神经机器阅读理解研究综述

摘　要: 摘要最后写
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

自然语言处理是人工智能的重要分支，是实现人工智能的核心技术，主要研究如何处理、分析以及应用自然语言。教会机器阅读文本并且理解人类语言是自然语言处理领域的重要任务，机器阅读理解（Machine Reading Comprehension，MRC）的目标就是利用自然语言处理技术使得计算机能够像人一样阅读并且理解文章，它有着很多应用场景，如搜索引擎中的智能问答，电商领域的智能客服以及对话系统等。

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，如 Lehnert[1] 构建的 QUALM 系统。但是系统非常小并且局限于手工编码，很难推广到更大的领域。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配，准确率可以达到 40% 左右，但是由于依靠手工规则，模型泛化能力很差。即便是利用传统的机器学习方法，模型也只是在句子级别粒度上回答问题，而且只能从文本中提取浅层特征，因此早期的 MRC 系统难以获得期望的性能也不能实际的应用，MRC 领域发展较为缓慢。

随着深度学习的兴起以及 NLP 领域技术的发展，为了弥补传统 MRC 技术的缺陷，2015 年 DeepMind 研究员 Hermann 等人[2] 使用神经网络模型解决 MRC 任务，同时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。他们提出两个基于神经网络和注意力机制构建的模型：Attentive Reader 和 Impatient Reader，模型在 CNN&Daily 数据集上效果远超过传统方法的效果。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基于神经网络的机器阅读理解模型，也简称为神经机器阅读理解模型。

本文的目的是对从 2015 年来机器阅读理解（MRC）领域的研究任务、相关数据集以及模型做综述。整体安排如下：第二章概述 MRC 任务并且简介各个任务相关的数据集；第三章介绍基于神经网络的 MRC 模型，包括经典的 MRC 模型以及基于预训练模型的 MRC 模型；第四章分析 MRC 任务目前的主要挑战；第五章讨论 MRC 的应用、面临的主要问题以及未来的发展趋势；第六章对全文作总结。

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，建模给定 D 和 Q 的条件下预测 A 的概率：$P(A|D,Q)$。

MRC 任务已经从早期的需要阅读一篇文章，答案是单个单词的阅读理解任务，发展到需要阅读多篇文章，并且答案需要从多篇文章中推理出来的阅读理解任务，使得阅读理解任务更加的贴近真实场景。本文按照 Chen[12] 提出的分类方式，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 $a \in D$。相关数据集如下：

CNN&Daily Mail: 典型的填空式数据集是由 Google DeepMind 和牛津大学发布于 2015 年的 CNN&Daily Mail[13] 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。

CBT: Hill 等人[14] 发布的（Children's Book Test，CBT）数据集，语料库来源于儿童读物（Gutenberg[4] 工程），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: CLOTH[15] 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。每一个问题是由专家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。

2.2 多项选择式

多项选择式这类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 A 中选择正确的答案，建模概率：$P(A_i|D,Q)$，其中 $A_i \in A$。相关数据集如下：

MCTest: MCTest[16] 是一个早期提出来的多项选择式数据集，问题形式为四选一。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: RACE[9] 数据集是从中国中学生英语考试题中的阅读理解题型建立的数据集，共有将近 2.8 万篇文章以及 10 万个问题，这些问题和候选答案都是由专家编写。此外 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。

[1] 本文中文章（Document）和段落（Passage）是同样的概念

[2] www.cnn.com

[3] www.dailymail.co.uk

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

2.3 抽取式

抽取式阅读理解任务可以看做是填空式任务的扩展，不像填空式任务仅仅要求答案是原文中的某个单词，抽取式任务要求模型从原文中抽取一段连续的文本作为答案，且答案的长度不固定。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。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，k 代表答案的长度。相关数据集如下：

SQuAD：SQuAD[3] 是抽取式阅读理解的代表性数据集，也是 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 含有 536 篇文章，总计十万多个问题-答案对。

NewsQA：NewsQA[17] 与 SQuAD 数据集的构造方式类似，区别在于 NewsQA 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[18]。

TriviaQA：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。基于这一思想，Joshi 等人[19] 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。最后构建出包含 65 万多个（问题-答案-文章）三元组的抽取式阅读理解数据集 TriviaQA，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。

此外还有一些典型的抽取式阅读理解数据集，不过这些数据集主要是考察模型的推理能力，如 HotpotQA[20]，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[21]，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。

2.4 自由答案式

抽取式任务这种从原文中抽取一段文本作为答案的约束并不能回答所有问题需要的答案，而且从文章中概括提炼出问题的答案也是更加的符合人们的阅读方式的。于是研究人员开始转向自由答案式阅读理解任务，这类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 $A\nsubseteq D$。相关数据集如下：

MS MARCO：MS MARCO[8] 是自由答案式阅读理解数据集的代表之一，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，段落是来源于必应搜索引擎的返回的 10 个最相关的查询段落，人工的从这些选择出来的段落中概括提炼出答案。MS MARCO 数据集中的问题可能有多个答案或没有答案。

DuReader：DuReader[22] 是与 MS MARCO 类似的一个中文机器阅读理解数据集，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，此外问题类型还包括是非题（答案是 Yes/No）。

NarrativeQA：Kovcisky 等人[10] 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，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，有的问题甚至需要模型在理解整部小说或剧本的前提下才能找到答案，要求模型有更强的理解能力和推理能力。

2.5 评估方法

对于不同的 MRC 任务有不同的评估指标。对于填空式任务与多项选择式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，模型的准确率是 n/m。

抽取式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。EM 评估指标可以看做是准确率的扩展，就抽取式任务来讲，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。

对于自由答案式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[23] 和 BLEU[24]。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 1 列举了本章介绍的所有数据集及相应的评估方法。

3 神经机器阅读理解模型

3.1 通用结构

想让机器能够阅读理解文本大致需要以下流程：

1. 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. 使段落或问题中某个单词能够关注其上下文信息，更好的表示段落或问题；
3. 根据问题检索出段落中与问题最相关的部分；
4. 从检索出来的文章片段中归纳得到答案。

从整个流程可以看出，每一个步骤都有明确的目的，对应着神经网络中的某一层。因此用于 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。其中词嵌入层对应于步骤 1，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；编码层对应于步骤 2，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；交互层对应于步骤 3，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；答案输出层对应于步骤 4，作用是从段落中查找出问题的答案。

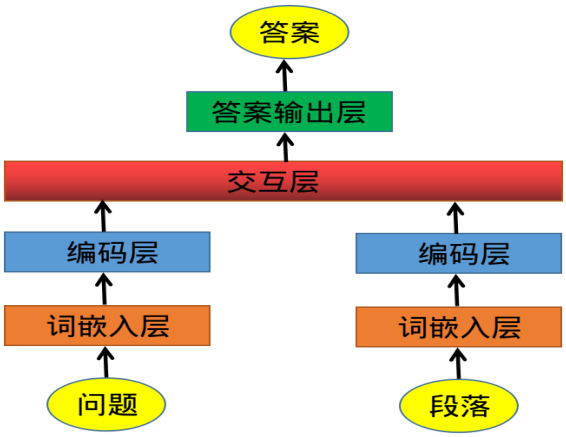

图 1　MRC 模型通用框架

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[25] 最早提出分布式表示的概念，分布式表示是将单词嵌入到一个低维向量空间中用一个低维度的稠密向量表示，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，这种词表示方法解决了 one-hot 编码的很多问题。Bengio 等人[26] 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[27] 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[28] 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

表 1　MRC 常用数据集对比，Acc 代表准确率

Table 1 Comparison of common dataset in MRC

数据集	发布时间	文章来源/数量	文章类型	问题来源/数量	答案类型	评估指标
CNN&Daily Mail[13]	2015	新闻/3×10^5	单段落型	人工合成/1.4×10^6	填空式	Acc
CBT[14]	2015	儿童读物/108	单段落型	人工合成/6.8×10^5	填空式	Acc
CLOTH[15]	2016	英语考试	单段落型	英语考试/1×10^5	填空式	Acc
MCTest[16]	2013	儿童读物/500	单段落型	众包/2×10^3	多项选择式	Acc
RACE[9]	2018	英语考试/5×10^4	单段落型	英语考试/8.7×10^5	多项选择式	Acc
SQuAD[3]	2016	维基百科/536	单段落型	众包/1×10^5	抽取式	EM/F1
TriviaQA[19]	2017	网页搜索/6.6×10^5	多段落型	搜索日志/4×10^4	抽取式	EM/F1
SQuAD 2.0[18]	2018	维基百科/536	单段落型	众包/1.5×10^5	抽取式	EM/F1
NewsQA[17]	2017	新闻/1×10^4	单段落型	众包/1×10^5	抽取式	EM/F1
HotpotQA[20]	2018	维基百科	多段落型	众包/1.13×10^5	抽取式	EM/F1
MS MARCO[8]	2016	搜索引擎/2×10^5	多段落型	搜索日志/1×10^5	自由答案式	ROUGE-L/BLEU
DuReader[22]	2018	搜索引擎/1×10^6	多段落型	搜索日志/2×10^5	自由答案式	ROUGE-L/BLEU
NarrativeQA[10]	2017	小说和电影剧本/1.5×10^3	多段落型	众包/4.6×10^4	自由答案式	ROUGE-L/BLEU

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[6] 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[29] 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[30] 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。ELMo 使用两层带有残差连接的双向 LSTM 来训练语言模型，前向和反向语言模型的目标函数如公式（1）和（2）所示。

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

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

最后的目标函数是最大化联合的前向和后向最大似然：

$$\begin{aligned}L(\Theta)=&\sum_{k=1}^{N}(\log p(t_1,\cdots,t_{k-1};\Theta))\\&+\log p(t_{k+1},\cdots,t_N;\Theta)\end{aligned} \quad (3)$$

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，将每一层的特征加权求和作为下游模型的输入。

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[31] 和 GRU[32] 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。但是这种序列式的结构不能并行计算，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[33] 提出了一种用于机器翻译的 seq2seq 结构的模型 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。关于 transformer 的细节介绍见3.3.1节。

3.1.3 交互层

交互层是整个四层网络模型中关键的一层，目的就是让段落的语义信息与问题的语义信息融合，达到对段落更深层次的理解，而交互层中最常用的方法就是注意力机制（Attention）。

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：

$$\alpha_i = \text{softmax}(f(Q, K_i))$$
$$\text{Attention}(Q, K, V) = \sum_{i=1}^{n} \alpha_i V_i \tag{4}$$

其中 Q 表示 query 的向量表示，(K_i, V_i) 代表 key-value pairs 向量表示的第 i 个值，函数 f 常采用计算方式有内积、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：

$$f(p_i, Q) = p_i^T Q \qquad \text{内积} \tag{5}$$
$$f(p_i, Q) = p_i^T W Q \qquad \text{二次型函数} \tag{6}$$
$$f(p_i, Q) = v^T \tanh(W p_i + U Q) \qquad \text{前馈神经网络} \tag{7}$$
$$f(p_i, Q) = p_i^T W^T U Q \qquad \text{双维度转换函数} \tag{8}$$

在 NLP 领域中 $K = V$，即对于两个序列，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 $Q = K = V$，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[35] 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 模型做注意力运算有两个方向：从问题到段落（Question-to-Context,Q2C），从段落到问题（Context-to-Question，C2Q）。以 Q2C 方向为例，Q2C 注意力是指将问题看做是 Q，段落看做 K, V。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 代表段落单词的个数，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：

$$\alpha_i = \text{softmax}(f(c_i, Q))$$
$$\text{Attention}(C, Q) = \sum_{i=1}^{n} \alpha_i c_i \tag{9}$$

计算得到的 $\text{Attention}(C, Q)$ 也叫段落感知的问题表示（context-aware query representation）。C2Q 注意力的计算方式类似，此时段落看作是 Q，问题看做 K, V，计算得到的 $\text{Attention}(C, Q)$ 也叫问题感知的段落表示（query-aware context representation）。

此外还可以将注意力机制分为 one-hop 和 multi-hop 形式。其中 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。与之对应的是 multi-hop，也叫“多跳结构”，可以看作是 one-hop 的堆叠。One-hop 形式下段落与问题仅仅只做一次交互计算，而对于复杂的问题通常是不能在一个句子中找出答案，需要在多个句子甚至多篇段落中推理，每一步推理过程中都会变换注意力关注的对象，因此需要 multi-hop 实现多步推理。

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。由于多项选择式任务的做法可以归结为填空式任务（将每一个选项看作是填空位置的候选项），而填空式任务又是抽取式任务的特例，所以本节主要重点介绍抽取式与自由答案式任务的输出层设计。

对于抽取式阅读理解任务，是从文章中提取出来一段连续的单词作为答案，由于提取文本的长度不固定，使得这一任务更具有挑战性。指针网络（Pointer

networks[44]）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权重分布直接输出预测结果。Wang 等人[4] 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预测答案的每一个位置，处理过程类似于 seq2seq 模型的解码过程。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实验结果表明更加高效。边界式模型的这种设计思想也被后来很多 MRC 模型采纳。抽取式模型的损失函数可以写为 $L(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \log P^S_{y^s_i} + \log P^E_{y^e_i}$。其中 θ 为模型参数，N 代表样本数目，y^s_i 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^e_i 表示第 i 个样本中标准答案的终止位置在文章中的位置。

对于自由答案式阅读理解任务，答案形式已经不是原文中某段文本，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，将段落看做是 encoder 端的输入，decoder 端根据词汇表中的单词生成答案。还有一类经典的处理生成任务的架构是 See 等人[46] 提出的指针生成网络模型（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。

3.2 经典 MRC 模型

这四层中，词嵌入层和编码层不是 MRC 模型特有的，NLP 的其它任务也都包括这两层。真正体现各个机器阅读理解模型特点的在于交互层与答案输出层，特别是交互层。因此本章着重分析各个模型在交互层中注意力机制的设计。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3.1.3节介绍的形式完全的区分开每一个模型，本文按照 Liu 等人[36] 的思路根据注意力计算的方向以及结构划分各个模型。

3.2.1 单向注意力

单向注意力通常是计算问题到段落（Q2C）注意力，为段落中每一个单词生成权重，目的是突出段落中与问题最相似的部分。Hermann 等人[2] 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模型 Attentive Reader 和 Impatient Reader，均是计算问题到文章（Q2C）的注意力，且注意力的运算方式采用前馈神经网络（公式 4）。在此基础上，Chen 等人[37] 利用双线性项（公式 3）取代原有的前馈神经网络计算方式，Kadlec 等人[38] 利用内积运算（公式 5）作为注意力计算方式。

3.2.2 双向注意力

单向注意力所能交互的信息有限，而双向注意力则可以达到两个方向的互补，提供更加全面的交互信息。Xiong 等人[5] 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 C2Q 以及 Q2C 两个方向的注意力，最后融合两个方向的注意力作为交互层的输出。Seo 等人[6] 提出（Bidirectionl Attention Flow，BiDAF）模型。同样计算两个方向（C2Q 和 Q2C）的注意力，与之前模型不同的是 BiDAF 将编码层输出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。

3.2.3 自匹配注意力

Q2C 注意力可以被认为是带着问题阅读文章，C2Q 注意力可以被认为是带着文章阅读问题。这两种注意力都属于交互注意力机制，而交互注意力过度的依赖先验信息，尤其是对于 Q2C 注意力，可能会导致模型只重视文章中与问题相关度较高的信息而忽视文章本身强调的语义信息。而自匹配注意力机制使

得文章中每一个单词可以关注到其余所有的单词，使得模型对文章达到更深层次的理解。

基于这一问题，很多模型在交互注意力的基础上添加自匹配注意力机制。如 Wang 等人[7] 提出的 RNet 模型，在 Match-LSTM[4] 的 C2Q 注意力的基础上添加一层自注意力，类似的还有 Clark 等人[60] 提出的 BiDAF++ 模型，在 BiDAF[6] 的双向注意力流基础上添加一层自注意力。

3.2.4 单跳结构

单跳结构是指段落与问题之间的交互仅仅计算一次。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[2]，AS Reader[38] 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[5]，BiDAF[6] 等。

3.2.5 多跳结构

单跳结构不能实现多步推理的效果，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加深模型对段落和问题的理解，每一次的交互计算都会适当的改变注意力关注的对象，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表示，计算下一时间步的段落和问题交互，以一种序列式的方式计算注意力，如 Impatient Reader[2]，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，如 Wang 等人[4] 提出的 Match-LSTM 模型，序列式的阅读段落，利用 LSTM 存储每个时间步计算得到的问题感知的段落表示。具体的，计算段落当前时刻的单词与问题的注意力，得到的问题感知的段落表示与当前时刻段落单词的向量表示拼接作为 LSTM 的输入。类似的模型如 RNet[7]，IA Reader[39] 等

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，如 Dhingra 等人[40] 提出的 Gated Attention Reader（GA Reader）模型，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 得到问题的向量表示，然后与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，作为下一层 BiGRU 的输入，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。Hu 等人[42] 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 2 对比了本节介绍的经典的 MRC 模型在注意力机制设计上的差异。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（C2Q+Q2C），self-attention 代表对段落做自匹配注意力运算，one-hop 代表单跳结构，multi-hop 代表多跳结构。

模型	注意力方向	推理模式
Attentive Reader[2]	Q2C	one-hop
Impatient Reader[2]	Q2C	multi-hop
Standford Reader[37]	Q2C	one-hop
AS Reader[38]	Q2C	one-hop
IA Reader[39]	Q2C	multi-hop
GA Reader[40]	C2Q	multi-hop
Match-LSTM[4]	C2Q	multi-hop
DCN[5]	Bidirectional	one-hop
BiDAF[6]	Bidirectional	one-hop
R-Net[7]	C2Q+self-attention	multi-hop
RMR[42]	Q2C+self-attention	multi-hop

表 2 基于注意力机制的模型对比

表 3 对比了经典的 MRC 模型在 SQuAD[3] 数据集上的表现[5]。

[5]统计数据源自 Yu 等人[34]

模型	EM/F1
Match-LSTM[4]	64.7/73.7
DCN[5]	66.2/75.9
BiDAF[6]	68.0/77.3
ReasoNet[43]	70.6/79.4
R-Net[7]	72.3/80.7
RMR[42]	73.2/81.8
QANet[34]	76.2/84.6

表 3 模型在 SQuAD[3] 数据集上的对比（acc 代表准确率）

3.3 基于预训练模型的 MRC

预训练模型近年来 NLP 领域获得了极大的关注度，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。在做机器阅读理解任务时，只需要设计具体任务的输出层拼接到预训练模型上进行微调即可达到很好的效果。基于预训练模型的 MRC 结构与传统结构对比如图 2 所示。从模型结构的角度看，预训练模型相当于将传统模型通用结构的编码层和交互层融合在一起，在编码的同时进行段落与问题的交互。

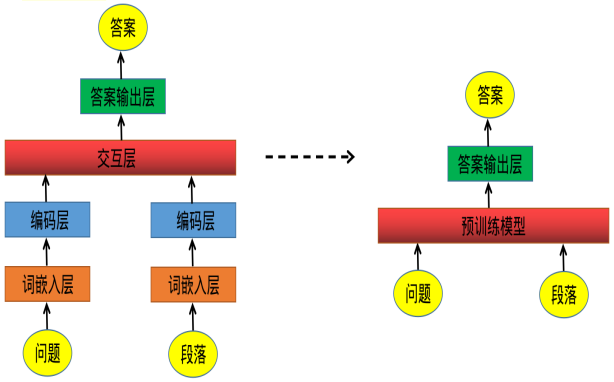

图 2 模型结构对比

3.3.1 Transformer

鉴于目前几乎所有的预训练模型都采用 transformer[33] 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序列到序列（seq2seq）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（4），自注意力机制下 $Q = K = V$，transformer 中采用的计算方式如下：

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{10}$$

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。

$$\begin{aligned}\text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\ \text{Multi-head}(Q, K, V) &= \text{concat}(\text{head}_1, \text{head}_2, \cdots, \text{head}_h)W^o\end{aligned} \tag{11}$$

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[52]）和残差连接（residual connection[53]）机制。

3.3.2 预训练模型

OpenAI[47] 提出一种生成式预训练模型（Generative Pre-Trained，GPT），使用多层的 transformer[33] 的解码端作为特征提取器。预训练阶段的目标函数是单向语言模型的目标函数，见公式（1）。

GPT 属于自回归语言模型，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 Devlin 等人 [48]

表 2　预训练模型对比

Table 2 Comparison of pre-trained model

模型	任务	模型结构	介绍
ELMo[30]	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[47]	前向 LM	Transformer	首次采用预训练＋微调形式
BERT[48]	MLM+NSP	Transformer	利用掩码语言模型（MLM）和下一句预测（NSP）共同作为训练任务
MT-DNN[54]	MLM+NSP	Transformer	预训练过程与 BERT 一致，微调阶段采用多任务学习
RoBERTa[49]	MLM	Transformer	采用动态掩码机制，去除 NSP 任务
UNILM[50]	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[51]	MLM+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）

提出 BERT 预训练模型，与 GPT 不同之处在于预训练方式上采用的降噪自编码方式，随机掩盖掉一些单词，在输出层获得掩盖位置的概率分布，让模型根据掩盖位置的上下文预测这个单词，这种机制也叫掩码语言模型（Masked Language Model，MLM）或双向语言模型。MLM 的目标函数为：

$$L(\Theta) = \sum_{i=1}^{N} \log P(t_k | t_1, t_2, \cdots, t_{k-1}, t_{k+1}, \cdots, t_N) \tag{12}$$

除了 MLM 任务外，还利用下一句预测（Next Sentence Prediction，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[3] 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 开启了 NLP 领域预训练模型的时代，此后很多更加强大的预训练模型相继提出。这些模型大部分都是基于 BERT 的改进模型，主要是针对 BERT 预训练阶段的两个任务 MLM 和 NSP 做改进，如 RoBERTa[49] 模型，使用动态掩码替换 BERT 的静态掩码同时去除 NSP 任务；UNILM[50] 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；ALBERT[51] 模型利用句子顺序预测（Sentence Order Prediction，SOP）任务改进了 BERT 的 NSP 任务；除了在预训练阶段改进训练任务外，MTDNN[54] 模型在微调阶段引入了多任务学习机制，使用多个任务来微调模型参数使得模型具有更好的泛化型。

3.3.3　迁移模型到 MRC 任务

预训练模型具有强大的文本表征能力，应用到机器阅读理解任务上根据具体的任务特点设计不同的微调网络结构即可。迁移预训练模型的方式比较灵活，可以将预训练模型迁移在传统模型通用结构中除了答案预测层外的任意一层，而3.1.4节所介绍的两种输出层都可以直接拼接在预训练模型上。

表 3 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 4 对比了几个预训练模型在两个常用的 MRC 数据集上的表现。

模型	SQuAD 2.0[18]	RACE[9]
	EM/F1	Acc
GPT_{v1}[47]	-	59.0
$BERT_{large}$[48]	80.0/83.1	72.0
XLNet[55]	86.4/89.1	81.8
RoBERTa[49]	86.8/89.8	83.2
ALBERT[51]	88.1/90.9	86.5

表 4 预训练模型表现对比

4 面临的主要挑战

以上介绍了典型的基于神经网络的机器阅读理解模型，这些模型所实验的数据集大多是 SQuAD[3] 等抽取式阅读理解数据集。Weissenborn 等人[59] 专门设计了一个不包含交互层的网络模型 FastQA，仅仅在编码层的输入中额外添加两个特征：binary 和 weighted。其中 binary 特征表示段落中的单词是否出现在问题中，weighted 特征表示段落中的单词与问题的相似度。FastQA 仅仅添加了这两个额外的特征就在 SQuAD 数据集上达到很好的效果，优于很多之前介绍的基于复杂交互机制的模型。FastQA 的提出甚至质疑设计复杂交互机制的必要性，但同时也反映出 SQuAD 等抽取式数据集难度不高，简单的从一篇段落中抽取某一片段作为答案确实很难考察模型的阅读理解与推理能力，而且这种限制也不符合真实场景中的阅读理解。

基于以上的问题，为了能够真实的考察模型的阅读理解与推理能力，研究人员在原来的阅读理解任务基础上提出更加复杂的问题与任务，如要求模型判断问题是否可以根据给定的段落找出答案；要求模型在多篇段落中逐步推理寻找答案等。这些任务的提出是为了使得相应的阅读理解任务更加的贴近真实场景下人们的阅读理解形式。我们将这些任务看作是阅读理解任务目前面临的挑战问题。本章主要介绍三个挑战问题：带有无答案问题的阅读理解，多段落阅读理解，对话型阅读理解。

4.1 带有无答案问题的阅读理解

早期的 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有限的，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。这两种问题都属于无答案问题，不能从给定的文章中找到问题的答案。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。

对于带有无答案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）判别无答案问题模块。答案抽取模块用来预测出答案在文章中的位置，第三章所介绍的 MRC 模型大部分都可以作为答案抽取模块，判别无答案问题模块用来判断当前问题是否可以回答。Clark 等人[60] 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无答案情况的网络层，此时损失函数定义如下：

$$L_{joint} = -\log(\frac{(1-\delta)e^{z} + \delta e^{\alpha_a \beta_b}}{e^{z} + \sum_{i=1}^{l_p}\sum_{j=1}^{l_p} e^{\alpha_i \beta_j}}) \tag{13}$$

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那么 $\delta = 1$，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（17）可以看出预测的答案跨度分数 α_a, β_b 和判断无答案问题的分数 z 是共同归一化的。Hu 等人[61] 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的无答案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 + 判别无答案问题模块，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：

$$L_{indep-span} = -\log(\frac{e^{\widetilde{\alpha}_{\widetilde{a}}\widetilde{\beta}_{\widetilde{b}}}}{\sum_{i=1}^{l_p}\sum_{j=1}^{l_p}\widetilde{\alpha}_i\widetilde{\beta}_j}) \quad (14)$$

$$L_{indep-unknown} = -(1-\delta)\log\sigma(z) - \delta\log(1-\delta(z)) \quad (15)$$

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。$L_{indep-unknown}$ 代表判断问题无答案的损失函数，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：

$$L_{Read} = L_{joint} + \gamma L_{indep-span} + \lambda L_{indep-unknown} \quad (16)$$

γ 和 λ 是两个超参数。实验表明去掉 $L_{indep-unknown}$ 后模型在判断无答案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 Verify 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[47] 中预测无答案的概率。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，具体的就是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。关于处理无答案问题阅读理解任务的其它相关模型可以参考[62, 63]。

4.2 多段落阅读理解

目前大多数的研究热点集中于单段落阅读理解，仅需要从一个段落上寻找答案，这对模型的阅读理解能力要求不高，此外真实场景中人们往往是从多篇段落中寻找与问题相关的答案最后互相比对得出最准确的答案。多段落阅读理解任务，一个问题 Q 会对应着多个相关的段落 $(D_1, D_2, \cdots, D_K)$，模型需要从这 K 个段落中寻找最佳答案 A，建模概率

$$P(A|D_1, D_2, \cdots, D_k, Q)$$

多段落阅读理解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[29] 提出利用检索 + 阅读（Retrieve+Read）的模式处理 open-domain 问答，先利用检索模块（Document Retriever）从维基百科中获取 5 个与问题最相关的段落，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，用来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。

多段落阅读理解的难点在于模型需要阅读多篇文章，更重要的是每篇文章都有可能包含与标准答案类似的句子，而有些是错误的答案，模型需要排除众多的干扰项选择最正确的答案。想要达到这个目的，要求模型能够解决跨段落实体消歧和跨段落答案验证等问题。

Tan 等人[64] 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，答案生成模块采用 seq2seq 模型，其中 encoder 端的输入是问题和段落的向量表示，同时将片段抽取模块的输出作为额外的特征拼接到段落中。实验证明 S-Net 在 MS MSRCO 数据集上的效果要显著地优于 R-Net[7]，ReasoNet[43] 这些单独做片段抽取任务的模型。

由于不同的段落都有可能包含与标准答案类似的句子，但是有些答案并不是正确的。基于这个问题，Wang 等人[65] 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。具体的就是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。这样做的原因是经观察发现，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。关于解决多段落阅读理解问题的相关工作可以参考[66, 67, 68]。

4.3 对话型阅读理解

无论是单段落阅读理解还是多段落阅读理解任务，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。而在真实场景中人们是通

过多轮对话形式来交流的，每一轮的问题和答案都是基于前面的问答情况。所以针对对话型阅读理解问题，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。即建模概率：

$$P(A_i|D, Q_i, Q_{i-1}, \cdots, Q_{i-k}, A_{i-1}, \cdots, A_{i-k}) \quad (17)$$

目前典型的对话型阅读理解数据集有 CoQA[11] 以及 QuAC[69]。不同之处在于 CoQA 数据集的答案形式较为简单，类似于 SQuAD[3]，但是包含有是非问题（答案是 yes/no）以及无答案问题。

对于这类任务模型必须解决指代消解问题以及如何利用对话历史信息。Reddy 等人[11] 首先提出将前几轮的问题与答案结合到段落中，从而能够利用历史的对话信息回答当前轮的问题。Choi 等人[69] 利用 BiDAF++[60] 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段落中的单词是否是历史答案，将问题的轮次作为额外特征添加在问题向量上。

Huang 等人[70] 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。因此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抽取式的做法在 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抽取式模型的效果好，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能力是值得进一步研究的方向。由于预训练模型 UNILM[50] 改进了 BERT 的训练任务，增加了自回归语言模型以及 seq2seq 语言模型，使得其在生成式任务上的效果很好，在 CoQA 数据集上远远的超过于 Reddy 等[11] 提出的生成式基准模型。关于如何迁移预训练模型在对话型阅读理解任务上的相关工作可以参考[56, 71, 57]。

5 讨论

本章主要讨论机器阅读理解模型的应用以及目前面临的主要问题和未来的发展趋势。

5.1 机器阅读理解应用

智能客服 客服机器人是一种基于自然语言处理技术的模拟人类进行对话的服务程序，通过文字或语音与客户进行多轮交流。将用户的查询需求看做问题，通过阅读产品说明文档，利用机器阅读理解模型回答用户对产品的相关问题。

辅助决策 将机器阅读理解模型引入专业性较强的领域，如医疗病例报告、法律裁判文书等，可以帮助用户更好的决策。以医疗为例，Suster 等人[72] 发布了关于医疗的阅读理解数据集 CliCR，包含着大量的病例报告，同时设计了包括确诊疾病、治疗用药等问题，这些与医生的日常工作息息相关。模型如果对数据集中的问题与文本有较深的理解就可以用于临床辅助医生诊断。

智能问答 搜索引擎的目标是根据用户的查询返回相关度较高的网页，机器阅读理解在搜索引擎中的一个重要应用就是智能问答。将网页内容视为文章，从网页文本中抽取或者生成最相关的答案，避免用户主动从网页中寻找答案。

5.2 面临的主要问题

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[73] 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，这表明模型很大程度上基于关键词匹配方式，离真正的阅读理解水平还很远。

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。

缺乏可解释性 目前绝大多数的机器阅读理解模型基于深度学习，模型缺乏可解释性。而某些场景下的阅读理解任务可解释性就尤为重要，如上面提到的医疗、法律领域，模型做出判断后一定要给出医学解释或法律条文。

5.3 未来的发展趋势

多模态融合 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。

引入外部知识 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，模型的性能大概率会显著提高。

结合其它 NLP 技术 MRC 任务涉及到 NLP 领域的许多关键任务，如文本生成、序列标注、句法分析等。此外 MRC 任务中的某些问题可以通过引入其它的技术解决，如通过指代消解技术可以更好的处理段落中代词所指代的实体，或利用实体识别技术区分出命名实体单词，这些都有助于提高答案的准确性。

6 总结

本文主要回顾了神经机器阅读理解近年来的研究进展，对比了各个不同的阅读理解任务以及介绍了相应的数据集和评估指标，分析了神经机器阅读理解模型通用结构的每一层所用到的技术，并且详细讨论了各个模型交互层的设计与差异。针对一些复杂形式的任务，我们将其视为机器阅读理解任务目前面临的挑战并且介绍了对应的解决方案。最后我们列举了一些机器阅读理解的应用场景，讨论了目前面临的问题和未来的研究趋势。