神经机器阅读理解研究综述

摘　要: 机器阅读理解的目的是使得机器能够理解自然语言文本，它是自然语言处理领域十分重要的研究方向。随着深度学习技术的进步以及大规模数据集的发布，在机器阅读理解方向上的研究已经取得了很大的突破。近年来随着自然语言处理领域预训练模型的出现，再一次推动了机器阅读理解领域的发展。本文主要从四个方面对自 2015 年以来机器阅读理解领域的发展做综述：介绍机器阅读理解的任务定义以及相关数据集；分析机器阅读理解领域经典的基于注意力机制或推理结构的模型以及目前流行的预训练模型；探讨更加复杂的机器阅读理解任务；总结机器阅读理解领域目前存在的问题并且对未来的研究趋势做展望。
关键词: 机器阅读理解；自然语言处理；预训练模型；注意力机制；推理结构
文献标识码： A　　**中图分类号：** TP391

Overview of Studies on Neural Machine Reading Comprehension

Abstract: Machine Reading Comprehension aims to make machines comprehend the natural language documents, which is an important research direction in the field of natural language processing. With the development of deep learning technology and release of large scale datasets, the research on the field of machine reading comprehension has made great breakthroughs. With the emergence of pre-trained model in natural language processing in recently years, it promotes the development of machine reading comprehension once again. This paper mainly make a survey from four aspects over the development of machine reading comprehension in recently years: to introduce the definition of machine reading comprehension tasks and its corresponding datasets; to analyze classical model in the field of machine reading comprehension which based on attention mechanism or reasoning structure as well as the currently popular pre-trained model; to discuss more complicated machine reading comprehension tasks; to summarize the existing problem and look into the future research trend about machine reading comprehension.
Key words: machine reading comprehension; natural language processing; pre-trained model; attention mechanism; reasoning structure

1　引言

随着机器学习、深度学习等人工智能技术的蓬勃发展，人类在图像识别、语音识别、围棋等领域已经接近人类甚至超越人类水平。自然语言处理是实现智能、人机交互的重要基石，然而由于人类语言的抽象和概括特征，目前自然语言处理领域仍然没有一个能够与人类正常沟通的问答系统。

机器阅读理解是认知智能领域一个具有挑战性的任务，早在 20 世纪 70 年代，学者们就已经意识到机器阅读技术是测试计算机理解人类语言的关键方法，1999 年出现首个自动阅读理解测试系统 Deep Read，该系统以故事为基础衡量阅读理解任务，利用词袋模型和人工编写规则进行模式匹，准确率可以达到 40% 左右，但是模型鲁棒性很差，MCTest 数据集于 2013 年发布，规模很小。总之由于没有合适的文本表示方法，数

据集的规模又比较小以及传统机器学习模型的有限拟合能力，机器阅读理解发展缓慢，早期的 MRC 系统难以获得期望的性能也不能实际的应用。随着深度学习的兴起以及 NLP 领域预训练词向量如 Word2Vec[33]、GloVe[37] 的发展以及注意力（Attention）机制在 NLP 领域的应用[2] 等。2015 年，DeepMind 研究员 Hermann 等人[15] 提出使用神经网络模型解决 MRC 任务。他们提出一种新颖且代价小的解决方案用来构建 MRC 领域大规模的监督训练数据，基于此方案他们构建了规模比以往数据集都要大的阅读理解数据集 CNN&Daily Mail，并且提出两个基于神经网络和注意力机制构建的模型（Attentive Reader，Impatient Reader），模型在 CNN&Daily 数据集上效果远超过传统方法的结果。这项工作可以视为机器阅读理解领域的奠基性工作。此后越来越多的学者在这两个模型的基础上构建效果更好的神经机器阅读理解模型，如 Chen 等人[4] 使用双线性函数简化 Attentive Reader 的加法形式，Kadlec 等人[25] 基于指针网络 [52] 的思想改进输出层的设计提升模型的预测准确率。

然而 CNN&Daily Mail 数据集的构造方式使得数据集有一定的噪声，此外数据集的问题是自动生成而且答案仅仅是原文中的某个实体名词，对模型的理解能力要求不高。为了解决这些限制问题，Rajpurkar 等人 [41] 发布了大规模数据集 SQuAD。SQuAD 包含有 536 篇维基百科的文章，问题由众包工人基于文章人工生成并且问题的答案来源于原文中某一片段而不像 CNN& Daily Mail 数据集仅仅是某个实体单词。由于数据集规模大、质量高，易于评估等特点获得了广泛的关注度，极大地推动了 MRC 领域的发展，很多经典的神经机器阅读理解模型（如 Match-LSTM[53]，DCN[60]，BiDAF[46]，R-Net[55]）都是在 SQuAD 数据集上构建出来的。

尽管如此，SQuAD 数据集仍然存在几个问题。现实生活中人们往往是先提出问题然后阅读文章查找答案，因此 SQuAD 数据集这种由文章生成问题的构造方式并不符合现实场景下的阅读理解任务，此外限制问题的答案是原文中某一连续的文本不足以回答复杂的问题，对模型的理解能力要求不高。在 SQuAD 发布不久后微软研究院发布了来源于真实场景下的数据集 MS MARCO[34]，数据集的问题来源于必应搜索日志上用户搜索的问题，从必应搜索的返回结果中选取 10 篇最相关的段落作为问题的答案依据，答案是人工生成的而不限于原文中某段文本，因此数据集难度更大。类似的数据集还有 TriviaQA[24]，DuReader[14] 等。之前的模型如 R-Net 在这种多段落自由答案形式的数据集上效果并不好，Tan 等人[49] 提出 S-Net 模型，在 R-Net 的基础上添加段落排名算法，然后利用生成模块生成答案，效果优于 R-Net 模型。

然而以上大部分的数据集，与问题相关的答案通常集中在单个句子的局部上下文这类数据集对模型的推理能力要求不高。因此研究者们尝试难度更大，更加接近于人类阅读理解形式的任务。如多步推理阅读理解任务，要求模型从多个段落中逐步检索推理才能找到答案，相关的数据集如 HotpotQA[61]，WIKIHOP[58]。

每一个新的数据集都会解决原有数据集存在的一些问题，从而不得不设计更加优秀的模型处理这些新的任务，MRC 领域也因此快速发展。自 2018 年随着 GPT[39]、BERT[10] 等预训练模型的出现，再一次提升了机器阅读理解模型的性能，甚至在某些数据集上模型的表现超过人类水平。

本文主要从 MRC 的具体任务概述出发，

2 机器阅读理解任务概述

机器阅读理解（MRC）任务是为了使得计算机具有对自然语言文本理解的能力，像人类一样阅读并且理解一篇文章。MRC 可以用一个三元组 $< D, Q, A >$ 来描述，其中 D 代表文章（Document）[1]，Q 表示问题（Question），A 表示答案（Answer），即给定一篇文章 D 和一些与文章 D 相关的问题 Q，要求模型通过阅读 D 之后给出 Q 的正确答案 A，即建模给定 D 和 Q 的条件下预测 A 的概率：$P(A|D,Q)$ 。

近年来 MRC 任务越来越复杂，具体的体现就是数据集越来越具有挑战性。对于早期的填空型任务，模

[1]本文中文章（Document）和段落（Passage）是同样的概念

型仅仅需要阅读一篇文章，甚至只需某个相关的上下文句子即可正确的填写出问题中的单词。经过近几年的 MRC 领域的快速发展，目前已经有很多数据集既需要模型从多篇文章中多步推理又要同时给出不限于文章中单词的答案，任务的难度显著上升。本文按照 Chen[3] 提出的根据答案形式的不同将阅读理解任务概括为 4 类任务：填空式、多项选择式、抽取式和自由答案式。下面对这四种类型任务分别进行叙述并介绍相关的数据集。

2.1 填空式

填空式阅读理解是指给定一篇文章 D 和一个与文章相关的问题 Q，Q 是通过删除掉句子中某一个单词构成，要求模型根据 D 能够正确的填写出 Q 缺失的单词 a，且 $a \in D$。填空型数据集的一个样例见表 1。

CNN&Daily Mail[16] 数据集是由 Google DeepMind 和牛津大学发布于 2015 年，这是第一个较大规模的完形填空型阅读理解型数据集。从 CNN[2]中收集 93k 篇文章，从 Daily Mail[3]上收集 220k 篇文章。删去句子中的一个命名实体单词，以此作为问题，构建了 (文章-问题-答案) 的三元组形式的语料库作为填空式的阅读理解任务。

Hill 等人[17] 发布了（Children's Book Test，CBT）数据集，语料库来源于儿童读物（Gutenberg[4]工程）。每一篇文章是由 20 个连续的句子构成，删除第 21 个句子中的某个单词作为问题。与 CNN&Daily Mail 的不同之处是删除的单词不局限于命名实体，还可能删除句子中的名词、动词和介词。

CLOTH[59] 数据集是收集自中国中学生的英语考试试题中的完型填空题型。与之前的填空型数据集最大的不同是问题中删除的单词不像之前数据集的构建方式那样是由系统根据规则自动构建的，这样生成的问题往往没有目的性并且答案可能会出现歧义。而是由教师为了测验学生英语水平而精心设计的，空白处位置的单词通常会考察学生的词汇、语法以及推理能力，因此 CLOTH 数据集相比于 CNN&Daily 和 CBT 更具有挑战性。

表 1　CNN&Daily Mail[16] 数据集的一个样例
Table 1 An example of CNN&Daily Mail dataset

文章：	What was supposed to be a fantasy sports car ride at Walt Disney World Speedway turned deadly when a Lamborghini crashed into a guardrail. The crash took place Sunday at the Exotic Driving Experience, which bills itself as a chance to drive your dream car on a race-track. The Lamborghini' s passenger, 36-year-old Gary Terry of Davenport, Florida, died at the scene, Florida Highway Patrol said. The driver of the **Lamborghini**, 24-year-old Tavon Watson of Kissimmee, Florida, lost control of the vehicle, the Highway Patrol said. (...)
问题：	Officials say the driver, 24-year-old Tavon Watson, lost control of a_____
答案：	Lamborghini

2.2 多项选择式

多项选择式这类问答任务是对于给定的文章 D 和问题 Q 以及多个候选答案 $A = \{A_1, A_2, \cdots, A_n\}$，从中选择正确的答案，即建模概率：$P(A_i|D, Q)$，其中 $A_i \in A$。相关的数据集如 MCTest[43] 和 RACE[28]。

[2]www.cnn.com

[3]www.dailymail.co.uk

[4]https://www.gutenberg.org/

多项选择型数据集的一个样例见表 2。

MCTest 是一个早期提出来的多项选择型数据集，问题形式为四选一。数据集收集自儿童故事语料库，类似这种基于故事文章的语料库构建数据集的还有上面提到的 CBT[17]。但是由于其规模比较小仅仅包含 500 篇故事文章很难利用神经网络模型来学习，因此 MCTest 通常用来作为验证集或测试集。

RACE[5]数据集是从中国中学生的英语考试题中的阅读理解题型建立的数据集。共有将近 28000 篇文章以及 100000 个问题，这些问题和候选答案都是由出题专家生成的，更加的接近真实世界的语义。RACE 数据集涵盖多个领域如新闻、故事、广告、传记等等，由于其类型的多样性因此可以更好的评估机器的阅读理解能力。

表 2　RACE[28] 数据集的一个样例

Table 2 An example of RACE dataset

文章：	Runners in a relay race pass a stick in one direction. However, merchants passed silk, gold, fruit, and glass along the Silk Road in more than one direction. They earned their living by traveling the famous Silk Road. .. The Silk Road was made up of many routes, not one smooth path. They passed through what are now 18 countries. The routes crossed mountains and deserts and had many dangers of hot sun, deep snow and even battles...
问题：	The Silk Road became less important because____
选项：	A. it was made up of different routes B. silk trading became less popular **C. sea travel provided easier routes** D. people needed fewer foreign goods
答案：	C

2.3 抽取式

这类阅读理解任务是 MRC 领域较为流行的研究方向，主要原因在于从数据集的构建、评测指标以及应用价值等角度上看抽取式阅读理解是最合适的。给出文章 D 和问题 Q，问题的答案是 D 中的一段连续的单词构成，答案的长度不固定，可以表示为 $P(A|D,Q)$，其中 $A=\{t_i,t_{i+1},\cdots,t_{i+k}\}(1\leq i\leq i+k\leq n)$，$n$ 代表 D 中单词的个数。片段选择型问答任务数据集较其它类型任务的数据集较多，常用的数据集如 SQuAD[41]、NewsQA[50]、TriviaQA[24] 等。片段选择型的一个样例见表 3。

SQuAD 是 MRC 领域最为广泛使用的数据集之一，它的提出极大地推动了 MRC 领域的发展。数据集由众包工人根据维基百科上面的文章给出问题，答案来源于文章中某段连续的文本，长度并不固定。SQuAD 1.1 含有 536 篇文章，总计十万多个问题-答案对。NewsQA 数据集很类似于 SQuAD，区别在于其文章来源于 CNN 新闻并且在 NewsQA 中某些问题是没有答案的，这也使得后来 Rajpurkar 等人在 SQuAD 版本上又增加了五万个不可回答的问题构建了数据集 SQuAD 2.0[40]。对于这些带有不可回答问题的数据集，模型首先要清楚问题是否可以根据文章回答，在可回答的前提下给出答案，因此要求模型对文章的理解要更加的深刻。

TriviaQA 数据集的构造方式不同于前面数据集的构造方式。之前的数据集都是给定文章后，由人工构造出与文章相关的问题和答案，但是现实世界中人们通常是先提出问题然后搜寻相关的文章再找到答案。

[5]www.cs.cmu.edu/glail/data/race/

基于这一思想，Joshi 等人首先从 trivia 上收集大量的问题-答案对，然后为每一个问题从网页上或者维基百科上搜索出相关的文章，这些文章就是答案的依据。最后构建出 65 万多个（问题-答案-文章）三元组，更重要的是这种由问题找文章的数据集构造方式使得问题和文章在句法和词汇上都有着较大的差异性，这使得数据集难度更高。

尽管上面的数据集难度越来越高，但是它们整体上来说对模型的推理能力要求不高，主要原因在于回答这些数据集的问题往往只需要集中于某个句子的上下文或者在一篇段落上推理即可。为了提高模型的多步推理能力，Yang 等人[61] 发布了 HotpotQA 数据集，每一个问题对应多个段落，问题的答案往往需要在多个段落上逐步推理才能获得，同时要求模型预测回答问题所必须的线索句子（supporting facts）类似的数据集还有 WIKIHOP[58]，每一个样本来源于知识库中的三元组，要求模型从多篇维基百科文章中推理然后从多个候选中选出正确的答案。

表 3　SQuAD 1.1[41] 数据集的一个样例
Table 3 An example of SQuAD 1.1 dataset

文章：	In 1870, Tesla moved to Karlovac, to **attend school at the Higher Real Gymnasium**, where he was profoundly influenced by a math teacher Martin Sekulić. The classes were held in German, as it was a school within the Austro-Hungarian Military Frontier. Tesla was able to perform int egral calculus in his head, which prompted his teachers to believe that he was cheating. He fini- shed a four-year term in three years, graduating in 1873.
问题：	Why did Tesla go to Karlovac?
答案：	attend school at the Higher Real Gymnasium

2.4 自由答案式

简单的从文章中摘取一段文本可能并不能回答问题需要的答案，自由答案型阅读理解任务的答案是自由形式的，不局限于文章中的某些单词，语法上往往是更加的灵活。可以表示为 $P(A|D,Q)$，其中 $A \subseteq D$ 或 $A \nsubseteq D$。从文章中概括提炼出问题的答案也是更加的符合人类的阅读方式的，基于这些原因，自由回答式问答的数据集也因此公布出来并且受到广泛的关注，相关的数据集如 MS MARCO[34]、DuReader[14] 和 NarrativeQA[27]，自由答案型数据集的一个样例见表 4。

MS MARCO 是由微软通过在必应搜索引擎的日志上收集用户提出的问题，对于文本段落是来源于必应搜索引擎的返回的搜索结果。具体的就是对于每一个问题，给出 10 个最相关的查询结果的文本段落，然后由标注人员从这 10 个文本段落中找出那些与这个问题有关的文本段落，人工的从这些选择出来的段落中概括提炼出答案，同时对于选出来的段落要标记为 is_select=1，表示这个段落和答案相关，从而可以训练模型。如果不能从给出的 10 个段落中推理出答案，那么这个问题就标记为不可回答的问题，同样要保留在数据集中，目的就是让模型能够判别出问题是否可以回答。

DuReader 是中文 MRC 数据集，类似于 MS MARCO 数据集的构造方式，问题和文章取自百度搜索和百度知道。答案同样是人工生成的，一个问题会给出 5 个相应的文本段落，很多问题需要推理多篇文章段落才能得到答案，甚至一些问题存在多个答案。MS MARCO 和 DuReader 数据集的问题都是收集自搜索引擎中用户提出的问题，因此这两个数据集也更加的贴近真实场景。

Kovcisky 等人认为现在 MRC 领域大部分数据集的问题过于肤浅，而且答案往往只关注上下文信息，因此很多问题仅仅通过浅层的模式匹配就可以找到答案，他们发布的 NarrativeQA 数据集，收集自小说和

电影剧本，要求模型在理解整部小说或剧本的前提下才能回答问题，要求模型需要有更强的理解能力和推理能力。

表 4　MS MARCO[34] 数据集的一个样例
Table 4 An example of MS MARCO dataset

文章 1:	Rachel Carson’ s essay on The Obligation to Endure,is a very convincing argument about the harmful uses of chemical, pesticides, herbicides and fertilizers on the environment.
.....	
文章 5:	Carson believes that as man tries to eliminate unwanted insects and weeds; however he is actually causing more problems by polluting the environmen with, for example, DDT and harming living things
.....	
文章 10:	Carson subtly defers her writing in just the right writing for it to not be subject to an induction run rampant style which grabs the readers interest without biasing the whole article.
问题:	Why did Rachel Carson write an obligation to endure?
答案:	Rachel Carson writes The Obligation to Endure because believes that as man tries to eliminate unwanted insects and weeds; however he is actually causing more problems by polluting the environment.

2.5　评估方法

对于不同的 MRC 任务有不同的评估指标。对于填空型任务与多项选择型任务都是属于客观题型，用准确率就可以衡量模型的性能。例如对于测试集合中的所有问题 $Q=\{Q_1,Q_2,\cdots,Q_m\}$，其中 m 代表问题的个数。如果模型预测出来的 m 个答案中有 n 个是正确的，那么模型的准确率是 n/m。

片段选择型任务属于半客观题型，通常用精确匹配 EM（Exact Match）和 F1 分数来评估模型。精确匹配 EM 评估指标可以看做是准确率的扩展，就片段选择型任务来讲，EM 要求预测出来的所有单词要和标准答案的所有单词要完全一致，EM 值才为 1，否则为 0。

F1 值的计算方式是一种模糊匹配，它是精确率和召回率之间的调和平均数。精确率是指模型预测的答案中有多大比例的单词是标准答案中的单词。召回率是指标准答案中的单词有多大比例在预测答案中出现。

对于自由答案型任务由于其答案形式不固定，一般采用单词水平的匹配率作为评分标准，常用标准用 ROUGE-L[31] 和 BLEU[36]。ROUGE-L 用来计算标准答案和预测答案的最长公共子序列（Longest Common Subsequence，LCS），BLEU 最初用于评估翻译性能，在应用到 MRC 任务上主要用来衡量预测答案和真实答案之间的相似性。表 5 列举了本章介绍的所有数据集。

3　神经机器阅读理解模型

Hermann 等人[15] 发布的 CNN&Daily Mail 数据集以及他们所设计的两个基于神经网络和注意力机制的模型可以看作是 MRC 领域的奠基性工作，开创了神经机器阅读理解模型。Rajpurkar 等人 [41] 在 2016 年发布的 SQuAD 数据集是 MRC 领域里程碑式的数据集，在 2016-2018 年期间掀起了一阵热潮，很多的神经机器阅读理解模型都是在此期间构建出来的。填空式数据集本质上可以认为是抽取式数据集的简化形

表 5　MRC 常用数据集对比，Acc 代表准确率

Table 5 Comparison of common dataset in MRC

数据集	发布时间	文章来源	文章类型	问题特征	答案类型	评估指标
CNN&Daily Mail[16]	2015	新闻	单段落型	自动生成	填空式	Acc
CBT[17]	2015	儿童读物	单段落型	自动生成	填空式	Acc
CLOTH[59]	2016	英语考试	单段落型	人工生成	填空式	Acc
MCTest[43]	2013	儿童读物	单段落型	众包生成	多项选择式	Acc
RACE[28]	2018	英语考试	单段落型	人工生成	多项选择式	Acc
SQuAD[41]	2016	维基百科	单段落型	众包生成	抽取式	EM/F1
TriviaQA[24]	2017	网页搜索	多段落型	收集于网站上的问题-答案对	抽取式	EM/F1
SQuAD 2.0[40]	2018	维基百科	单段落型	新增无答案问题	抽取式	EM/F1
NewsQA[50]	2017	新闻	单段落型	众包生成，包含无答案问题	抽取式	EM/F1
HotpotQA[61]	2018	维基百科	多段落型	众包生成，需多步推理	抽取式	EM/F1
WIKIHOP[58]	2018	维基百科	多段落型	自动生成，需多步推理	抽取式	EM/F1
MS MARCO[34]	2016	搜索引擎	多段落型	收集于搜索引擎	自由答案式	ROUGE-L/BLEU
DuReader[14]	2018	搜索引擎	多段落型	收集于搜索引擎	自由答案式	ROUGE-L/BLEU
NarrativeQA[27]	2017	小说和电影剧本	多段落型	众包生成	自由答案式	ROUGE-L/BLEU

式，而后续的很多任务如对话形式、开放领域形式、多段落形式的阅读理解任务也都是在抽取式任务的形式上设计模型。因此抽取式阅读理解任务是 MRC 领域的核心，本章主要以抽取式任务的 MRC 模型为出发点，安排如下：3.1 节分析经典的基于抽取式任务的 MRC 模型通用架构，3.2 节介绍复杂任务下的 MRC 模型，3.3 节介绍目前流行的预训练模型以及如何利用预训练模型设计性能更强大的 MRC 模型。

3.1　基于抽取式任务的经典 MRC 模型

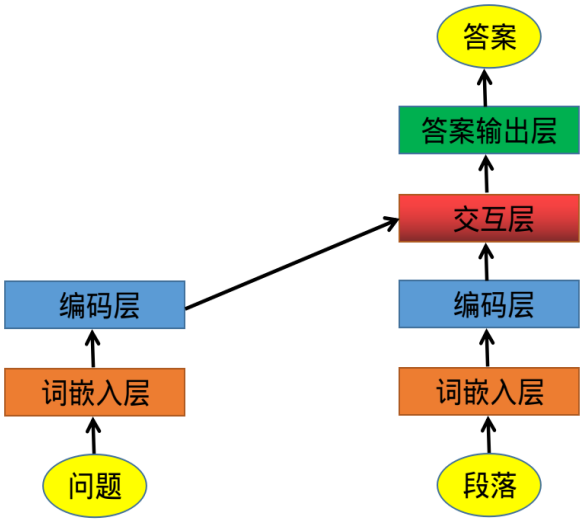
想让机器能够阅读理解文本需要解决以下几个问题：

1. 如何将段落和问题这种文本形式的无结构数据表示为计算机可以处理的形式；
2. 如何根据问题检索出段落中与问题最相关的部分；
3. 如何从检索出来的文章片段中归纳得到答案。

用于 MRC 任务的深度学习模型的整体框架主要包括如下几个层：词嵌入层、编码层、交互层、答案输出层，如图 1 所示。词嵌入层的作用是将段落和问题嵌入到低维的向量空间中，用每一个向量表示每一个单词。编码层的作用是编码段落和问题中单词的语义信息，使得每一个单词可以关注到它的上下文。交互层的作用是将段落的语义信息与问题的语义信息融合，让模型学习到段落中与问题最相关的部分。答案输出层的作用是从段落中查找出问题的答案。

3.1.1　词嵌入层

如何将文本有效的表示成计算机可以处理的形式同时可以有效地利用单词之间的语义一直的 NLP 领域的重点问题。早期的 one-hot 形式编码用一个二值向量表示单词，但是存在数据稀疏并且随着单词个数的增加出现维度灾难的问题，此外这种形式的编码也不能够表示出单词之间的语义关系。



Rumelhart 等人[44] 最早提出分布式表示的概念，分布式表示是将单词用一个低维度的稠密向量表示，即将单词嵌入到一个低维向量空间中，因此这种表示方式也叫词嵌入。语义相近的单词在向量空间中距离也相近，因此这种词表示方法解决了 one-hot 编码的很多问题。Bengio 等人 [**NNLM**] 最早将深度学习的思想融入到语言模型中提出神经网络语言模型（Neural Network Language Model，NNLM）模型，模型的第一层映射矩阵就是学习到的词向量，Mikolov 等人[33] 受到这种思想的启发提出 Word2Vec。Word2Vec 提出两种模型 CBOW 和 Skip-gram 来学习单词的分布式表示，CBOW 使用中心词的上下文来预测这个单词而 Skip-gram 利用中心词来预测其周围的单词。但是无论是 CBOW 还是 Skip-gram 都只是考虑了单词局部上下文的信息，GloVe[37] 利用单词共现矩阵考虑了全局统计信息。

大量实验表明利用 Word2Vec 或者 GloVe 预训练好的词向量作为下游任务文本的词表征来初始化下游任务模型的第一层可以显著地提升模型的效果。除了词嵌入方法外，还有很多细粒度的嵌入方式。如 Seo 等人[46] 提出在词嵌入的基础上结合单词的字符嵌入，以缓解 NLP 领域常见的 OOV（out-of-vocabulary）问题。Chen 等人[5] 提出引入单词的语义特征来增强嵌入表示，如段落单词与问题单词之间的完全匹配特征、词性特征以及单词的命名实体特征等。

然而 Word2Vec 和 GloVe 训练出来的词向量是静态的词向量，即训练好模型后一个单词的表示向量就是固定的，没有考虑上下文的信息，因此无法解决多义词问题。为了解决这个问题，Peters 等人 [38] 提出一种动态的基于上下文的词嵌入模型 ELMo，每一个单词的词向量都是根据它所在的上下文语义表示的，很好的解决了一词多义的问题。关于 ELMo 以及预训练模型的细节见3.3.2节。

从早期的 one-hot 形式编码到分布式表示技术最后到基于上下文的词嵌入技术，每一种技术的出现都证明了一个好的文本表示方法可以极大地提升模型的性能。

3.1.2 编码层

这一层的目的是在词嵌入层的基础上通过对词嵌入层的输入文本做特征提取，进一步获得句子层面的语义信息。NLP 领域最为常用的特征提取器是基于循环神经网络（RNNs）的变体如 LSTM[18] 和 GRU[6] 等，因为这种循环结构适合处理文本这类序列数据，绝大部分的 MRC 模型编码层都是利用 RNNs 作为特征提取器。但也正是这种序列式的结构使得计算不能并行，训练耗时，更重要的是由于梯度消失所以不能解决单词之间长距离依赖问题，使得其特征提取能力始终受限。Vaswani 等人[51] 提出了一种用于机器翻译的 encoder-decoder 结构 transformer，舍弃了常用的循环神经网络结构，完全的基于自注意力机制构建模

型，实验表明 transformer 的特征提取能力强于循环神经网络而且可以并行计算加快训练。文献 [63] 提出一种网络模型 QANet，不像之前的那些模型几乎都是用 RNNs 来做编码器，QANet 提出一种新颖的编码结构，利用卷积结合 transformer[51] 中的多头注意力结构。卷积方式采用的是文献 [26] 提出的深度可分离卷积（depthwise separable convolutions），相比传统的卷积计算方式深度可分离卷积可以减少运算次数。整个结构的思想是先利用卷积操作建模局部特征的交互，再用自注意力机制建模全局交互，实验结果表明这种架构不仅加快训练速度同时在 SQuAD 数据集上模型性能优于那些利用 RNNs 作为编码器的模型。关于 transformer 的细节介绍见3.3.1节。

3.1.3 交互层

交互层是整个网络模型中关键的一层，前面的编码层输出的是问题和文章中每个单词的上下文语义编码，每个单词关注了自己所在句子的上下文单词，但是却并没有关注对应的句子。而我们在做阅读理解问题时，通常是带着问题去文章中找答案，我们要知道文章中每一个单词和问题之间的相关度。因此交互层的目的就是让文章的语义信息与问题的语义信息融合，以此达到对文章更深层次的理解，而交互层中最常用的方法就是注意力机制。

注意力机制可以被视为是一个查询向量（query）和一组键值对向量（key-value pairs）的映射过程。整个过程首先是利用函数 f 衡量 query 和 key 之间的相似度，生成一个权重分数向量，然后将权重分数向量归一化（通常利用 softmax 函数）后对 value 加权求和得到的结果就是 query 对 key-value pairs 的注意力。具体计算公式形式如下：

$$\begin{aligned} \alpha_i &= \text{softmax}(f(Q, K_i)) \\ \text{Attention}(Q, K, V) &= \sum_{i=1}^{n} \alpha_i V_i \end{aligned} \tag{1}$$

其中 (K_i, V_i) 代表 key-value pairs 中的第 i 个值，函数 f 常采用计算方式有内积函数、二次型函数、前馈神经网络，双维度转换函数，分别见如下公式：

$$f(p_i, Q) = p_i^T Q \qquad \text{内积函数} \tag{2}$$

$$f(p_i, Q) = p_i^T W Q \qquad \text{二次型函数} \tag{3}$$

$$f(p_i, Q) = v^T \tanh(W p_i + U Q) \qquad \text{前馈神经网络} \tag{4}$$

$$f(p_i, Q) = p_i^T W^T U Q \qquad \text{双维度转换函数} \tag{5}$$

在 NLP 领域中 $K = V$，简单的来说就是两个序列中其中一个序列为另一个序列的每一个位置生成一个权重值，这个值代表当前位置的单词对另一个序列的重要性。如果是自注意力（self attention），那么此时 $Q = K = V$，目的是计算序列中某个单词和其它单词之间的相关性从而增强自身的语义表示。Bahdanau 等人[2] 最早将注意力机制应用在机器翻译领域，获得了极大的反响，为 NLP 领域的其它任务的模型提供了启发式的思想。

MRC 模型做交互注意力运算有两个方向，即从问题到段落（Question-to-Context,Q2C），从段落到问题（Context-to-Question，C2Q）这两个方向。从 Q2C 的注意力是指将问题看做是 Q，段落看做 K, V。利用问题去和文章做注意力计算，定义 $C = [c_1, c_2, \cdots, c_n] \in R^{n \times d}$ 代表段落的表示向量，其中 n 代表段落

长度，d 代表向量维度，$Q \in R^d$ 代表整个问题的表示向量 Q2C 的注意力计算步骤如下：

$$\alpha_i = \text{softmax}(f(c_i, Q))$$
$$\text{Attention}(C, Q) = \sum_{i=1}^{n} \alpha_i c_i$$

C2Q 注意力类似，此时段落看作是 Q，问题看做 K, V。

上面的式子是将问题压缩成一个固定维度的向量，得到的注意力权重 α 也是一维的，因此也称为一维注意力。一维注意力方法所关注的是问题序列的整体对文章的注意力，没有考虑问题序列的不同单词之间对文章的关注程度差异。与其相对应的是二维注意力，即对于问题序列中的每一个单词都会和段落做注意力计算，得到的注意力权重是二维向量。

C2Q 和 Q2C 这两种都属于交互的计算注意力，然而这种注意力机制可能导致只重视文章中与问题相关度高的单词，而忽视了文章所强调自身的语义信息。在文章上利用自注意力机制则可以看做是反复的阅读文章，从而加深对文章语义信息的理解。

如果按照注意力的计算次数上区分，又可以分为 one-hop 和 multi-hop 形式。one-hop，也叫“单跳结构”是指仅仅通过一次计算得到注意力权值然后加权求和得到注意力结果，这也是一种静态的计算形式。与之对应的是 multi-hop，也叫“多跳结构”。one-hop 形式下仅仅只做一次交互计算，而注意力机制虽然可以提取相关的重要信息，但是它仍然是基于浅层语义信息的相似度计算。在机器阅读理解任务中，对于复杂的问题通常是不能在一个句子中找出答案，需要多步推理才能寻找答案，如表 6 所示

表 6　WIKIHOP[58] 多跳推理的样例

文章 1:	The Hanging Gardens, in [Mumbai], also known as Pherozeshah Mehta Gardens, are terraced gardens ⋯ They provide sunset views over the [Arabian Sea] ⋯
文章 2:	Mumbai (also known as Bombay, the official name until 1995) is the capital city of the Indian state of Ma It is the most populous city in India ⋯
文章 10:	The Arabian Sea is a region of the northern Indian Ocean bounded on the north by Pakistan and Iran, on the west by northeastern Somalia and the Arabian Peninsula, and on the east by India ⋯
问题:	Hanging gardens of Mumbai, country,?
答案:	Iran, India, Pakistan, Somalia

我们可以看到想要得到最终的答案需要在多个段落中进行多次推理，在每一个推理过程中都会变换注意力关注的对象，显然 one-hop 结构是不能实现多步推理的。

鉴于目前多数模型交互层所使用的注意力机制较为复杂，很难按照上述形式完全的区分开每一个模型，本文按照 Liu 等人 [**Survey**] 的思路按照注意力计算的方向以及次数划分各个模型。

单向注意力　Hermann 等人 [15] 最早利用神经网络模型并且融入注意力机制做 MRC 任务。文中提出两种不同的单向注意力机制 Attentive Reader 和 Impatient Reader，均是计算问题到文章的注意力。Attentive Reader 是将问题表示为一个固定长度的向量然后与文章中每一个单词做注意力计算，然后利用注意力权重对文章中的单词的向量表示加权求和得到一个固定长度的向量即为注意力运算后的结果，然后与问题联合预测答案，其中注意力的运算方式采用前馈神经网络（公式 4）。Chen 等人 [4] 在 Attentive Reader 的基础上利用双线性项（公式 3）取代原有的前馈神经网络（公式 4）并且直接将对段落加权求和后得到的向量作为预测答案的输入而不是联合问题 Q 的语义信息，实验证明这种简化反而提高了模型的准确度。文献 [53]

提出一种 Match-LSTM 模型，与之前的模型不同，Match-LSTM 计算的方向是段落到问题的注意力，将问题的语义信息融入到 Match-LSTM 中。具体计算过程如下：

$$
\begin{aligned}
s_t &= v^T \tanh(W^Q H^Q + W^P h_t^P + W_r h_{t-1}^r) \\
\alpha_t &= \text{softmax}(s_t)
\end{aligned} \tag{6}
$$

其中 H^Q 是问题通过编码层的输出，h_t^P 是段落的第 t 个单词通过编码层的输出，h_{t-1}^r 是 Match-LSTM 上一时刻的隐藏状态。α_t 是段落的第 t 个单词与问题的每一个单词之间的注意力权重。计算得到的注意力权重对问题的语义表示加权求和，然后与段落当前时刻单词的上下文表示拼接作为 Match-LSTM 当前时刻的输入。

$$
\begin{aligned}
z_t &= [h_t^p; \alpha_t H^q] \\
h_t^r &= \text{LSTM}(z_t, h_{t-1}^r)
\end{aligned} \tag{7}
$$

此外为了使得段落从后向前的对问题做关注，模型将段落序列翻转再次按照上述方式计算，最后将两个方向的计算结果拼接作为交互层的输出。

双向注意力　上述的模型全都属于单方向注意力，要么仅计算 C2Q 方向的注意力或者仅计算 Q2C 方向的注意力。Xiong 等人 [60] 提出 Dynamic Co-attention Network（DCN）模型，在交互层中采用协同注意力机制，协同注意力同步的计算文章对问题的注意力以及问题对文章的注意力。最后按照公式 (8) 融合两个方向的注意力作为交互层的输出。

$$
\widetilde{C} = \beta[Q, \alpha C] \tag{8}
$$

其中 α 和 β 分别表示段落与问题之间的注意力权重，$\widetilde{C}$ 同时融合了问题的语义信息和段落的语义信息。文献 [46] 提出（Bidirectionl Attention Flow，BiDAF）模型。同样计算两个方向（C2Q 和 Q2C）的注意力，但是与之前模型不同的是 BiDAF 将之前的段落语义表示和交互层计算得到的问题感知的段落语义表示一起流向后面的层，这样一定程度上避免了过早的对段落语义信息概括而导致信息的损失。模型的简化实验表明 C2Q 方向的注意力对模型的重要性大于 Q2C 方向的注意力，一种可能的原因是由于问题序列的长度小于段落文本的长度所以计算得到的段落感知的问题语义向量的信息不够充分。

单跳结构　单跳结构是指段落与问题仅仅通过一次计算得到注意力权值然后加权求和得到注意力结果，要么是将问题整体压缩为一个向量与段落计算一次注意力，如 Attentive Reader[15]，AS Reader[25] 等，或者问题与段落的的整体表示采用并行化的计算方式，如 DCN[60]，BiDAF[46]，QANet[63] 等。

多跳结构　多跳结构可以视为单跳结构的堆叠，目的是通过多次计算段落与问题的交互信息加深模型对段落和问题的理解，从而达到多步推理的目的。实现多步推理这种机制通常有以下几种方式：

第一种方式是基于之前时间步所计算得到的问题感知的段落语义信息计算下一时间步的段落和问题交互，如 Impatient Reader[15]，并不是像 Attentive Reader 将问题表示为一个固定长度的向量，而是对于问题中的每一个单词都要和整个段落做注意力计算，而且计算的结果要和下一个单词以及段落共同做注意力计算，最后一个单词的注意力结果作为整个 Impatient Reader 计算注意力过程的输出。这种方式类似于人在阅读过程中不断的在问题和文章之间做关注。

第二种方式是利用 RNNs 这种基于上一时刻隐藏状态更新下一时刻隐藏状态的循环特性来达到多步推理，Sordoni 等人[48] 提出 Iterative Attention Reader（IA Reader）模型，利用 BiGRU 存储每一次迭代计算得到的问题和段落的交互信息。在每一时间步上，首先利用上一次的 BiGRU 的状态与问题做一维注意力匹配提取出问题的语义信息，然后再结合上一次的 BiGRU 的状态与段落再做一维注意力匹配从而提取出段

落的语义信息。将问题与段落的语义信息通过各自的门控单元作为 BiGRU 当前时刻的输入，其中门控单元采用前馈神经网络用来解决当前时间步下问题和段落的语义信息提取不充分的问题。Shen 等人[47] 提出一种动态决定推理次数的模型 ReasoNet，不同于 IA Reader 模型在整个推理过程中有着固定的推理次数。这种固定推理次数的缺点就是不考虑问题的复杂性，对于复杂的问题往往需要模型多次的推理，因此不同题目难度需要不同的推理次数，应当让模型学会什么时候终止推理。为了达到这一目的，ReasoNet 模型利用一个终止门产生二元值输出来动态的决定是否继续推理。ReasoNet 模型大致分为外部记忆单元模块、内部控制器模块、终止门模块以及答案输出模块。具体的，将段落和问题通过 Bi-GRU 编码后的语义表示作为外部的记忆单元 M，利用内部控制器（采用 GRU）当前时刻的状态与 M 做二维注意力匹配，得到注意力结果输入到内部控制器中更新内部控制器的状态。终止门模块以当前时刻内部注意力的状态作为输入来判断是否需要继续推理。由于产生了二元离散输出值，使得模型不能用梯度下降法训练，因此模型引入强化学习机制训练。

第三种方式通过堆叠多个计算注意力的层数达到多步推理的目的，Dhingra 等人 [11] 提出 Gated Attention Reader（GA Reader）模型，类似于 IA Reader 模型，同样采用 BiGRU 作为编码模块实现多跳结构。在每一步的推理过程中，首先通过 BiGRU 得到问题的语义信息，然后对段落的每一个单词做注意力的计算得到问题感知的段落表示，同时采用点乘计算的门控机制建模问题感知的段落表示和原来的段落语义向量之间的交互关系，目的是利用问题更新文章的语义表示。这种处理过程类比于带着问题反复的阅读文章，每一次都加深对文章的语义理解。Wang 等人 [55] 提出一种带有门控机制的注意力循环神经网络以及自注意力机制联合的交互层设计模型 RNet。RNet 在交互层的设计分为两部分。第一部分是带有门控机制的注意力循环神经网络，整体计算方式类似于 Match-LSTM，而且额外加入了门控机制使得模型可以有选择的输出语义信息。具体的，在公式（7）中的 z_t 上添加一个门控单元：

$$
\begin{aligned}
g_t &= \text{sigmoid}(W_g z_t) \\
z_t^* &= g_t \odot z_t
\end{aligned} \tag{9}
$$

其中 $\odot$ 表示元素之间的点乘。通过添加门控单元使得模型可以有选择的决定哪部分作为重要的语义信息输出。这种机制类似于人在阅读过程中要忽略段落中那些与问题无关的信息，凸显出重要的信息才能更加准确的找到答案。第二部分是利用自注意机制对段落的语义信息再次交互建模，通过自注意机制可以使得段落中每一个单词关注到其余所有的单词，使得模型对段落达到更深层次的理解。之前的模型在交互层利用注意力机制融合段落和问题时都是利用句子的高层级别的语义信息而忽略了句子在低层次级别的语义信息如单词级别的词嵌入等。Huang 等人[22] 提出 FusionNet 模型，将每一个单词在第一层到后面所有层的向量表示拼接成一个向量，原文中称为单词历史（history of word），因为它包含了一个单词所有层的语义编码。但是随着层数的增加维度会变得越来越大，为了解决维度问题同时不损失单词的历史信息，FusionNet 提出全关注注意力机制的概念：即利用段落和问题的单词历史计算得到注意力权重，然后对问题的某一层语义向量加权求和。这种机制使得两个输入向量可以互相关注到对方的历史信息同时压缩维度，文中对注意力权重的计算方式如下：

$$
\alpha_i = \text{ReLU}(Up_i)^T D\text{ReLU}(Uq_j) \tag{10}
$$

其中 $p_i \in R^d$ 和 $q_j \in R^d$ 分别代表段落第 i 个单词和问题第 j 个单词的单词历史，U 和 D 是训练的参数。Hu 等人[19] 认为在多层架构中，当前层的注意力计算并没有直接考虑到之前层计算得到的注意力信息，这可能导致两个不同但是相关的问题：（1）多层注意力分布集中在相同的文本上导致注意力冗余；（2）多层注意力未能集中在文本的重要部分造成注意力缺乏。针对这两个问题他们提出强化助记阅读器（Reinforced Mnemonic Reader，RMR）模型，利用重关注机制，通过直接利用之前层计算的注意力信息来微调当前层注意力分布的计算。

表 7 对比了本节介绍的经典的基于抽取式任务的 MRC 模型之间的差异。其中 Q2C 代表问题到段落注意力，C2Q 代表段落到问题注意力，Bidirectional 代表双向注意力，self-attention 代表对段落做自注意力运算，one-dim 代表一维注意力，two-dim 代表二维注意力，one-hop 代表单跳结构，multi-hop 代表多跳结构。

表 7　基于注意力机制的模型对比

Table 7 Comparison of models based on attention mechanism

模型	注意力方向	注意力维度	推理模式
Attentive Reader[15]	Q2C	one-dim	one-hop
Impatient Reader[15]	Q2C	two-dim	multi-hop
Standford Reader[4]	Q2C	one-dim	one-hop
AS Reader[25]	Q2C	one-dim	one-hop
IA Reader[48]	Q2C	one-dim	multi-hop
GA Reader[11]	C2Q	two-dim	multi-hop
Match-LSTM[53]	C2Q	two-dim	multi-hop
DCN[60]	Bidirectional	two-dim	one-hop
BiDAF[46]	Bidirectional	two-dim	one-hop
ReasoNet[47]	Bidirectional	two-dim	multi-hop
R-Net[55]	C2Q+self-attention	two-dim	multi-hop
RMR[19]	Q2C+self-attention	two-dim	multi-hop
QANet[63]	Bidirectional	two-dim	one-hop

3.1.4 答案预测层

这是整个模型架构的最后一层，用来输出预测的答案。MRC 任务按照答案形式的不同大致分成四类，因此这一层的设计需要考虑到答案形式。下面介绍各个模型在四类不同的 MRC 任务上的输出层设计。

1）填空式：这类任务答案的形式是预测问题中缺失的单词，而且缺失的答案来源于文章中。Hermann 等人 [15] 最早提出将问题的语义向量与问题感知的段落语义向量拼接成一个向量然后映射到整个词典中预测那个缺失的单词。这种方法存在的一个问题就是不能够确保预测的单词一定是段落中的词汇，这就使得模型的预测准确率受到影响。指针网络（Pointer networks[52]）模型由 seq2seq 模型演变而来，主要就是为了解决输出源自于输入的问题，实现方式是利用计算的注意力的权重分布直接输出预测结果，而这种机制正适合填空型任务以及片段选择型任务。Kadlec 等人[25] 提出 AS Reader 模型正是受到指针网络的启发，对于计算得到的注意力权重分布，将其中相同单词的注意力权值相加，最后输出具有最大权值的单词最为答案。填空式任务模型的损失函数可以写为 $L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log P_{y_i}$。其中 θ 为模型参数，N 代表样本数目，y_i 表示段落中第 i 个样本在段落中标准答案的位置。

2）多项选择式：这类任务是从多个候选答案选项中选择正确的选项。处理这种任务最简单的一种方式

就是计算模型输出后的段落语义信息和选项之间的相似程度，相似程度最高的作为预测的选项，从而将问题变化为句子之间的语义匹配问题。Wang 等人 [54] 提出将问题、段落、选项一起放在模型中做交互计算输出一个向量作为输出层的输入，输出层采用简单的输出维度是 1 的全连接层，输出的值代表模型对这个选项的打分值，其它的选项类似的处理，值最高的选项作为预测的答案。最后对所有选项的打分值做归一化作为模型的损失函数。多项选择型任务模型的损失函数可以写为 $L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}\log P_{y_j^i}$。其中 θ 为模型参数，N 代表样本数目，m 代表选项个数，y_j^i 表示第 i 个样本中第 j 个选项是正确答案。

3）抽取式：这类任务是从文章中提取出来一段连续的单词作为答案，虽然类似于填空型任务输出来源自输入的性质，但是不像填空型任务仅仅只是预测一个单词。因此填空型任务答案输出层的设计不能直接用来作为片段选择型任务答案预测层。由于提取文本的长度不固定，使得这一任务更具有挑战性。Wang 等人[53] 受到指针网络的启发提出了两种基于指针网络的输出模型，第一种是序列式模型，利用指针网络以一种序列式的形式生成答案的每一个位置，处理过程类似于 seq2seq 模型的解码过程，这种模型下答案的每一个单词可能出现在文本段落的任何一个位置，这是因为指针网络并没有要求从输入中选择的输出具有连续性。由于答案的长度不固定，因此在段落中设置一个特殊的位置表示答案的终止点，当预测到这个位置时终止答案的生成。第二种是边界式模型，不同于序列式模型那样序列的生成答案的每一个位置，由于要预测的答案是一段连续的文本，因此可以利用指针网络仅仅预测答案的起始位置和终止位置。所预测答案的概率是预测这两个位置概率的乘积，这种方式相比于第一种更加的简单而且测试结果表明更加高效。

边界式模型的这种设计思想也被后来很多 MRC 模型采纳。尽管边界式模型简单有效，但是边界式模型有可能陷入局部极值的情况从而提取错误的文本片段。为了处理这个问题，Xiong 等人[60] 提出一种动态迭代的指针网络作为解码端，利用上一次预测的答案的起始位置和终止位置以及解码端当前的状态来重新评估下一次预测答案的起始位置和终止位置。多次迭代后选取所有迭代次数中概率最大的情形作为预测答案。抽取式模型的损失函数可以写为 $L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\log P_{y_i^s}^S + \log P_{y_i^e}^E$。其中 θ 为模型参数，N 代表样本数目，y_i^s 表示第 i 个样本中标准答案的起始位置在文章中的位置，y_i^e 表示第 i 个样本中标准答案的终止位置在文章中的位置。如果考虑到不可回答的问题，最简单的方式是额外在输出层加上一个输出维度是 1 的全连接层。此时的损失函数可以写为 $L(\theta) = -\frac{1}{N}\sum_{i=1}^{N}(\log P_{y_i^s}^S + \log P_{y_i^e}^E) + \log P_{y_i^u}^U$。其中 y_i^u 表示第 i 个样本中的问题是不可回答的问题。关于带有不可回答问题的阅读理解任务细节见3.2.1节。

4）自由答案型：这类任务的答案形式已经不再是原文中某段文本，而是需要根据文章和问题生成符合语法规范的文本。这类任务对答案生成模块的能力要求较高。处理生成任务典型的架构是 seq2seq 模型，See 等人[45] 提出一种指针生成网络模型（Pointer-Generator Network，PGNet），最早用在文本摘要领域，模型结合了 seq2seq 的生成机制以及指针网络的拷贝机制，使得模型既能从词典中生成单词又能在原文中拷贝单词，实验结果表明该模型的效果优于传统的 seq2seq 模型。

表 8 对比了经典的基于抽取式任务的 MRC 模型在 SQuAD[41] 数据集上的性能[6]。

3.2 复杂任务下的 MRC 模型

以上介绍了经典的神经机器阅读理解模型并且详细的对比了各个模型在交互层注意力机制的差异。这些模型大多是基于 SQuAD[41] 数据集设计的。Weissenborn 等人[57] 提出的 FastQA 模型，在编码层的输入中对段落的每一个单词额外的添加了两个特征（binary，weighted）：binary 特征表示原文中的词是否出现在问题中，weighted 特征表示原文中的单词与问题的相似度。FastQA 没有交互层复杂的注意力机制的设

[6]统计数据源自 Yu 等人[63]

表 8　模型在 SQuAD[41] 数据集上的对比（acc 代表准确率）

模型	EM/F1
Match-LSTM[53]	64.7/73.7
DCN[60]	66.2/75.9
BiDAF[46]	68.0/77.3
ReasoNet[47]	70.6/79.4
R-Net[55]	72.3/80.7
RMR[19]	73.2/81.8
QANet[63]	76.2/84.6

计，仅仅依靠这两个特征就在 SQuAD 数据集上取得了很好的效果。这一方面质疑那些复杂的注意力机制是否真的可以提升模型的效果，另一方面也说明 SQuAD 数据集难度不高，达不到测验模型推理和理解能力。而且抽取式的问答要求答案是原文连续的文本片段，显然不接近人类现实世界中的问答。本节介绍复杂任务下的 MRC 模型，与之前的模型不同，在应对复杂的阅读理解任务下模型需要根据任务的特点来设计相应的结构。

3.2.1　带有不可回答问题的阅读理解任务

之前的 MRC 数据集全都有一个共同的特点就是默认每一个问题都可以在给定的文本中找到答案，然而一段文本所包含的知识是限的，因此有下述两点是需要考虑的：（1）这段文本不能回答那些与文本表达内容无关的问题；（2）某些问题可能与文本内容类似但是问题含义与文本含义不同，这种问题仍然是不可回答的。目前最流行的带有不可回答问题的数据集如 SQuAD 2.0[40]，在 SQuAD 的基础上增加了五万多个不可回答的问题。一个样例如表 9 所示。

表 9　SQuAD 2.0 的一个样例
Table 9 An example of SQuAD 2.0

文章：	Other legislation followed, including the Migratory Bird Conversation Act of 1929, a **1937 treaty** prohibiting the hunting of right and gray whales, and the **Bald Eagle Protection Act of 1940**. These later laws had a low cost to society —the species were relatively rare—and little opposition was raised.
问题：	What was the name of the 1937 treaty
看似合理的答案：	Bald Eagle Protection Act

从表 9 中可以看出题目问的是 1937 treaty 的名字，而 Bald Eagle Protection Act 指的是 1940 treaty 的名字，这对于模型来说是一个非常迷惑的答案。对于这类任务模型必须区分出哪些问题是不可回答的，对于不可回答的问题模型不能再给出 “貌似合理” 的答案。在 3.2 节所介绍的模型里，很多模型在 SQuAD 数据集上表现很好然而在 SQuAD 2.0 数据集上效果显著下降，这说明很多模型只是基于浅层的语义匹配来寻找答案而不是真正的理解了文章的含义。

因此对于带有不可回答问题的阅读理解任务，模型要分为两个模块：（1）答案抽取模块；（2）判别不可回答问题模块。Clark 等[8] 尝试在原有的答案抽取模块的基础上额外添加一个专门用来预测不可回答情况的网络层，损失函数定义如下：

$$L_{joint} = -\log(\frac{(1-\delta)e^{z} + \delta e^{\alpha_a \beta_b}}{e^{z} + \sum_{i=1}^{l_p}\sum_{j=1}^{l_p} e^{\alpha_i \beta_j}}) \tag{11}$$

其中 z 表示模型预测该问题是不可回答问题的分数，如果问题是可以回答的则 $\delta = 1$，反之 $\delta = 0$。α 和 β 分别表示输出层预测的文章中每一个单词作为答案起始位置和终止位置的概率，a 和 b 分别代表标准答案在文章中的起始位置和终止位置。

由公式可以看出预测的答案跨度分数 α_a, β_b 和不可回答问题的分数 z 是共同归一化的。Hu 等人[20] 认为两个分数共同归一化会出现冲突，如果模型过分信任预测的答案跨度分数那么就会在预测不可回答问题时产生较低的分数。此外之前的模型并没有验证答案抽取模块预测的答案跨度的合理性。为了解决以上问题，他们提出 Read+Verify 架构。其中 Read 模块就是指答案抽取模块 + 判别不可回答问题模块，Verify 模块用来进一步验证是否答案抽取模块预测的答案跨度所在的句子（原文中称为 answer sentence）就是标准答案所在的句子。为了解决上面提到的冲突问题，在 Read 模块中额外增加了两个辅助损失函数：

$$L_{indep-span} = -\log(\frac{e^{\widetilde{\alpha}_{\widetilde{a}} \widetilde{\beta}_{\widetilde{b}}}}{\sum_{i=1}^{l_p}\sum_{j=1}^{l_p} \widetilde{\alpha}_i \widetilde{\beta}_j}) \tag{12}$$

$$L_{indep-unknown} = -(1-\delta)\log\sigma(z) - \delta\log(1-\delta(z)) \tag{13}$$

其中 $L_{indep-span}$ 代表答案抽取模块的损失函数，而此时的答案抽取模块是独立的预测答案片段而不考虑问题是否可以回答，$\widetilde{\alpha}_{\widetilde{a}}$ 和 $\widetilde{\beta}_{\widetilde{b}}$ 表示的就是答案抽取模块所预测出来的答案跨度。$L_{indep-unknown}$ 代表判断问题不可回答的损失函数，同样它是独立于答案抽取模块的。σ 代表 sigmoid 函数。最后整个 Read 模块的损失函数定义为：

$$L_{Read} = L_{joint} + \gamma L_{indep-span} + \lambda L_{indep-unknown} \tag{14}$$

γ 和 λ 是两个超参数。实验表明去掉 $L_{indep-unknown}$ 后模型在判断不可回答问题上的准确率显著下降，证明了上述提出的冲突确实存在。对于验证模块，他们采用三种结构。第一种将预测出来的答案片段连同问题以及 answer sentence 连接成一个句子送入预训练模型 GPT[39] 中预测不可回答的概率。第二种采用交互式结构，通过注意力机制计算它们之间的关联。第三种结构是前两个结构的结合，将前两个结构的输出张量拼接，实验证明这种混合结构使得模型效果更好。

3.2.2 多段落型阅读理解任务

多段落式阅读理解，即一个问题会对应着多个相关的段落，也可以认为是开放领域（Open-domain）问答的一种形式。Open-domain 问答目的是从广泛的领域资源（如维基百科，网页搜索等）寻找问题的答案而不限于仅仅在某段文本中，这更贴近于真实场景但同时具有相当大的难度。Chen 等人[5] 提出利用检索 + 阅读（Retrieve+Read）的模式处理 open-domain 问答。具体的就是先利用检索模块（Document Retriever）从维基百科中获取 5 个与问题最相关的文章，然后利用阅读器（Document Reader）预测出答案所在的位置。其中 Document retriever 采用基于 TF-IDF 权重的词袋向量模型比较问题和文章的关联程度并且在此基础上用 bigram 哈希优化。

对于 open-domain 问答任务，检索模块要检索出与问题相关的文章，因此检索模块的性能极大地影响着模型整体的效果。如果简单的增加其检索文章的数量就可能导致有不相关的文章被检索出，仍然影响后续阅读模块。为了解决这个问题，Lee 等人[30] 提出段落排序（Paragraph Ranker）机制，利用 BiLSTM 获

得每一篇段落和问题的表示向量，然后计算两个向量的内积作为这篇段落与问题的相似度，目的是从多篇文章中的多个段落选出与问题最相关的几个段落。

目前典型的多段落型数据集如 MS MARCO[34]、TriviaQA[24]，中文的有 DuReader[14]。以 MS MARCO 数据集为例，数据集样例见表 4. MS MARCO 由微软亚洲研究院发布，问题和文章来源于必应搜索，答案由人工生成，因此数据集接近真实应用场景而且答案不在局限于文章中。每个问题对应 10 个由必应搜索引擎返回的文本段落，其中与问题答案相关的段落用 is_select=1 标记为 1。Tan 等人[49] 提出 S-Net 模型，先通过片段抽取模块提取出一段文本作为答案的预测依据，然后利用生成模块生成答案。其中片段抽取模块采用多任务学习策略，除了预测文本片段之外还添加一个段落排名任务，将标记为 is_select=1 的段落视为正例。答案生成模块采用 seq2seq 模型，其中 encoder 端的输入是问题单词的向量表示以及将片段抽取模块的输出作为额外的特征和文章单词的向量表示拼接。实验证明 S-Net 在 MS MSRCO 数据集上的效果要显著地优于 R-Net[55]，ReasoNet[47] 这些用来做片段抽取任务的模型。

多段落型阅读理解任务复杂的原因之一就是由于有多个段落，不同的段落都有可能会包含与问题语义相近的答案，但是有些答案并不是正确的。基于这个问题，Wang 等人[56] 提出一种模型使得来自不同段落的候选答案在基于它们所在的上下文内容里互相验证对方的正确性。将每一篇段落中预测出来的答案与其它段落预测的答案做交互验证。这样做的原因是因为相比于错误的答案，正确答案中的单词往往会在多个段落中重复出现，因此通过交互验证可以凸显出正确答案。最后模型在 MS MARCO 数据集上的效果优于 S-Net。

3.2.3 对话型问答任务

无论是单段落型阅读理解还是多段落型阅读理解任务，它们都属于单轮对话问答，即问答的形式只有一轮，后面的问题与前面的问题和答案无关，每一个问题都是互相独立的。而在现实世界中人们是通过多轮对话形式来交流的，每一轮的问题和答案都会影响后面的问答情况。所以对话型任务来讲，在回答当前轮的问题时不仅需要考虑文章还需要考虑前几轮的问题和答案。具体可以表示为：给定 $Q_i, D, Q_{i-1}, \cdots, Q_{i-k}$ 以及 $A_{i-1}, \cdots, A_{i-k}$ 要求模型给出 A_i。其中 Q_i, A_i 表示第 i 轮的问题和答案，D 表示文章，$Q_{i-1}, \cdots, Q_{i-k}$ 和 $A_{i-1}, \cdots, A_{i-k}$ 分别表示前 k 轮的问题和答案，建模概率：

$$P(A_i|D, Q_i, Q_{i-1}, \cdots, Q_{i-k}, A_{i-1}, \cdots, A_{i-k}) \tag{15}$$

目前典型的对话型问答数据集有 CoQA[42] 以及 QuAC[7]。不同之处在于 CoQA 数据集的答案形式较为简单，类似于 SQuAD[41]，但是包含有 yes/no 以及 unknown 问题，其中 unknown 代表不可回答问题，此外还有一定比例的问题是自由答案形式。而 QuAC 数据集的构造过程中提问者没有看过文章而仅仅了解文章的标题，由回答者根据文章的内容选择出文章的一段文本作为答案，这种数据集构造形式类似于用户在搜索引擎中输入问题查找答案，目的是减少问题和文本之间的依赖，使得模型尽量避免通过浅层的匹配方式获得答案。对话型阅读理解数据集的一个样例见表 10。

其中每一个 Q_i 和 A_i 代表问题和对应的答案，每一个 R_i 表示给出这个答案的依据，用来训练模型。测试集中是没有答案依据的。从图中可以清楚地看到 Q_2 和 Q_3 中的 she 指代的是 Q_1 的答案 A_1，而 Q_4 的 How many? 以及 Q_5 的 Who? 所问的是 Q_3 中的 visitors。显然仅仅靠一轮的问题是无法回答的，对话历史信息在对话型问答任务中尤为重要。

Reddy 等人[42] 采用三种模型在 CoQA 数据集上进行实验。第一种是传统的 seq2seq 模型，decoder 用来生成答案。第二种是指针生成网络（Pointer-Generator Network[45]，PGNet），既可以从词典中生成答案又可以从原文中拷贝单词，很好的解决了 OOV（Out Of Vocabulary）问题。第三种是 DrQA+PGNet 模型，其中 DrQA[5] 是一个片段抽取模块。整个模型的思想是先利用片段提取模块从文章中提取中与问题最

表 10　CoQA[42] 数据集的一个样例

Table 10 An example of CoQA

文章	Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80. Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.
第一轮	Q_1: Who had a birthday? A_1: Jessica R_1: Jessica went to sit in her rocking chair. Today was her birthday and she was turning 80.
第二轮	Q_2: How old would **she** be? A_2: 80 R_2: she was turning 80.
第三轮	Q_3: Did **she** plan to have any **visitors**? A_3: Yes R_3: Her granddaughter Annie was coming over
第四轮	Q_4: **How many?** A_4: Three R_4: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.
第五轮	Q_5: **Who?** A_5: Annie, Melanie and Josh R_5: Her granddaughter Annie was coming over in the afternoon and Jessica was very excited to see her. Her daughter Melanie and Melanie's husband Josh were coming as well.

相关的一段文本，然后利用答案生成模块在这个被抽取出来的文本上生成答案，实验结果表明这种结合模型的效果是优于前两个模型的。为了能够利用历史的对话信息，做法是将前几轮的问题与答案结合到文章当中作为上下文来回答当前轮的问题。

Choi 等人[7] 利用 BiDAF++[8] 模型在 QuAC 数据集上进行实验，为了利用历史的对话信息，在文章中设置一个标记向量用来标记文章中的单词是否出现在历史答案中，在问题向量的基础上添加问题的轮次，这是另一种处理历史对话信息的方式。

Huang 等人[21] 认为上述的方法只是简单的添加之前轮的问题和答案，而忽略了在回答之前轮问题时模型对整篇文章的推理过程状态，他们提出一种带有流机制的模型 FlowQA，目的是将模型处理每一轮的问答过程下的对文章的语义理解状态流向下一轮的问答过程。FlowQA 模型整体上利用双向循环神经网络编码文章，利用单向循环神经网络编码对话历史，对比之前的模型，FlowQA 能够集成更加深层次的对话历史状态。

值得注意的是尽管 CoQA 数据集有部分答案是自由答案形式的，但是上面的模型大多是利用片段提取式的做法在 CoQA 数据集上实验，主要原因在于生成式模型的效果往往不如提取式模型的效果好，因为生成式模型对答案生成模块要求较高。因此如何提高模型的答案生成效果是值得进一步研究的方向。

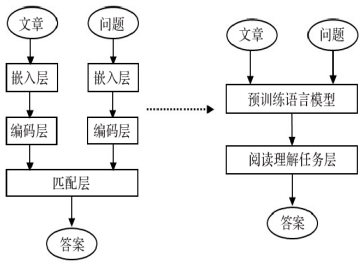


图1 机器阅读理解模型结构对比

3.3 基于预训练模型的 MRC 模型

预训练模型近年来 NLP 领域获得了极大的关注度，基于预训练模型的方法在 NLP 几乎所有的任务上都要优于之前的模型。预训练方式源自于迁移学习的概念: 首先在其它相关任务上预训练模型，使得模型已经学习到一些知识，然后在目标任务上做进一步优化，实现模型所学知识的迁移。对于 NLP 领域来讲，预训练过程就是在大量的文本数据上学习到通用的语言表示。在应用到下游任务时，预训练所学习到的知识提供了一个很好的初始化点，从而加快模型的收敛并且提高模型的性能。此外预训练也对模型起到正则化的作用，使得模型避免在数据不充分的数据集上过拟合。目前最流行的几个预训练模型如 ELMo[38]，GPT[39]，BERT[10] 以及基于 BERT 改进的预训练模型 RoBERTa[32]，UNILM[12]，ALBERT[29] 等全都是基于语言模型做预训练，因此也叫预训练语言模型。下面将主要概述 NLP 领域一些流行的预训练模型以及它们在 MRC 任务上的应用和效果对比。

3.3.1 Transformer

鉴于目前几乎所有的预训练模型都采用 transformer[51] 结构或者其变体作为模型的特征提取器,因此本节首先介绍 transformer 结构。Transformer 是由 Vaswani 等人提出了一种用于机器翻译的 encoder-decoder 结构。Encoder 端由六个相同的层堆叠而成，每一层有两个子层，第一个子层采用多头（multi-head）自注意力机制，第二个子层采用前馈神经网络（Feed-Forward Network，FFN）构成。之所以用自注意力机制是因为它既可以捕获句子中每一个单词的全局依赖关系而不受距离影响又可以并行计算。对比公式（1），自注意力机制下 $Q = K = V$，transformer 中采用的计算方式如下:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{16}$$

其中 $\sqrt{d_k}$ 代表张量维度。此外 transformer 采用的是多头（multi-head）自注意力机制，将 Q, K, V 三个张量线性映射成多份，每一份之间做注意力的运算最后拼接。

$$\begin{aligned} \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ \text{Multi-head}(Q, K, V) &= \text{concat}(\text{head}_1, \text{head}_2, \cdots, \text{head}_h)W^o \end{aligned} \tag{17}$$

其中 h 代表头的数目，是一个超参数，W_i^Q, W_i^K, W_i^V, W^o 都是训练参数。

采用 multi-head 的目的是让模型联合关注序列中不同位置单词的不同表示子空间的信息，可以类比于卷积神经网络中利用多个卷积核做特征提取，目的同样是使得不同的卷积核关注的不同的特征。此外每一个子层都利用层正则化（layer normalization[1]）和残差连接（residual connection[13]）机制。Decoder 端与 Encoder 端类似，区别在于每一层额外添加了 encoder-decoder 注意力。

Transformer 的 encoder 端和 decoder 端都可以做特征提取器如 BERT[10] 用 encoder 端特征提取，GPT[39] 用 decoder 端特征提取，实验证明在大规模数据集上 transformer 的特征提取能力要强于基于 RNN 变体的编码器，目前几乎所有的 NLP 预训练模型都是利用 transformer 作为特征提取器。

3.3.2 预训练模型

ELMo[38] 是在 2018 年提出的一种预训练语言模型。传统的词嵌入模型如 Word2Vec[33],GloVe[37] 属于静态的词向量，训练好模型后一个单词的表示向量就是固定的，没有考虑上下文的信息，因此无法解决多义词问题。ELMo 提出一个三层网络的模型，第一层就是词嵌入层用来提取单词特征，随后是两层 BiLSTM 网络分别提取单词的词性特征和语义特征。前向 LSTM 的目标是根据前 $k-1$ 个词预测第 k 个词，从而计算出一个句子的概率，如公式（10）。反向 LSTM 的目标是根据最后的单词直到第 $k+1$ 个单词预测第 k 个单词，具体如公式（11）。

$$p(t_1, t_2, \cdots, t_N) = \prod_{k=1}^{N} p(t_k | t_1, t_2, \cdots, t_{k-1}) \tag{18}$$

$$p(t_1, t_2, \cdots, t_N) = \prod_{k=1}^{N} p(t_k | t_{k+1}, t_{k+2}, \cdots, t_N) \tag{19}$$

最后的目标函数是最大化联合的前向和后向最大似然：

$$\begin{aligned} L(\Theta) = \sum_{k=1}^{N} (&\log p(t_1, \cdots, t_{k-1}; \Theta)) \\ &+ \log p(t_{k+1}, \cdots, t_N; \Theta) \end{aligned} \tag{20}$$

在做阅读理解任务时，将文章和问题输入到模型中，每一层都会得到句子的语义表示，然后将每一层的特征加权求和作为 ELMo 的输出，此时得到的每一个单词的向量表示都是考虑了上下文的，将其作为下游模型嵌入层的输入。利用 ELMo+BiDAF[46] 结构超过之前单模型 8.3 个百分点。可以看出 ELMo 属于自回归语言模型，模型的迁移方式是基于特征的方式。

OpenAI 提出一种生成式预训练模型 GPT[39]，使用多层的 transformer[51] 的解码端作为特征提取器。模型采用两阶段的训练方式，先利用大规模无监督语料库训练语言模型，然后在下游任务的少量监督数据集上微调模型，因此 GPT 也是一种半监督学习方式。采用单向语言模型作为训练的任务，预训练阶段的目标函数就是标准的单向语言模型的目标函数：

$$L(\Theta) = \sum_{k=1}^{N} (\log p(t_1, \cdots, t_{k-1}; \Theta)) \tag{21}$$

GPT 与 ELMo 的共同点是它们都属于自回归语言模型，而最大的不同之处在于迁移的设计上并不是像 ELMo 那种基于特征方式的迁移。GPT 有统一的输入数据表示形式，而且在输入层并不需要继续设计复杂的网络模型而仅仅只需要简单的结构，在训练时整个网络模型的参数共同训练。这种基于微调的迁移方式被后续的其它预训练模型广泛采用。利用 GPT 在 RACE[28] 数据集上微调后达到的效果比之前最好的模型提高了 5.7 个百分点。GPT 是 NLP 领域首先提出的一种基于微调（fine-tune）的通用式网络结构，不仅仅在 MRC 领域，在很多其它的 NLP 领域都取得了很大的进步。

BERT[10] 与 ELMo 和 GPT 最本质的不同在于预训练方式上采用的自编码语言模型。自回归语言模型的缺点就是由于自回归的性质使得它不能同时利用一个单词的上下文信息预测这个单词，ELMo 虽然利用双向 LSTM 来预测单词但是这也只是两个单向的语言模型的拼接并不能当做双向语言模型。BERT 在整

个预训练的流程上采用 GPT 的方式，即预训练然后微调。但是 BERT 采用降噪自编码（DAE）的方式训练，具体的就是在输入数据中加入噪声，也就是随机掩盖掉一些单词，让模型根据掩盖掉单词的上下文预测这个单词，这种训练方式也叫掩码语言模型（MLM)。对比 ELMo 和 GPT 的目标函数，BERT 的掩码语言模型的目标函数为：

$$L(\Theta) = \sum_{i=1}^{N} \log P(t_k|t_1, t_2, \cdots, t_{k-1}, t_{k+1}, \cdots, t_N) \tag{22}$$

此外利用下一个句子预测（NSP）任务使得模型在诸如文本蕴含、问答这类需要判断两个句子关系的下游任务表现更好。BERT 的预训练过程实质上是一个多任务学习的过程，通过 MLM 和 NSP 两个任务提高了预训练模型的语义表达能力，其中 MLM 任务用来学习句子中词与词之间的语义关联而 NSP 任务用来学习两个句子之间的逻辑关系。BERT 在 SQuAD 1.1[41] 数据集上的效果超过了人类的水平，在其它的 NLP 任务上也都有提升。

BERT 这种降噪自编码的方式虽然可以达到双向的利用上下文信息，但是由于其在预训练过程中对输入数据加入掩码而在微调时又不会加入掩码，导致了预训练过程与微调过程不匹配，存在一定的数据分布偏差，此外 BERT 对于屏蔽词的预测是独立的。基于上述问题，文献 [62] 提出一种新的预训练模型 XLNet，它是一种可以获得双向的上下文信息的自回归语言模型，克服了传统的自回归语言模型和自编码语言模型各自的问题。XLNet 采用的是排列语言模型（Permutation Language Model，PLM），自回归模型中利用文本的前向或者后向序列的最大似然来建模，而 PLM 排列这个文本所有可能的序列顺序，仍然利用自回归语言模型的目标函数，但是综合所有可能的排列后每一个单词都可以获得双向的上下文信息。此外 XLNet 借鉴 transformer-XL[9] 中的片段循环机制引入循环机制可以捕获更长的句子依赖关系。

文献 [32] 提出一种基于 BERT 改进其训练方式的预训练模型 RoBERTa，其改进的方式包括：使用动态掩码替换静态掩码、去除 NSP 任务、使用更大的 batch、更多的训练语料以及更长的训练时间。其中动态掩码是指对于输入数据中随机掩盖的单词并不是固定的，因为 BERT 的掩码机制是静态的，即对于每一个输入序列一旦选定了其中的的某个单词将其屏蔽，那么之后的整个训练过程该单词始终被掩盖。RoBERTa 提出将输入数据复制 10 份，每一份都是随机掩盖部分单词，这样同一个输入序列就会有 10 种不同的掩码方式，从而达到动态掩码的目的。

UNILM[12] 扩展了 BERT 预训练的任务，由于双向语言模型的性质使得 BERT 在生成任务上效果不好，UNILM 同时训练单向语言模型（包括从左到右和从右到左）、双向语言模型以及 Seq2Seq 语言模型，使用掩码机制来解决不同的语言模型约束问题。虽然是三个不同的语言模型作为训练任务但是共享同一个网络结构，也就是利用三个任务来联合的优化模型的参数。这种多任务学习的方式缓解了模型在某一个单一任务上容易出现过拟合的问题，使得预训练后的模型不仅在原有的自然语言理解任务上效果进一步提升，同时在自然语言生成任务上也达到了很好的效果。在微调阶段，对于自然语言理解任务（如文本分类、抽取式问答等）同 BERT 的微调方式一样，对于自然语言生成任务（如自动化摘要、生成式问答等）采用与预训练阶段 Seq2Seq 语言模型类似的方式在目标序列中随机的掩盖掉一些单词从而让模型根据源序列生成目标序列的单词达到生成任务的目的。UNILM 不仅在抽取式 QA 数据集（如 SQuAD）上超过 BERT，在生成式 QA 数据集（如 CoQA）上的表现远超过最初的基准模型。

ALBERT[29] 改进了 BERT 的 NSP 任务，BERT 的 NSP 任务包含了两个子任务，来源于同一篇文章的两个连续的句子判别为正例，来源于不同文章的句子判别为负例，这使得模型不能集中于判断两个句子之间的顺序反而更加关注句子所表达的主题是否一致。因此 ALBERT 中用 SOP(sentence-order prediction) 取代 NSP 任务，SOP 是指句子顺序预测，两个句子都是来源于同一篇文章的连续的句子，调换顺序后便是负例，这使得模型集中于预测句子之间的顺序关系，实验表明 SOP 任务使得模型的效果提升一个百分点。ALBERT 的改进机制使得模型的预训练后的效果更好，在 RACE[28]，SQuAD 2.0[40] 等机器阅读理解

数据集上的准确率超过 BERT 以及其它的预训练模型。表 11 从预训练任务、模型采用的特征提取器等详细对比了本文介绍的所有预训练模型。表 12 对比了几个预训练模型在两个常用的 MRC 数据集上的表现。

表 11　预训练模型对比

Table 11 Comparison of pre-trained model

模型	任务	模型结构	介绍
ELMo[38]	前向 LM 反向 LM	LSTM	拼接两个单向语言模型的语义信息， 基于特征形式迁移
GPT[39]	前向 LM	Transformer	首次采用预训练＋微调形式
BERT[10]	MLM NSP	Transformer	利用掩码语言模型（MLM）和下一个句子预测（NSP）共同作为训练任务
XLNet[62]	PLM	Transformer-XL	采用排列语言模型（PLM）， 使得模型以自回归方式训练但是基于上下文预测
RoBERTa[32]	MLM	Transformer	采用动态掩码机制，去除 NSP 任务
UNILM[12]	前向 LM，反向 LM MLM Seq2SeqLM	Transformer	同时训练多种语言模型， 采用掩码机制解决不同语言模型的约束问题
ALBERT[29]	双向 LM+SOP	Transformer	对比 BERT 采用矩阵分解和共享参数减少模型的参数量， 同时用句子顺序预测任务（SOP）取代下一个句子预测任务（NSP）

表 12　预训练模型对比

模型	SQuAD 2.0[40]	RACE[28]
	EM/F1	Acc
GPT_{v1}[39]	-	59.0
$BERT_{large}$[10]	80.0/83.1	72.0
XLNet[62]	86.4/89.1	81.8
RoBERTa[32]	86.8/89.8	83.2
ALBERT[29]	88.1/90.9	86.5

3.3.3　更强大的 MRC 模型

在预训练模型出现后不仅仅在 MRC 任务上，在其它的 NLP 任务上模型的结构都发生了较大的变化，在模型的基础上利用具体任务的数据微调模型即可达到很好的效果。但是预训练模型毕竟是一个通用式的模型，它给了模型很好的初始化参数，不过要想达到更好的效果仍然要根据具体任务的形式来设计模型。下面列举几个 MRC 数据集上目前效果较好的基于预训练模型的 MRC 模型。

人在做阅读理解问题的时候通常会先带着问题大致的浏览一下这篇文章，对这篇文章的含义有一个大致的了解。之后再根据问题详细的阅读文章寻找答案。受到这种阅读形式的启发，Zhang 等人[66] 提出一种回顾式阅读器（Retrospective Reader，Retro-Reader）模型。整个模型由两个步骤构成：（1）第一步先简要的略读文章，建模文章与问题的大致关联给出初步的判断该问题是否可以回答。（2）第二步是精读模块，

目的是验证可回答性并且给出最终判断。模型的编码器采用强大的预训练模型 ALBERT。Retro-Reader 在 SQuAD 2.0[40] 数据集上显著优于其它模型。

Zhang 等人[65] 利用 BERT[10] 和 XLNet[62] 作为编码器同时采用文献 [54] 提出的 co-matching 方法提出了 DCMN 模型，在 RACE[28] 数据集上达到了很高的准确率。而后在 DCMN 的基础上引入选项交互模块和段落选择模块提出 DCMN+[64] 模型，模型的性能进一步提升。

BERT 以及基于 BERT 改进的预训练模型其数据输入形式只能是两个句子的拼接因此并不适合直接处理 CoQA 这种对话型阅读理解数据集。Zhu 等人[67] 提出 SDNet 模型，以基于特征的方式迁移 BERT 作为编码器，同时将之前轮次的问题和答案拼接到当前轮的问题上构成一个新问题。模型采用自注意力机制获得历史对话信息之间的交互语义，具体的计算方式采用 FusionNet[22] 模型提出的融合方法。Ohsugi 等人[35] 以基于微调的方式迁移 BERT 模型。将历史对话信息每一轮的问题与答案分别与文章连接送入 BERT，将每一个 BERT 的输出连接作为输出层的输入。两个模型在 CoQA[42] 和 QuAC[7] 两个对话型数据集上的效果均超过前面的 BiDAF++[8]，FlowQA[21] 等利用复杂交互机制的模型。

虽然设计一个强大的预训练模型具有更好的泛化型和应用价值，但是预训练模型所消耗的计算资源是巨大的，因此如何利用预训练模型结合具体任务改进模型的结构是至关重要的。

4 讨论

本章主要讨论 MRC 领域的发展历史和目前 MRC 领域存在的主要问题

4.1 回顾 MRC 的发展历史

从 MRC 数据集的发展角度看，从最简单的填空型数据集，到抽取型数据集再到复杂的需要从多段落中归纳答案的数据集等每一个新的数据集都会在原有数据集的基础上增加各种各样的难度，从而不得不设计更加优秀的模型处理这些任务。从模型内部各个层次的发展角度上看，嵌入层和编码层在预训练模型出现之前并没有较大的变动。各个模型主要集中在交互层的注意力机制上，为了获得更全面的交互信息从单向发展到双向，为了可以达到多步推理的目的从单跳结构发展到多跳结构。而对于更加复杂的阅读理解任务时，模型整体发生了较大的改动。面对无答案的阅读理解任务设计答案验证模块来验证问题是否可以回答，面对多段落型阅读理解任务设计段落选择模块减小答案搜索范围，面对对话形式的阅读理解任务通过将之前轮的对话信息以文本拼接或者信息流动的方式使得模型在回答当前轮问题的时候可以关注到之前的对话信息。

4.2 MRC 面临的主要问题

预训练模型出现后基于神经网络的 MRC 模型的性能再一次都取得了显著的进步，但是目前 MRC 领域还有一些问题需要解决。

如何设计高质量的数据集 前面介绍了很多 MRC 领域常用的数据集，也同时分析了各个数据集的难度、特点、规模等。虽然目前的数据集已经不像之前的数据集模型仅仅通过浅层的匹配机制就能够达到很高的效果，但是一个高质量的机器阅读理解数据集不应该是普通的大规模、高难度的特点，它要能够准确的考察模型是否具有人类阅读能力，包括推理能力，常识能力等。

模型缺乏推理能力 如前面所述，基于注意力机制的匹配模型大多是浅层的语义匹配模型，基于多跳结构的推理模式还过于单一，这些均没有形成深层次的阅读理解模型。Jia 等人[23] 在 SQuAD 数据集的基础上设计对抗样本来测试 MRC 模型，在文章中加入人工设计的句子，这些句子与问题相似但是与答案无

关以此用来误导模型。这些误导的句子对人来说没有什么影响然而实验证明几乎所有模型的准确率都显著下降，这表明模型靠的是关键词匹配方式寻找答案而真正的阅读理解能力和推理能力很弱，让模型具有较强的推理能力至关重要。

答案生成技术不足 生成答案的技术还需要进一步提升，回顾目前机器阅读理解领域的数据集以及相应的模型，大多集中于片段抽取式问答且模型准确度很高，而对于自由答案型这种需要生成答案的模型效果很差，有些模型直接将生成式问题转为抽取式问题效果反而优于生成式的做法，主要原因在于生成答案模块的效果不够好。

5 总结与展望

本文从机器阅读理解任务的定义出发，第二章概述了机器阅读理解任务以及介绍了不同任务下的数据集和相应的评估标准。第三章神经机器阅读理解模型进行了分析与研究，主要涉及经典模型的整体框架，其中详细分析了各个模型在交互层注意力机制的设计。此外还介绍了复杂任务下 MRC 模型的设计，同时也总结了目前一些主流的预训练模型，分析了它们之间的差异，列举了一些在预训练模型的基础上改进的 MRC 模型。通过各个模型的实验对比结果可以看到基于预训练的模型性能要显著的优于传统的仅仅基于注意力机制的模型。第四章回顾了 MRC 领域的发展并且指出了目前 MRC 领域存在的问题。

机器阅读理解赋予了计算机阅读理解文本的能力，在搜索、对话、医疗以及教育领域都有着广阔的应用空间，未来的研究方向有以下几点值得关注：

1）目前机器阅读理解领域的数据集大多是通用领域方向，而设计专业领域数据集也尤为重要。比如通过利用机器阅读理解技术分析产品说明文档和用户的问题语义从而解决用户对产品的问题达到智能客服服务，或者在医疗诊断中分析大量病例和知识库提供智能医疗服务。

2）目前机器阅读理解主要集中于非结构化的文本领域，而还有许多其它结构、不同模态的数据如表格、视频、音频、图片等，相关的研究方向如数据库问答，视觉问答等。多模态阅读理解模型是未来的机器阅读理解发展方向之一。

3）目前很少有机器阅读理解模型融合外部知识，都是直接根据给定的文档回答相关的问题，而人在阅读一篇文章的时候对这篇文章的理解程度和他已经掌握的知识水平有很大关系。因此如果将外部知识源融入模型中那么模型的性能大概率会显著提高。

参考文献

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: http://arxiv.org/abs/1409.0473.

[3] Danqi Chen. “Neural reading comprehension and beyond”. PhD thesis. Stanford University, 2018.

[4] Danqi Chen, Jason Bolton, and Christopher D. Manning. “A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task”. In: *CoRR* abs/1606.02858 (2016). arXiv: 1606.02858. URL: http://arxiv.org/abs/1606.02858.

[5] Danqi Chen et al. "Reading Wikipedia to Answer Open-Domain Questions". In: *CoRR* abs/1704.00051 (2017). arXiv: 1704.00051. URL: http://arxiv.org/abs/1704.00051.

[6] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, 2014, pp. 1724–1734. DOI: 10.3115/v1/d14-1179. URL: https://doi.org/10.3115/v1/d14-1179.

[7] Eunsol Choi et al. "QuAC: Question Answering in Context". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff et al. Association for Computational Linguistics, 2018, pp. 2174–2184. DOI: 10.18653/v1/d18-1241. URL: https://doi.org/10.18653/v1/d18-1241.

[8] Christopher Clark and Matt Gardner. "Simple and Effective Multi-Paragraph Reading Comprehension". In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*. Ed. by Iryna Gurevych and Yusuke Miyao. Association for Computational Linguistics, 2018, pp. 845–855. DOI: 10.18653/v1/P18-1078. URL: https://www.aclweb.org/anthology/P18-1078/.

[9] Zihang Dai et al. "Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context". In: *CoRR* abs/1901.02860 (2019). arXiv: 1901.02860. URL: http://arxiv.org/abs/1901.02860.

[10] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

[11] Bhuwan Dhingra et al. "Gated-Attention Readers for Text Comprehension". In: *CoRR* abs/1606.01549 (2016). arXiv: 1606.01549. URL: http://arxiv.org/abs/1606.01549.

[12] Li Dong et al. "Unified Language Model Pre-training for Natural Language Understanding and Generation". In: *CoRR* abs/1905.03197 (2019). arXiv: 1905.03197. URL: http://arxiv.org/abs/1905.03197.

[13] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[14] Wei He et al. "DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications". In: *CoRR* abs/1711.05073 (2017). arXiv: 1711.05073. URL: http://arxiv.org/abs/1711.05073.

[15] Karl Moritz Hermann et al. "Teaching Machines to Read and Comprehend". In: *CoRR* abs/1506.03340 (2015). arXiv: 1506.03340. URL: http://arxiv.org/abs/1506.03340.

[16] Karl Moritz Hermann et al. "Teaching Machines to Read and Comprehend". In: *CoRR* abs/1506.03340 (2015). arXiv: 1506.03340. URL: http://arxiv.org/abs/1506.03340.

[17] Felix Hill et al. "The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations". In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Yoshua Bengio and Yann LeCun. 2016. URL: http://arxiv.org/abs/1511.02301.

[18] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[19] Minghao Hu, Yuxing Peng, and Xipeng Qiu. "Mnemonic Reader for Machine Comprehension". In: *CoRR* abs/1705.02798 (2017). arXiv: 1705.02798. URL: http://arxiv.org/abs/1705.02798.

[20] Minghao Hu et al. "Read + Verify: Machine Reading Comprehension with Unanswerable Questions". In: *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 6529–6537. DOI: 10.1609/aaai.v33i01.33016529. URL: https://doi.org/10.1609/aaai.v33i01.33016529.

[21] Hsin-Yuan Huang, Eunsol Choi, and Wen-tau Yih. "FlowQA: Grasping Flow in History for Conversational Machine Comprehension". In: *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL: https://openreview.net/forum?id=ByftGnR9KX.

[22] Hsin-Yuan Huang et al. "FusionNet: Fusing via Fully-aware Attention with Application to Machine Comprehension". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL: https://openreview.net/forum?id=BJIgi%5C_eCZ.

[23] Robin Jia and Percy Liang. "Adversarial Examples for Evaluating Reading Comprehension Systems". In: *CoRR* abs/1707.07328 (2017). arXiv: 1707.07328. URL: http://arxiv.org/abs/1707.07328.

[24] Mandar Joshi et al. "TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. Ed. by Regina Barzilay and Min-Yen Kan. Association for Computational Linguistics, 2017, pp. 1601–1611. DOI: 10.18653/v1/P17-1147. URL: https://doi.org/10.18653/v1/P17-1147.

[25] Rudolf Kadlec et al. "Text Understanding with the Attention Sum Reader Network". In: *CoRR* abs/1603.01547 (2016). arXiv: 1603.01547. URL: http://arxiv.org/abs/1603.01547.

[26] Lukasz Kaiser, Aidan N. Gomez, and François Chollet. "Depthwise Separable Convolutions for Neural Machine Translation". In: *CoRR* abs/1706.03059 (2017). arXiv: 1706.03059. URL: http://arxiv.org/abs/1706.03059.

[27] Tomás Kociský et al. "The NarrativeQA Reading Comprehension Challenge". In: *Trans. Assoc. Comput. Linguistics* 6 (2018), pp. 317–328. URL: https://transacl.org/ojs/index.php/tacl/article/view/1197.

[28] Guokun Lai et al. “RACE: Large-scale ReAding Comprehension Dataset From Examinations”. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*. Ed. by Martha Palmer, Rebecca Hwa, and Sebastian Riedel. Association for Computational Linguistics, 2017, pp. 785–794. DOI: 10.18653/v1/d17-1082. URL: https://doi.org/10.18653/v1/d17-1082.

[29] Zhenzhong Lan et al. “Albert: A lite bert for self-supervised learning of language representations”. In: *arXiv preprint arXiv:1909.11942* (2019).

[30] Jinhyuk Lee et al. “Ranking Paragraphs for Improving Answer Recall in Open-Domain Question Answering”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 565–569. DOI: 10.18653/v1/D18-1053. URL: https://www.aclweb.org/anthology/D18-1053.

[31] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://www.aclweb.org/anthology/W04-1013.

[32] Yinhan Liu et al. “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *CoRR* abs/1907.11692 (2019). arXiv: 1907.11692. URL: http://arxiv.org/abs/1907.11692.

[33] Tomas Mikolov et al. “Distributed Representations of Words and Phrases and their Compositionality”. In: *CoRR* abs/1310.4546 (2013). arXiv: 1310.4546. URL: http://arxiv.org/abs/1310.4546.

[34] Tri Nguyen et al. “MS MARCO: A Human Generated MAchine Reading COmprehension Dataset”. In: *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*. Ed. by Tarek Richard Besold et al. Vol. 1773. CEUR Workshop Proceedings. CEUR-WS.org, 2016. URL: http://ceur-ws.org/Vol-1773/CoCoNIPS%5C_2016%5C_paper9.pdf.

[35] Yasuhito Ohsugi et al. “A Simple but Effective Method to Incorporate Multi-turn Context with BERT for Conversational Machine Comprehension”. In: *CoRR* abs/1905.12848 (2019). arXiv: 1905.12848. URL: http://arxiv.org/abs/1905.12848.

[36] Kishore Papineni et al. “Bleu: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. ACL, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://www.aclweb.org/anthology/P02-1040/.

[37] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. “Glove: Global Vectors for Word Representation”. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. Ed. by Alessandro Moschitti, Bo Pang, and Walter Daelemans. ACL, 2014, pp. 1532–1543. DOI: 10.3115/v1/d14-1162. URL: https://doi.org/10.3115/v1/d14-1162.

[38] Matthew E. Peters et al. “Deep contextualized word representations”. In: *CoRR* abs/1802.05365 (2018). arXiv: 1802.05365. URL: http://arxiv.org/abs/1802.05365.

[39] Alec Radford et al. “Improving language understanding by generative pre-training (2018)”. In: *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_understanding_paper. pdf* (2018).

[40] Pranav Rajpurkar, Robin Jia, and Percy Liang. “Know What You Don’t Know: Unanswerable Questions for SQuAD”. In: *CoRR* abs/1806.03822 (2018). arXiv: 1806.03822. URL: http://arxiv.org/abs/1806.03822.

[41] Pranav Rajpurkar et al. “SQuAD: 100, 000+ Questions for Machine Comprehension of Text”. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*. Ed. by Jian Su, Xavier Carreras, and Kevin Duh. The Association for Computational Linguistics, 2016, pp. 2383–2392. DOI: 10.18653/v1/d16-1264. URL: https://doi.org/10.18653/v1/d16-1264.

[42] Siva Reddy, Danqi Chen, and Christopher D. Manning. “CoQA: A Conversational Question Answering Challenge”. In: *Trans. Assoc. Comput. Linguistics* 7 (2019), pp. 249–266. URL: https://transacl.org/ojs/index.php/tacl/article/view/1572.

[43] Matthew Richardson, Christopher J. C. Burges, and Erin Renshaw. “MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text”. In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. ACL, 2013, pp. 193–203. URL: https://www.aclweb.org/anthology/D13-1020/.

[44] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. “Learning representations by back-propagating errors”. In: *nature* 323.6088 (1986), pp. 533–536.

[45] Abigail See, Peter J. Liu, and Christopher D. Manning. “Get To The Point: Summarization with Pointer-Generator Networks”. In: *CoRR* abs/1704.04368 (2017). arXiv: 1704.04368. URL: http://arxiv.org/abs/1704.04368.

[46] Min Joon Seo et al. “Bidirectional Attention Flow for Machine Comprehension”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL: https://openreview.net/forum?id=HJ0UKP9ge.

[47] Yelong Shen et al. “ReasoNet: Learning to Stop Reading in Machine Comprehension”. In: *CoRR* abs/1609.05284 (2016). arXiv: 1609.05284. URL: http://arxiv.org/abs/1609.05284.

[48] Alessandro Sordoni, Philip Bachman, and Yoshua Bengio. “Iterative Alternating Neural Attention for Machine Reading”. In: *CoRR* abs/1606.02245 (2016). arXiv: 1606.02245. URL: http://arxiv.org/abs/1606.02245.

[49] Chuanqi Tan et al. “S-Net: From Answer Extraction to Answer Generation for Machine Reading Comprehension”. In: *CoRR* abs/1706.04815 (2017). arXiv: 1706.04815. URL: http://arxiv.org/abs/1706.04815.

[50] Adam Trischler et al. “NewsQA: A Machine Comprehension Dataset”. In: *Proceedings of the 2nd Workshop on Representation Learning for NLP, Rep4NLP@ACL 2017, Vancouver, Canada, August 3, 2017*. Ed. by Phil Blunsom et al. Association for Computational Linguistics, 2017, pp. 191–200. DOI: 10.18653/v1/w17-2623. URL: https://doi.org/10.18653/v1/w17-2623.

[51] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.

[52] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. “Pointer Networks”. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. Ed. by Corinna Cortes et al. 2015, pp. 2692–2700. URL: http://papers.nips.cc/paper/5866-pointer-networks.

[53] Shuohang Wang and Jing Jiang. “Machine Comprehension Using Match-LSTM and Answer Pointer”. In: *CoRR* abs/1608.07905 (2016). arXiv: 1608.07905. URL: http://arxiv.org/abs/1608.07905.

[54] Shuohang Wang et al. “A Co-Matching Model for Multi-choice Reading Comprehension”. In: *CoRR* abs/1806.04068 (2018). arXiv: 1806.04068. URL: http://arxiv.org/abs/1806.04068.

[55] Wenhui Wang et al. “Gated Self-Matching Networks for Reading Comprehension and Question Answering”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 189–198. DOI: 10.18653/v1/P17-1018. URL: https://www.aclweb.org/anthology/P17-1018.

[56] Yizhong Wang et al. “Multi-Passage Machine Reading Comprehension with Cross-Passage Answer Verification”. In: *CoRR* abs/1805.02220 (2018). arXiv: 1805.02220. URL: http://arxiv.org/abs/1805.02220.

[57] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. “Making Neural QA as Simple as Possible but not Simpler”. In: *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), Vancouver, Canada, August 3-4, 2017*. Ed. by Roger Levy and Lucia Specia. Association for Computational Linguistics, 2017, pp. 271–280. DOI: 10.18653/v1/K17-1028. URL: https://doi.org/10.18653/v1/K17-1028.

[58] Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. “Constructing Datasets for Multi-hop Reading Comprehension Across Documents”. In: *CoRR* abs/1710.06481 (2017). arXiv: 1710.06481. URL: http://arxiv.org/abs/1710.06481.

[59] Qizhe Xie et al. “Large-scale Cloze Test Dataset Designed by Teachers”. In: *CoRR* abs/1711.03225 (2017). arXiv: 1711.03225. URL: http://arxiv.org/abs/1711.03225.

[60] Caiming Xiong, Victor Zhong, and Richard Socher. “Dynamic Coattention Networks For Question Answering”. In: *CoRR* abs/1611.01604 (2016). arXiv: 1611.01604. URL: http://arxiv.org/abs/1611.01604.

[61] Zhilin Yang et al. “HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*. Ed. by Ellen Riloff et al. Association for Computational Linguistics, 2018, pp. 2369–2380. DOI: 10.18653/v1/d18-1259. URL: https://doi.org/10.18653/v1/d18-1259.

[62] Zhilin Yang et al. “XLNet: Generalized Autoregressive Pretraining for Language Understanding”. In: *CoRR* abs/1906.08237 (2019). arXiv: 1906.08237. URL: http://arxiv.org/abs/1906.08237.

[63] Adams Wei Yu et al. “QANet: Combining Local Convolution with Global Self-Attention for Reading Comprehension”. In: *CoRR* abs/1804.09541 (2018). arXiv: 1804.09541. URL: http://arxiv.org/abs/1804.09541.

[64] Shuailiang Zhang et al. “DCMN+: Dual Co-Matching Network for Multi-choice Reading Comprehension”. In: *CoRR* abs/1908.11511 (2019). arXiv: 1908.11511. URL: http://arxiv.org/abs/1908.11511.

[65] Shuailiang Zhang et al. “Dual Co-Matching Network for Multi-choice Reading Comprehension”. In: *CoRR* abs/1901.09381 (2019). arXiv: 1901.09381. URL: http://arxiv.org/abs/1901.09381.

[66] Zhuosheng Zhang, Junjie Yang, and Hai Zhao. “Retrospective Reader for Machine Reading Comprehension”. In: *CoRR* abs/2001.09694 (2020). arXiv: 2001.09694. URL: https://arxiv.org/abs/2001.09694.

[67] Chenguang Zhu, Michael Zeng, and Xuedong Huang. “SDNet: Contextualized Attention-based Deep Network for Conversational Question Answering”. In: *CoRR* abs/1812.03593 (2018). arXiv: 1812.03593. URL: http://arxiv.org/abs/1812.03593.