovy E. Race: Large-scale reading comprehension dataset from examinations [J]. arXiv preprint arXiv:1704.04683, 2017.
 - [16] Yang Z, Qi P, Zhang S, Bengio Y, Cohen W W, Salakhutdinov R, Manning C D. Hotpotqa: A dataset for diverse, explainable multi-hop question answering [C]// Conference on Empirical Methods in Natural Language Processing. 2018.
 - [17] Seo M, Kembhavi A, Farhadi A, Hajishirzi H. Bidirectional attention flow for machine comprehension [J]. ArXiv, 2016, abs/1611.01603.
 - [18] Shen Y, Huang P S, Gao J, Chen W. Reasonet: Learning to stop reading in machine comprehension [J]. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.
 - [19] Yu A W, Dohan D, Luong M T, Zhao R, Chen K, Norouzi M, Le Q V. Qanet: Combining local convolution with global self-attention for reading comprehension [J]. ArXiv, 2018, abs/1804.09541.
 - [20] Wang S, Jiang J. Machine comprehension using match-lstm and answer pointer [J]. ArXiv, 2016, abs/1608.07905.
 - [21] Devlin J, Chang M W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding [J]. arXiv preprint arXiv:1810.04805, 2018.
 - [22] Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V. Roberta: A robustly optimized bert pretraining approach [J]. arXiv preprint arXiv:1907.11692, 2019.
 - [23] He P, Gao J, Chen W. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing [J]. arXiv preprint arXiv:2111.09543, 2021.
 - [24] He W, Liu K, Liu J, Lyu Y, Zhao S, Xiao X, Liu Y, Wang Y, Wu H, She Q, Liu X, Wu T, Wang H. Dureader: a chinese machine reading comprehension dataset from real-world applications [C]//QA@ACL. 2017.

- [25] Xu L, Hu H, Zhang X, Li L, Cao C, Li Y, Xu Y, Sun K, Yu D, Yu C, et al. Clue: A chinese language understanding evaluation benchmark [J]. arXiv preprint arXiv:2004.05986, 2020.
- [26] Chen W, Fan C, Wu Y, Wang Y. Chinese machine reading comprehension based on language model containing knowledge [J]. Proceedings of the 6th International Conference on Computer Science and Application Engineering, 2022.
- [27] Xue Y. Machine reading comprehension model based on multi-head attention mechanism [C]//Advanced Intelligent Technologies for Industry: Proceedings of 2nd International Conference on Advanced Intelligent Technologies (ICAIT 2021). Springer, 2022: 45-58.
- [28] Joshi M, Choi E, Weld D S, Zettlemoyer L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension [C]//Annual Meeting of the Association for Computational Linguistics. 2017.
- [29] Kociský T, Schwarz J, Blunsom P, Dyer C, Hermann K M, Melis G, Grefenstette E. The narrativeqa reading comprehension challenge [J]. Transactions of the Association for Computational Linguistics, 2017, 6: 317-328.
- [30] Joshi M, Levy O, Weld D S, Zettlemoyer L. Bert for coreference resolution: Baselines and analysis [J]. arXiv preprint arXiv:1908.09091, 2019.
- [31] Chen D, Bolton J, Manning C D. A thorough examination of the cnn/daily mail reading comprehension task [J]. ArXiv, 2016, abs/1606.02858.
- [32] Wang W, Yang N, Wei F, Chang B, Zhou M. R-net: Machine reading comprehension with self-matching networks [C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 189-198.
- [33] Wang W, Yang N, Wei F, Chang B, Zhou M. Gated self-matching networks for reading comprehension and question answering [C]//Annual Meeting of the Association for Computational Linguistics. 2017.
- [34] Karpukhin V, Oğuz B, Min S, Lewis P, Wu L Y, Edunov S, Chen D, tau Yih W. Dense passage retrieval for open-domain question answering [J]. ArXiv, 2020, abs/2004.04906.
- [35] Dai Z, Yang Z, Yang Y, Carbonell J G, Le Q V, Salakhutdinov R. Transformer-

- xl: Attentive language models beyond a fixed-length context [J]. ArXiv, 2019, abs/1901.02860.
- [36] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L, Polosukhin I. Attention is all you need [J]. arXiv preprint arXiv:1706.03762, 2017.
- [37] Ding M, Zhou C, Yang H, Tang J. Cogltx: Applying bert to long texts [C]// Neural Information Processing Systems. 2020.
- [38] Allam A M N, Haggag M H. The question answering systems: A survey [J]. 2012.
- [39] Kwiatkowski T, Palomaki J, Redfield O, Collins M, Parikh A P, Alberti C, Epstein D, Polosukhin I, Devlin J, Lee K, Toutanova K, Jones L, Kelcey M, Chang M W, Dai A M, Uszkoreit J, Le Q V, Petrov S. Natural questions: A benchmark for question answering research [J]. Transactions of the Association for Computational Linguistics, 2019, 7: 453-466.
- [40] Dunn M, Sagun L, Higgins M, Güney V U, Cirik V, Cho K. Searchqa: A new q&a dataset augmented with context from a search engine [J]. ArXiv, 2017, abs/1704.05179.
- [41] Sun H, Dhingra B, Zaheer M, Mazaitis K, Salakhutdinov R, Cohen W W. Open domain question answering using early fusion of knowledge bases and text [C]// Conference on Empirical Methods in Natural Language Processing. 2018.
- [42] Jones K S. A statistical interpretation of term specificity and its application in retrieval [J]. J. Documentation, 2021, 60: 493-502.
- [43] Chen D, Fisch A, Weston J, Bordes A. Reading wikipedia to answer open-domain questions [C]//Annual Meeting of the Association for Computational Linguistics. 2017.
- [44] Lee K, Chang M W, Toutanova K. Orqa: Open retrieval question answering [J]. arXiv preprint arXiv:1906.00300, 2019.
- [45] Roberts A, Raffel C, Shazeer N M. How much knowledge can you pack into the parameters of a language model? [J]. ArXiv, 2020, abs/2002.08910.
- [46] Song L, Wang Z, Hamza W, Gildea D. Multi-passage machine reading comprehension with cross-passage answer verification [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 33. 2019: 6079-6086.
- [47] Ding M, Zhou C, Chen Q, Yang H, Tang J. Cogqa: A question answering system

- over comprehensive knowledge bases [J]. arXiv preprint arXiv:1905.07129, 2019.
- [48] Luo K, Lin F, Luo X, Zhu K Q. Knowledge base question answering via encoding of complex query graphs [C]//Conference on Empirical Methods in Natural Language Processing. 2018.
- [49] Zhang Z, Zhang Y, Liu K, Sun H, Han X, Sun L. Semantic parsing for complex question answering over knowledge bases [J]. Information Processing & Management, 2021, 58(5): 102653.
- [50] Tu M, Wang G, Huang J, Tang Y, He X, Zhou B. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs [C]// Annual Meeting of the Association for Computational Linguistics. 2019.
- [51] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks [J]. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015.
- [52] Wu Y, Wu W, Yang D, Xu C, Li Z. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots [C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 496-505.
- [53] Gan Z, Cheng Y, Kholy A E, Li L, Liu J, Gao J. Multi-step reasoning via recurrent dual attention for visual dialog [J]. ArXiv, 2019, abs/1902.00579.
- [54] Zhou T, Gong P. An object detection framework for span extraction in question answering [J]. 2021 7th IEEE International Conference on Network Intelligence and Digital Content (IC-NIDC), 2021: 36-40.
- [55] Sordoni A, Galley M, Auli M, Brockett C, Ji Y, Mitchell M, Nie J Y, Gao J, Dolan W B. A neural network approach to context-sensitive generation of conversational responses [C]//North American Chapter of the Association for Computational Linguistics. 2015.
- [56] Moschitti A, Quarteroni S. Kernels on linguistic structures for answer extraction [C]//Annual Meeting of the Association for Computational Linguistics. 2008.
- [57] Thomas A, Sivanesan S. An adaptable, high-performance relation extraction system for complex sentences [J]. Knowl. Based Syst., 2022, 251: 108956.
- [58] Shi P, Lin J J. Simple bert models for relation extraction and semantic role

- labeling [J]. ArXiv, 2019, abs/1904.05255.
- [59] Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C. Neural architectures for named entity recognition [C]//North American Chapter of the Association for Computational Linguistics. 2016.
- [60] Cui Y, Liu T, Che W, Wang L, Hu G, Wei F. Multi-span extraction for multi-hop reading comprehension: A sequential approach [C/OL]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online: Association for Computational Linguistics, 2020: 7319-7331. <https://www.aclweb.org/anthology/2020.acl-main.662>.
- [61] Zhang Y, Liu K, He S, Ji G, Liu Z, Wu H, Zhao J. Question answering over knowledge base with neural attention combining global knowledge information [J]. ArXiv, 2016, abs/1606.00979.
- [62] Lewis M, Liu Y, Goyal N, Ghazvininejad M, Mohamed A, Levy O, Stoyanov V, Zettlemoyer L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension [C]//Annual Meeting of the Association for Computational Linguistics. 2019.
- [63] Ni J, 'Abrego G H, Constant N, Ma J, Hall K B, Cer D M, Yang Y. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models [C]//Findings. 2021.
- [64] Gao T, Yao X, Chen D. Simcse: Simple contrastive learning of sentence embeddings [J]. arXiv preprint arXiv:2104.08821, 2021.
- [65] Xie Q, Dai Z, Hovy E, Luong M T, Le Q V. Unsupervised data augmentation for consistency training [J]. arXiv preprint arXiv:1904.12848, 2019.
- [66] Beltagy I, Peters M E, Cohan A. Longformer: The long-document transformer [J]. arXiv preprint arXiv:2004.05150, 2020.
- [67] Zaheer M, Guruganesh G, Dubey A, Ainslie J, Alberti C, Ontanon S, Pham P, Ravula A, Wang Q, Yang L, et al. Big bird: Transformers for longer sequences [J]. arXiv preprint arXiv:2007.14062, 2020.
- [68] Ding S, Shang J, Wang S, Sun Y, Tian H, Wu H, Wang H. Ernie-doc: The retrospective long-document modeling transformer [J]. arXiv preprint arXiv:2012.15688, 2020.

- [69] Atkinson R C, Shiffrin R M. Human memory: A proposed system and its control processes [M]//Psychology of learning and motivation: volume 2. Elsevier, 1968: 89-195.
- [70] Joshi M, Chen D, Liu Y, Weld D S, Zettlemoyer L, Levy O. Spanbert: Improving pre-training by representing and predicting spans [J]. Transactions of the Association for Computational Linguistics, 2020, 8: 64-77.
- [71] Tay Y, Tuan L A, Hui S C, Su J. Densely connected attention propagation for reading comprehension [J]. arXiv preprint arXiv:1811.04210, 2018.
- [72] Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations [J]. arXiv preprint arXiv:1909.11942, 2019.
- [73] Zhang Z, Yang J, Zhao H. Retrospective reader for machine reading comprehension [J]. arXiv preprint arXiv:2001.09694, 2020.
- [74] Wang S, Jiang J. Learning natural language inference with lstm [J]. arXiv preprint arXiv:1512.08849, 2015.
- [75] Seo M, Kembhavi A, Farhadi A, Hajishirzi H. Bidirectional attention flow for machine comprehension [J]. arXiv preprint arXiv:1611.01603, 2016.
- [76] Kundu S, Ng H T. A question-focused multi-factor attention network for question answering [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 32. 2018.
- [77] Mikolov T, Karafiat M, Burget L, Černocky J, Khudanpur S. Recurrent neural network based language model [C]//Eleventh annual conference of the international speech communication association. 2010.
- [78] Ding M, Zhou C, Yang H, Tang J. Cogltx: Applying bert to long texts [J]. Advances in Neural Information Processing Systems, 2020, 33.
- [79] Wu Y, Schuster M, Chen Z, Le Q V, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, et al. Google's neural machine translation system: Bridging the gap between human and machine translation [J]. arXiv preprint arXiv:1609.08144, 2016.
- [80] Fang Y, Sun S, Gan Z, Pillai R R, Wang S, Liu J. Hierarchical graph network for multi-hop question answering [C]//Conference on Empirical Methods in Natural

- Language Processing. 2019.
- [81] Tu M, Huang K, Wang G, Huang J, He X, Zhou B. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents [J]. ArXiv, 2019, abs/1911.00484.
- [82] Ran Q, Li P, Hu W, Zhou J. Option comparison network for multiple-choice reading comprehension [J]. arXiv preprint arXiv:1903.03033, 2019.
- [83] Zhang S, Zhao H, Wu Y, Zhang Z, Zhou X, Zhou X. Dcmn+: Dual co-matching network for multi-choice reading comprehension [C]//Proceedings of the AAAI Conference on Artificial Intelligence: volume 34. 2020: 9563-9570.
- [84] Pang R Y, Parrish A, Joshi N, Nangia N, Phang J, Chen A, Padmakumar V, Ma J, Thompson J, He H, et al. Quality: Question answering with long input texts, yes! [J]. arXiv preprint arXiv:2112.08608, 2021.
- [85] Daniel K. Thinking, fast and slow [Z]. 2017.
- [86] Karpukhin V, Oğuz B, Min S, Lewis P, Wu L, Edunov S, Chen D, Yih W t. Dense passage retrieval for open-domain question answering [J]. arXiv preprint arXiv:2004.04906, 2020.
- [87] Sukhbaatar S, Grave E, Lample G, Jegou H, Joulin A. Augmenting self-attention with persistent memory [J]. arXiv preprint arXiv:1907.01470, 2019.
- [88] Hendrycks D, Gimpel K. Gaussian error linear units (gelus) [J]. arXiv preprint arXiv:1606.08415, 2016.
- [89] Yang Z, Dai Z, Yang Y, Carbonell J, Salakhutdinov R R, Le Q V. Xlnet: Generalized autoregressive pretraining for language understanding [J]. Advances in neural information processing systems, 2019, 32.
- [90] Hershey J R, Olsen P A. Approximating the kullback leibler divergence between gaussian mixture models [C]//2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07: volume 4. IEEE, 2007: IV-317.

攻读学位期间的成果

• 论文

- (1) Mengxing Dong, Bowei Zou, Jin Qian, Rongtao Huang, Yu Hong. 2021. ThinkTwice: A Two-Stage Method for Long-Text Machine Reading Comprehension. In Proceedings of NLPCC, pages 427-438, Qingdao, China. (CCF-C 类会议)
- (2) Mengxing Dong, Bowei Zou, Yanling Li, Yu Hong. 2023. CoLISA: Inner Interaction via Contrastive Learning for Multi-Choice Reading Comprehension. In Proceedings of ECIR, pages 264-278, Dublin, Ireland. (CCF-C 类会议)
- (3) Jin Qian, Bowei Zou, Mengxing Dong, Xiao Li, Ai Ti Aw, Yu Hong. 2022. Capturing Conversational Interaction for Question Answering via Global History Reasoning. In Proceedings of NACCL, Online. (CCF-C 类会议)
- (4) Xibo Li, Bowei Zou, Mengxing Dong, Jianmin Yao, Yu Hong. 2022. Concession-First Learning and Coarse-to-Fine Retrieval for Open-Domain Conversational Question Answering. In Proceedings of ICTAI, Online. (CCF-C 类会议)

• 实习

- (1) 2021/8--2022/1. 北京-百度-搜索策略部.

致谢

在完成本篇毕业论文之际，我想向那些在我求学过程中给予我帮助和支持的人们表示最诚挚的感谢。

首先，我要感谢我的指导老师洪宇老师和邹博伟老师，是您们在我的研究过程中给予我耐心的指导和鼓励，让我逐步深入了解自然语言处理相关的知识，提高了自己的专业水平。同时，我还要感谢周国栋、朱巧明、陈文亮、李培峰、李寿山、孔芳、姚建民、钱龙华、王红玲、段湘煜、李军辉、贡正仙、李正华、朱晓旭、周夏冰、王中卿等苏州大学自然语言处理实验的所有老师，感谢他们为苏州大学自然语言处理实验室的同学们提供的优良实验氛围与计算资源。

其次，我要感谢我的家人和朋友，是他们一直以来的支持和鼓励，让我有勇气和信心坚持学习和研究。在我的学习和研究过程中，他们无时无刻不在我身边，给予我关爱和帮助，让我感到无比幸福和温暖。

此外，我还要感谢实验室的所有同学们。感谢黄荣涛、唐竑轩、尉桢楷、李志峰、钱锦、李烨秋、朱鸿雨、王捷、潘雨晨和徐旻涵师兄以及孙雨、朱朦朦、武凯莉、陈佳丽、李晓、苏玉兰、徐庆婷、范怡帆师姐为我的科研提供的帮助和指导；感谢同届的李志峰、金志凌、窦祖俊、刘皓、刘东、李中秋同学，在三年共处时光里给予我的帮助和关心；感谢李希博、邢小林、陈家祥、彭睿、何仕铭、陆煜翔、徐浩宇、刘超群、王军杰、杨帅师弟和李妍灵、丁楚瑶师妹，感谢大家一起营造了实验室良好的氛围。

感谢我在百度实习期间给与我帮助的石磊、文武、余沾、王振师兄，以及李志峰、周涛同学。

最后，我还要感谢各位评审老师，感谢各位老师们在百忙之中抽取时间对本文进行评审，并提出宝贵的修改意见。

学位论文答辩委员会决议

- 包括：1、对论文的评价，包括选题的理论价值和实践意义，论文理论、方法上的开拓与创新，论据的可靠充分与结论的正确性；论文所反映的作者学术视野（对本学科及相关领域研究动态的把握）、基础理论、专业知识、写作能力等；
2、对答辩的评价；
3、是否同意通过论文答辩，是否建议授予学位或是否建议在规定时间内修改论文后重新答辩一次的结论。

论文在依存句法分析和成分句法分析这两种句法分析任务上，探讨了基于树形条件随机场的高阶方法对句法分析器性能和效率的影响。选题具有很好的理论价值和实践意义，以及一定的创新性。目前主流的句法分析方法大多基于神经网络方法，并采用了一个简化的学习目标，相对应地，传统方法中大多采用了结构化学习以及高阶建模。论文针对这一对比，提出了将当前的句法分析器与传统方法做一个联结。论文提出在神经网络模型中采用树形条件随机场来最大化树概率，并进一步提出采用高阶建模。为了改善带来的效率问题，论文分别尝试了批次化计算和变分推断近似方法来加速。结果表明结构化建模和高阶方法对于目前的句法分析器仍然是有益的。

作者比较全面地论述了相关研究领域国内外研究情况，所采用的研究方法和技术手段体现了作者良好的学术研究基础和能力。论文成果在研究方法等方面有所创新，其中的新方法、新思路具备了很好的应用价值。论文写作层次清晰，逻辑结构合理，文字流畅，符合学术规范。

论文质量优秀。答辩过程中陈述清楚，回答问题准确。

答辩委员会经讨论，认为该论文已达到硕士学位论文水平，一致同意其通过论文答辩，建议授予硕士学位。

答辩委员会主席：_____ 孔芳 _____ 秘书：_____ 张雅静 _____

委员：_____ 陈文亮 _____、_____ 李培峰 _____、_____ 钱龙华 _____、_____ 朱晓旭 _____

_____、_____、_____、_____、_____

2021 年 5 月 22 日

注：本表内容（包括答辩名单）可手签或打印