软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(7):2095−2126 [doi: 10.13328/j.cnki.jos.006048]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于神经网络的机器阅读理解综述*

顾迎捷[1,2], 桂小林[1,2], 李德福[1,2], 沈 毅[1,2], 廖 东[1,2]

[1](西安交通大学 电子与信息工程学部,陕西 西安 710049)

[2](陕西省计算机网络重点实验室(西安交通大学),陕西 西安 710049)

通讯作者: 桂小林, E-mail: xlgui@mail.xjtu.edu.cn

摘 要: 机器阅读理解的目标是使机器理解自然语言文本,并能够正确回答与文本相关的问题.由于数据集规模的制约,早期的机器阅读理解方法大多基于人工特征以及传统机器学习方法进行建模.近年来,随着知识库、众包群智的发展,研究者们陆续提出了高质量的大规模数据集,为神经网络模型以及机器阅读理解的发展带来了新的契机.对基于神经网络的机器阅读理解相关的最新研究成果进行了详尽的归纳:首先,概述了机器阅读理解的发展历程、问题描述以及评价指标;然后,针对当前最流行的神经阅读理解模型架构,包括嵌入层、编码层、交互层和输出层中所使用的相关技术进行了全面的综述,同时阐述了最新的 BERT 预训练模型及其优势;之后,归纳了近年来机器阅读理解数据集和神经阅读理解模型的研究进展,同时,详细比较分析了最具代表性的数据集以及神经网络模型;最后展望了机器阅读理解研究所面临的挑战和未来的研究方向.

关键词: 机器阅读理解;自然语言处理;注意力机制;神经阅读理解模型

中图法分类号: TP18

中文引用格式: 顾迎捷,桂小林,李德福,沈毅,廖东.基于神经网络的机器阅读理解综述.软件学报,2020,31(7):2095−2126. http://www.jos.org.cn/1000-9825/6048.htm

英文引用格式: Gu YJ, Gui XL, Li DF, Shen Y, Liao D. Survey of machine reading comprehension based on neural network. Ruan Jian Xue Bao/Journal of Software, 2020,31(7):2095−2126 (in Chinese). http://www.jos.org.cn/1000-9825/6048.htm

Survey of Machine Reading Comprehension Based on Neural Network

GU Ying-Jie[1,2], GUI Xiao-Lin[1,2], LI De-Fu[1,2], SHEN Yi[1,2], LIAO Dong[1,2]

[1](Faculty of Electronics & Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

[2](Shaanxi Province Key Laboratory of Computer Network (Xi'an Jiaotong University), Xi'an 710049, China)

Abstract: The task of machine reading comprehension is to make the machine understand natural language text and correctly answer text-related questions. Due to the limitation of the dataset scale, most of the early machine reading comprehension methods were modeled based on manual features and traditional machine learning methods. In recent years, with the development of knowledge bases and crowdsourcing, high quality large-scale datasets have been proposed by researchers, which has brought a new opportunity for the advance of neural network models and machine reading comprehension. In this survey, an exhaustive review on the state-of-the-art research efforts on machine reading comprehension based on neural network is made. First, an overview of machine reading comprehension, including development process, problem formulation, and evaluation metric, is given. Then, a comprehensive review is conducted of related technologies in the most fashionable neural reading comprehension framework including the embedding layer, encoder layer, interaction layer, and output layer as well as the latest BERT pre-training model and its advantages are discussed. After that, this paper concludes the recent research progress of machine reading comprehension datasets and neural reading comprehension model, and gives a comparison

* 基金项目: 国家重点研发计划(2018YFB1800304); 陕西省重点研发项目(2019GY-005, 2017ZDXM-GY-011, 2020GY-033)

Foundation item: National Key Research and Development Program of China (2018YFB1800304); Key Development Program in Shaanxi Province of China (2019GY-005, 2017ZDXM-GY-011, 2020GY-033)

收稿时间: 2019-04-26; 修改时间: 2019-06-29; 采用时间: 2020-02-13; jos 在线出版时间: 2020-04-21

and analysis of the most representative datasets and neural network models in detail. Finally, the research challenges and future direction of machine reading comprehension are presented.

Key words: machine reading comprehension; natural language processing; attention mechanism; neural reading comprehension model

教会机器理解自然语言文本并能够正确回答相关问题,是自然语言处理(natural language processing,简称NLP)领域最具挑战性的任务之一,同时也是该领域追求的最终目标[1].对机器阅读理解(machine reading comprehension,简称 MRC)的研究有着悠久的历史,早在 20 世纪 70 年代,研究者们就已经发现该任务是测试机器语言理解能力的一种非常重要的方法[2].然而在当时,由于数据集规模以及受计算机硬件设备的制约,该领域的研究被忽视了几十年,直到最近它才受到了国内外学者的广泛关注.MRC 之所以在近期能够得到快速发展,其最重要的两个原因是:(1) 百科类知识库(例如 WikiPedia)、众包群智服务模式(crowd)的发展,推动了大规模的阅读理解监督数据集的创建,这类数据集以(段落,问题,答案)的三元组形式存储,为 MRC 模型的性能测评提供了机会;(2) 计算机硬件设备的性能提升(尤其是 GPU,TPU 的快速发展),推动了以神经网络为代表的深度学习技术[3]的发展,从而使神经阅读理解模型的构建与应用成为了可能.

与信息检索(information retrieval,简称 IR)[4]不同,MRC 不是简单地让机器根据问题匹配文本数据库中相似度最高的字符串,而是让机器能够理解用户所描述的自然语言问题,这些问题的答案可能存在于文本段落中,可能是“是或否”,也有可能是无法回答的,甚至需要机器根据自己的理解生成或计算出正确的答案.例如:在查询“在跑步机上跑多久才能消耗 10 个麦当劳巨无霸的热量”时,在 MRC 中,机器会找到麦当劳官网的巨无霸卡路里说明,然后在健身网站上找到跑步机平均一小时的热量消耗,最后将两者进行计算,返回给用户准确的答案;而 IR 只会简单地给出与“麦当劳巨无霸”和“跑步机”字符串相似度最大的结果页面.

本文归纳并分析近年来MRC领域中神经网络模型以及大规模数据集的研究进展,并进行较为全面的综述.第 1 节描述 MRC 的发展历程与任务定义.第 2 节阐述基于神经网络的机器阅读理解模型架构,重点围绕词嵌入层、编码层、交互层以及输出层中所使用的技术进行综述,同时介绍近期最流行的 BERT 预训练模型.第 3 节介绍具有代表性的数据集以及神经网络模型,并分别对两者进行详细的分析和比较.最后,在第4节讨论MRC的发展趋势以及未来的研究挑战,以期对其在国内的研究起到一定的推动作用.

1 机器阅读理解的发展历程与任务定义

1.1 机器阅读理解的发展历程

机器阅读理解在近半个世纪以来经历了 3 个阶段的发展:从 20 世纪 70 年代开始,利用基于规则的方法[5]构建早期系统的基于规则(rule-based)时代,到尝试使用有监督的机器学习模型[6]解决 MRC 任务的机器学习(machine learning)时代,再到最近利用神经网络(深度学习)[1]搭建神经阅读理解模型的神经网络(neural network)时代.下面将详细阐述机器阅读理解经历的这 3 个时代.

1.1.1 基于规则时代(1970s~2012)

构建自动阅读理解系统的历史可以追溯到 20 世纪 70 年代,在当时,研究者们已经意识到了机器阅读理解是测试机器语言理解能力的一种非常重要的方法.早期最著名的工作之一是由 Lehnert[2]在 1977 年提出的 QUALM 系统,该系统首次表明了回答问题时文本语境的重要性,但由于该系统受限于需要手工编码的脚本,因此很难得到更广泛的应用.在 20 世纪 80 年代~90 年代之间,由于 MRC 任务的复杂性远超当时的技术水平,因此这十几年来该领域的研究被长期搁置.直到 1999 年,Hirschman[7]提出了一个用于开发与测试的数据集,人们才对 MRC 重新有了兴趣.该数据集包含 60 个故事,主要由一些简短的事实类问答对,例如 who、what、when、where 以及 why 等问题与对应答案组成,它只需要系统返回包含正确答案的句子.针对上述数据集,我们提出了 DEEP READ 系统,该系统主要采用基于规则的方法,例如在规则中加入各类特征(词干提取、语义类标识、指代消解、BoW 等特征)以实现对文本的理解,最终达到了 30%~40%的精确度.同样地,在 21 世纪早期,也是由于缺少大规模数据集,导致 MRC 的研究进展十分缓慢.

1.1.2 机器学习时代(2013~2015)

在 2013 年~2015 年之间,由于机器学习方法的崛起,研究者们尝试将机器阅读理解定义为一种监督学习(supervised learning)问题[8],他们收集人工标注的(段落,问题,答案)三元组训练数据,希望能够通过已标注的数据训练一个统计学模型,使得该模型在测试时能将(段落,问题)映射到对应的答案.其中,最为重要的工作之一是Richardson[9]在 2013 年提出的 MCTest 数据集(http://research.microsoft.com/mct),该数据集的提出直接推动了当时机器学习模型的发展[10−12].这些模型大多数是基于简单的最大边缘(max-margin)学习框架,通过加入丰富的语义特征集(句法依存、共指消解、语义框架、词嵌入等),实现对(段落,问题,答案)三元组的拟合.虽然这些机器学习模型相比于早期基于规则的方法取得了一定的进展,例如,针对 MCTest-500 数据集,上述 3 种模型相比于基线模型(63.33%)各自提升了 0.42%[10]、4.5%[11]以及 6.61%[12].但我们发现,上述机器学习模型带来的性能提升相当有限.研究者们通过分析认为,主要原因有以下两点:(1) 机器学习模型主要依赖于现有的语言工具,例如依存解析器(DP)以及语义角色标记系统(SRL)来实现特征的提取,但这些工具大都是由单一领域的数据训练所得,泛化能力较弱,因此对于 MCTest 数据集来说,这些特征反而是噪声;(2) MCTest 数据集的规模太小,不足以支持上述模型的训练(只有 1 480 个例子供模型训练).

1.1.3 神经网络时代(2015~至今)

Hermann 等人[1]在 2015 年提出了一个新型的、大规模的监督训练数据集 CNN/Daily Mail(约 126 万条训练数据,https://cs.nyu.edu/~kcho/DMQA/),同时,针对上述数据集提出了一个基于注意力的 LSTM 模型 THE ATTENTIVE READER,该模型的性能远超传统 NLP 方法(约 12.9%),标志着机器阅读理解正式进入了神经网络时代.虽然 CNN/Daily Mail 数据集的规模足以达到深度学习模型的训练要求,但由于 CNN/Daily Mail 数据集属于完形填空(cloze-type)类型,它的问题不符合人类的自然语言描述.因此,为了突破 CNN/Daily Mail 数据集的局限性,Rajpurkar 等人[13]在 2016 年提出了一个著名的数据集 Stanford Question Answering Dataset(SquAD, https://rajpurkar.github.io/SQuAD-explorer/).得益于百科类知识库以及众包群智服务模式的发展,SQuAD 数据集从 536 篇 Wikipedia 段落中收集了 107 785 个问题答案对;同时,由于 SQuAD 数据集中每一个答案都是与问题相关的段落文本中的一段跨距(aspan),使得该数据集成为了学术界第一个包含大规模自然语言问题的阅读理解数据集.借助于 SQuAD 这一高质量的 MRC 数据集,两年来,研究者们提出了一系列全新的神经阅读理解模型,同时刷新着该数据集榜单上 EM 与 $F1$ 值(数据集的一种评价方法)的记录——从 2016 年原作者提出的 Logistic Regression 基线模型(EM 值为 40.4%,$F1$ 值为 51%)到 2018 年 10 月由 Google 公司提出的远超人类(EM 值为 82.304%,$F1$ 值为 91.221%)的 BERT 模型(EM 值为 87.433%,$F1$ 值为 93.16%)[14].尽管 SQuAD 数据集的提出是 MRC 领域的里程碑,但这并不意味着该数据集代表着整个 MRC 领域的终极目标,因此之后,学者们提出了更多大规模的具有挑战性的数据集,用于处理以前没有解决的问题,例如抽取式(extractive)类别的 TriviaQA(http://nlp.cs.washington.edu/triviaqa/)[15],WikiQA(http://aka.ms/WikiQA)[16],NewsQA(https://datasets.maluuba.com/NewsQA)[17],SQuAD 2.0(https://rajpurkar.github.io/SQuAD-explorer/)[18],SearchQA(https://github.com/nyu-dl/SearchQA)[19]等,多项选择(multiple choice)类别的 SciQ(http://allenai.org/data.html)[20],ARC(http://data.allenai.org/arc)[21],RACE(http://www.cs.cmu.edu/glai1/data/race/)[22],TQA(http://textbookqa.org)[23],MCScript(http://www.sfb1102.uni-saarland.de/?page_id=2582)[24]等,完形填空类别的 CBT(http://fb.ai/babi/)[25],CLOTH(https://www.cs.cmu.edu/~glai1/data/cloth/)[26]等,会话(conversation)类别的 CoQA(https://stanfordnlp.github.io/coqa/)[27],QuAC(http://quac.ai/)[28],CSQA(https://github.com/iitm-nlp-miteshk/AmritaSaha/tree/master/CSQA)[29],SQA(http://aka.ms/sqa)[30],CQA(http://nlp.cs.tau.ac.il/compwebq)[31]等,生成式(generative)类别的 NarrativeQA(http://deepmind.com/publications)[32],MSMARCO(http://www.msmarco.org)[33]等以及最新的多跳(multi-hop)推理的 HotpotQA(https://HotpotQA.github.io)[34].之后,国内学者也开始陆续提出中文领域的 MRC 数据集,具有代表性的有 People Daily/CFT(同时也是完型填空,http://hfl.iflytek.com/chinese-rc/)[35]以及 DuReader(同时也是生成式,http://ai.baidu.com/broad/download?dataset=dureader)[36].除了上述标准化的数据集以外,还有针对开放域(open-domain)的阅读理解任务,本文将在第 3.1 节对各类数据集进行详细的归纳分析.与此同时,各种大规模 MRC 数据集的提

出也推动着神经阅读理解模型的发展,从刚开始人们采用的简单记忆网络模型[37],到 match-LSTM+Ptr-Net 模型[38],再到具有代表性的通用四层架构,最后到最新的 Transformer 架构[39].逐渐丰富的神经阅读理解模型对各类数据集中段落与问题的理解能力以及泛化能力越来越强,本文将在第 3.2 节对现阶段最具代表性的神经机器阅读模型进行详细的归纳分析,总结其适用范围与优缺点.总的来说,神经网络时代下的机器阅读理解仍然处于起步阶段,即使是使用最先进的神经阅读理解模型.至今为止,上述大部分数据集仍不能得到完美解决,有一些甚至与人类水平还有较大差距,因此,MRC 的研究与发展仍有很长的路要走.

1.2 机器阅读理解的任务定义

1.2.1 问题描述

机器阅读理解任务可以形式化成一个有监督的学习问题:给出三元组形式的训练数据(C,Q,A),其中,C 表示段落,Q 表示与之相关的问题,A 表示对应的答案.我们的目标是学习一个预测器 f,能够将相关段落 C 与问题 Q 作为输入,返回一个对应的答案 A 作为输出:

$$f:(C,Q)\rightarrow A \tag{1}$$

一般地,我们将段落表示为 $C=\{w_1^C,w_2^C,...,w_m^C\}$,将问题表示为 $Q=\{w_1^Q,w_2^Q,...,w_n^Q\}$,其中,$m$ 和 n 分别为段落 C 的长度和问题 Q 的长度,所有 w 都属于预先定义的词典 $\mathcal{V}$.由于数据集的类型不同,问题和答案会被表示成不同的形式,表 1 中的 6 个例子分别来自 6 种类型中具有代表性的数据集:CBT、SciQ、SquAD、CoQA、NarrativeQA 以及 HotpotQA.具体地,可以将数据集定义为以下 6 种类型.

(1) 完形填空:在这类数据集中,机器的目标是根据问题和当前段落,从预定义的选项集合 $\mathcal{A}$ 中选出正确答案 a,并填入问题的空白处.例如,在 CBT 数据集中,$|\mathcal{A}|$=10.这类问题的答案往往是一个单词或实体;

(2) 多项选择:在这类数据集中,机器的目标是根据问题和当前段落信息,从包含正确答案的 k(k 一般为 4)个设定的选项集合 $\mathcal{A}=\{a_1,...,a_k\}$ 中选出正确答案 a,a 可以是一个单词、一个短语甚至一个句子;

(3) 抽取式:也可称为跨距预测类型数据集(span prediction),在这类数据集中,机器的目标是根据问题在当前段落中找到正确的答案跨距,因此在这类数据集中,我们可以将答案表示为(a_{start},a_{end}),其中,$1\leqslant a_{start}\leqslant a_{end}\leqslant m$;

(4) 会话:在这类数据集中,目标与机器进行交互式问答,因此,答案可以是文本自由形式(free-text form),即可以是跨距形式,可以是“不可回答”形式,也可以是“是/否”形式等等;

(5) 生成式:在这类数据集中,问题的答案都是人工编辑生成的(human manual generated),不一定会以片段的形式出现在段落原文中,机器的目标是阅读给出段落的摘要甚至全文,之后根据自身的理解来生成问题的答案;

(6) 多跳推理:在这类数据集中,问题的答案无法从单一段落或文档中直接获取,而是需要结合多个段落进行链式推理才能得到答案.因此,机器的目标是在充分理解问题的基础上从若干文档或段落中进行多步推理,最终返回正确答案.

Table 1 A few examples from representative datasets

表 1 一些来源于代表性数据集的例子

序号	描述
(1)	CBT(完形填空) 段落:…“Never mind her, go on!” interrupted Happy Jack. Then I flew all around the barn, but I didn’t see any one there but that ugly little upstart, Bully the English sparrow, and he wanted to pick a fight with me right away. “Tommy looked very indignant. “Never mind him, go on!” 问题:cried Happy XXXXX impatiently 选项:Brown|Jack|Pussy|Sparrow|Tit|Tommy|doorsteps|left|right|upstart 答案:*Jack*

Table 1 A few examples from representative datasets (Continued)

表 1 一些来源于代表性数据集的例子(续)

序号	描述
(2)	SciQ(多项选择) 段落:Without Coriolis Effect the global winds would blow north to south or south to north. But Coriolis makes them blow northeast to southwest or the reverse in the Northern Hemisphere. The winds blow northwest to southeast or the reverse in the southern hemisphere. 问题:What phenomenon makes global winds blow northeast to southwest or the reverse in the northern hemisphere and northwest to southeast or the reverse in the southern hemisphere? 选项:1) coriolis effect; 2) muon effect; 3) centrifugal effect; 4) tropical effect 答案:1) coriolis effect
(3)	SQuAD(抽取式) 段落:Precipitation forms as smaller droplets coalesce via collision with other rain drops or ice crystals *within a cloud*. Short, intense periods of rain in scattered locations are called "showers" 问题:Where do water droplets collide with ice crystals to form precipitation? 答案:within a cloud
(4)	CoQA(多轮对话) 段落:*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. Jessica had… 问题 1:Who had a birthday? 答案 1:Jessica 问题 2:How old would she be? 答案 2:80 问题 3:Did she plan to have any visitors? 答案 3:Yes
(5)	NarrativeQA(生成式) 段落(摘要片段):... Peter's former girlfriend Dana Barrett has had a son, Oscar... 段落(故事片段):*DANA* (setting the wheel brakes on the buggy): Thank you, Frank. I'll get the hang of this eventuall. //She continues digging in her purse while Frank leans over the buggy and makes funny faces at the baby, OSCAR, a very cute nine-month old boy. //*FRANK* (to the baby): Hiya, Oscar. What do you say, slugger? //*FRANK* (to DANA): That's a good-looking kid you got there, Ms. Barrett. 问题:How is Oscar related to Dana? 答案:her son
(6)	HotpotQA(多跳推理) 段落 A:[1] Return to Olympus is the only album by the alternative rock band *Malfunkshun*. [2] It was released after the band had broken up and after lead singer Andrew Wood (later of mother love bone) had died of a drug overdose in 1990. [3]… 段落 B:[4] Mother Love Bone was an American rock band that formed in Seattle, Washington in 1987. [5]… [6] Frontman Andrew Wood's personality and compositions helped to catapult the group to the top of the burgeoning late 1980s/early 1990s Seattle music scene. [7] Wood died only days before the scheduled release of the band's debut album, "Apple", thus ending the group's hopes of success. [8]… 问题:What was the former band of the member of Mother Love Bone who died just before the release of "Apple"? 答案:Malfunkshun 支持段落:1,2,4,6,7

1.2.2 评价指标

机器阅读理解中对于模型的评价指标主要由数据集的类型决定.

(1) 对于完形填空和多项选择类型的任务,由于答案都是来源于已经给定的选项集合$\mathcal{A}$,因此使用Accuracy这一指标最能直接反映模型的性能,即,在问题数据中模型给出的正确答案数n占总问题m的百分比:

$$Accuracy=\frac{n}{m} \tag{2}$$

(2) 对于抽取式和多跳推理类型的任务,需要对模型预测的答案字符串和真实答案进行比对,因此一般使用Rajpurkar等人提出的Exact Match(EM)和$F1$值[13].EM是指数据集中模型预测的答案与标准答案相同的百分比,$F1$值是指数据集中模型预测的答案和标准答案之间的平均单词的覆盖率,P代表准确率(precision),R代表召回率(recall).其中,多跳推理数据集,例如HotpotQA,还提出了针对支持证据(supporting fact)的EM和$F1$值,以此来衡量模型是否真正理解问题并做出回答而非盲目猜测,同时将两者结合得到最终的联合EM和$F1$值(joint

EM,*F*1)[34]:

$$F1=\frac{2\times P\times R}{(P+R)} \tag{3}$$

(3) 对于会话类型的任务,由于其答案是文本自由形式,因此并没有一种通用的评价指标,该类任务的评价指标主要由数据集本身决定.例如:CoQA 数据集使用了 *F*1 值;而 QuAC 数据集除了 *F*1 值以外,还提出了 HEQQ 以及 HEQD 两种新的评价指标[28].

(4) 对于生成式类型的任务,由于答案是人工编辑生成的,而机器的目标是使生成的答案最大限度地拟合人工生成的答案,因此该类任务一般使用机器翻译任务中常用的 BLEU-4[40]和 Rouge-L[41]两种指标.其中,BLEU 的本质是依靠计算共现词频率来判断机器生成答案和人工编辑答案的相似或接近程度,在这里,BLEU-4 指的是采用四元精度(4-gram precision)对原始的 BLEU 算法进行改进后的版本(*N*=4),其中,BP 是惩罚因子:

$$P_n=\frac{\sum_{C\in\{Candidates\}}\sum_{n\text{-}gram\in C}Count_{clip}(n\text{-}gram)}{\sum_{\tilde{C}\in\{Candidates\}}\sum_{n\text{-}gram''\in\tilde{C}}Count(n\text{-}gram'')} \tag{4}$$

$$BP=\begin{cases}1, & c>r\\ \mathrm{e}^{(1-r/c)}, & c\leqslant r\end{cases} \tag{5}$$

$$BLEU=BP\cdot\exp\left(\sum_{n=1}^{N}w_n\log P_n\right) \tag{6}$$

Rouge(recall-oriented understudy for gisting evaluation)同时也是自动文本摘要任务的重要评测指标,与 BLEU 类似,它也是通过将自动生成的语句或摘要与人工编辑进行比较计算后得出相应的分值,以此来衡量生成本文的质量.而 Rouge-L 是采用了最长公共子序列(longest common subsequence,简称 LCS)的 Rouge 的一种改进版本,其中,*X* 表示长度为 *m* 的正确答案,*Y* 为模型生成的长度为 *n* 的答案:

$$R_{lcs}=\frac{LCS(X,Y)}{m} \tag{7}$$

$$P_{lcs}=\frac{LCS(X,Y)}{n} \tag{8}$$

$$F_{lcs}=\frac{(1+\beta)^2R_{lcs}P_{lcs}}{R_{lcs}+\beta^2P_{lcs}} \tag{9}$$

除此之外,有些生成式数据集,例如,NarrativeQA 还采用 BLEU-1、Meteor[42]以及 MRR 等指标进行评测.

与本文工作相似,Gardner[43]于 2016 年在自己的 Github 上以博客的形式写了一篇关于阅读理解(RC)的综述文章,但他只是简单地分析了部分数据集以及列举一些神经网络的方法;Arivuchelvan[44]于 2017 年发表了一篇关于 RC 系统的综述,但其重点在于归纳机器学习方法而不是深度学习技术,同时,文章中也没有针对数据集进行分析;Lai 等人[45]于 2018 年发表了一篇关于深度学习技术在答案选择(answer selection)中的应用综述,但他仅仅归纳了答案选择任务下神经网络模型的演化历程,除此之外,Lai 等人只简单分析了 4 个数据集.与上述英文综述文献的不同之处在于:本文较为全面地分析了近年来机器阅读理解的发展历程,并对相关数据集和神经网络模型进行了详细归纳,以期对其在国内的研究起到一定的推动作用.

2 基于神经网络的机器阅读理解架构

基于端到端神经网络的机器阅读理解模型大都采用图 1 所示的 4 层架构.

(1) 嵌入层:通过字符、词、上下文和特征级别的嵌入方法将段落 *C* 和问题 *Q* 表示为 *d* 维的词向量作为模型的输入;也有一些模型在嵌入层采用注意力机制将问题的词向量信息融入段落的词向量之中作为段落的最终输入[46];

(2) 编码层:使用循环或卷积神经网络对段落和问题序列进行编码,用以提取内部特征;之后采用注意力机制生成问题感知的段落表示或段落感知的问题表示[47];

(3) 交互层:通过自注意力机制捕捉融合了问题(段落)信息的段落(问题)单词之间的信息;最后通过循环或卷积神经网络解码形成最终表示;

(4) 输出层:根据最终任务(数据集)类型的不同,输出层将会有不同的表示方式.

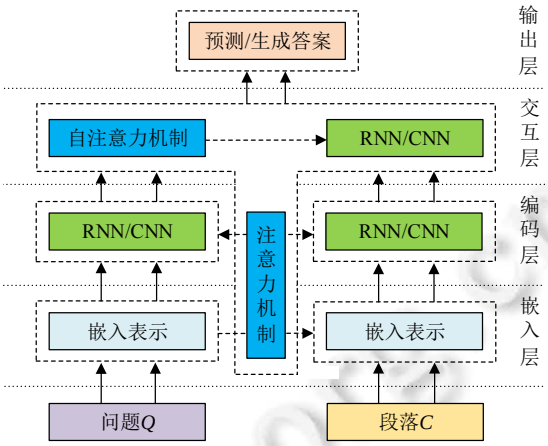


Fig.1 Common four-layer framework of neural reading comprehension

图 1 常用的神经阅读理解 4 层架构

研究者们根据不同类型 MRC 数据集的特点,不断尝试并改进每一层架构中使用的方法,以期在各种数据集中获得最好的性能.本节剩余部分将会具体阐述每一层使用的相关技术及其优缺点,最后介绍最新的 BERT 模型及其优势.

2.1 嵌入层(embedding layer)

基于神经网络的机器阅读理解模型的第 1 个关键步骤就是将单词表示成高维、稠密的实值向量.在深度学习时代之前,研究者们通常将单词表示成词典的索引,称为 One-Hot 词向量表示:每一个单词都会被表示成一个词典中该单词对应位置为 1 而其他位置为 0 的稀疏的向量,例如,$v_{apple}=[0,0,\ldots,0,0,1,0,\ldots,0]^T$,$v_{banana}=[0,1,\ldots,0,0,0,0,\ldots,0]^T$,而 One-Hot 词向量表示方法最大的问题在于这种稀疏向量无法体现任何单词之间的语义相似度信息,因为对于任何单词 a 与 b,$\cos(v_a,v_b)=0$.而高维映射的词嵌入方法可以有效解决上述方法带来的问题,即语义相似的单词可以在几何空间中被编码成距离相近的向量,例如,$\cos(v_{apple},v_{banana})<\cos(v_{apple},v_{car})$.除此之外,研究者们还发现:将嵌入级别细粒度化至字符级别、将静态单词向量变成上下文相关的动态向量或将单词本身的特征一起嵌入到词向量中,都会在一定程度上提高模型的性能.

2.1.1 字符嵌入(character embedding)

字符嵌入用来获取一个单词在字符级别的向量表示,采用 char-level 的词向量能够在一定程度上缓解文本中出现未登录词(out-of-vocabulary,简称 OOV)的问题.Seo 等人[47]首先在 MRC 领域采用了 char-level 嵌入,他们基于 Kim 等人的思想[48],对段落和问题的单词采用一维卷积神经网络(1D-CNN)来获取每个单词字符级别的向量,然后将 char-level 嵌入和词嵌入进行拼接,作为模型的输入.

2.1.2 词嵌入(word embedding)

词向量能够基于单词的分布式假设[49],从大规模无标签的文本语料库中学习获得.在机器阅读理解任务中,使用最多的是 Word2Vec、GloVe 以及 Fasttext 这 3 种单词级别的嵌入模型,下面本文对其进行简单归纳.

(1) Word2Vec 词向量

Word2Vec 词向量[50]本质上是利用一个浅层的神经网络模型在大规模数据集上进行训练得到的结果,该模型具体可以分为连续词袋模型(continuous bag-of-word,简称 CBoW)以及跳跃元语法模型(skip-gram)两种:前者是从上下文对目标单词的预测中学习词向量;后者则相反,是从目标单词对其上下文的预测中学习词向量.除基

本模型以外,Mikolov 等人同时采用了层次 Softmax(hierarchical softmax)和负采样(negative sampling)两种优化算法[51],用来降低模型计算复杂度,缩短训练时间.经由 Word2Vec 模型得到的词向量可以衡量词与词之间的相似程度,例如 *king−man+woman=queen*.

(2) GloVe 词向量

由于 Word2Vec 中常用的 Skip-gram 模型是在独立的局部上下文窗口中训练的,因此没有很好地利用大规模语料库的统计学信息.为了填补 Word2Vec 的局限性,Pennington 等人[52]提出了一种结合局部上下文窗口和全局矩阵分解方法的全局对数双线性回归模型(global log-bilinaer regression model),即 GloVe 模型,该模型不是在整个稀疏矩阵或大型语料库中独立下文窗口上进行训练,而是通过词与词的共现矩阵(co-occurrence)中非零元素的统计信息训练获得.特别地,GloVe 学习的是词向量的共现概率比值,而不是词本身的出现概率.例如,针对热力学领域的词汇 w_i=*ice*,w_j=*steam*,通过研究它们与各种词 w_k 之间的共现概率比值,就可以验证其间的关系.这里,我们将 P_{ij} 视为 w_i 出现在包含 w_j 的上下文中的概率,对于 w_k=*solid* 这一与 w_i 关联度大但与 w_j 关联度小的词,我们期望 P_{ik}/P_{jk} 的值尽可能地大;反之,对于 w_k=*gas* 这一与 w_i 关联度小但与 w_j 关联度大的词,我们期望 P_{ik}/P_{jk} 的值尽可能地小;对于 w_k=*water*/*fashion* 这些与两者都相关或不相关的词,我们期望 P_{ik}/P_{jk} 的值尽可能接近 1.

(3) Fasttext 词向量

Fasttext 词向量[53]本质上是 Fasttext 快速文本分类算法的副产物,该模型的提出,旨在解决 Word2Vec 和 GloVe 模型忽略单词内部结构,从而导致词的形态学特征缺失的问题.Bojanowski 等人通过在 Skip-gram 模型上加入子词信息,实现了单词形态学特征的捕捉,同时解决了文本中出现 OOV 的问题.他们将每一个单词都视为字符袋(bag-of-character)*n*-gram 模型,通过加入边界符号〈和〉来表示词袋的起始,例如:当 *n*=3 时,单词 *apple* 将会被表示为 3-grams:〈*ap*,*app*,*ppl*,*ple*,*le*〉以及一个特殊的序列〈*apple*〉.这种做法可以让模型学习到各类子词特征,例如词根词缀等特征,从而使词向量的表达更加细粒度.

针对究竟上述哪一种词嵌入方法的效果最好这一问题,目前学术界主要将测评方法分为内部测评(intrinsic evaluation)和外部测评(extrinsic evaluation)两种,研究者们通过实验得到的结论也各不相同,较为著名的工作有文献[54−56],由于词嵌入不是本文综述重点,因此这里不再详细介绍.除了 MRC 领域,词嵌入在其他诸如情感分析[57]、主题分类[58]等领域也起到非常重要的作用,因此在构建模型时,根据数据集的特点选择合适的词嵌入方法,是相当重要的一步,可以在一定程度上提升模型的性能.

2.1.3 上下文嵌入(contextualized embedding)

除了词嵌入以外,研究者们还发现:将词的表征扩展到上下文级别,将每个单词表示为整个输入句子的函数映射,即根据当前的句子来体现一个词在特定上下文的语境里面该词的语义表示,使其动态地蕴含句子中上下文单词的特征,从而提高模型的性能.目前较为流行的用于上下文级别嵌入的模型有 CoVe、ELMo 以及 BERT 等预训练模型.

(1) CoVe 上下文向量

虽然使用上述词嵌入方法将单词表示为 *d* 维的词向量可以有效提高包括机器阅读理解在内的多种下游任务的模型性能,但 McCann 等人[59]发现:一个单词除了和与之有着语义相似性的单词在向量空间中距离较近以外,还和出现这个单词的句子中的上下文单词也有着一定的关联.他们认为,一个句子中的每一个单词应该共享其上下文中其他单词的表征能力,这样可以进一步提升模型性能.因此,McCann 等人提出了一种 MT-LSTM 模型,该模型将机器翻译(machine translation,简称 MT)中对句子的编码解码思想运用于词向量,并将模型的输出称为上下文向量(context vector,简称 CoVe),即 *CoVe*(*w*)=*MT-LSTM*(*GloVe*(*w*)).他们通过实验表明:将上下文向量与词向量拼接得到新的词嵌入表征,即 $\tilde{w}=[GloVe(w);CoVe(w)]$ 作为模型的输入,能进一步提高模型性能.例如在 SQuAD 数据集上,与未使用 *CoVe* 向量相比提高了 0.5%~3.9%的 *F*1 值.

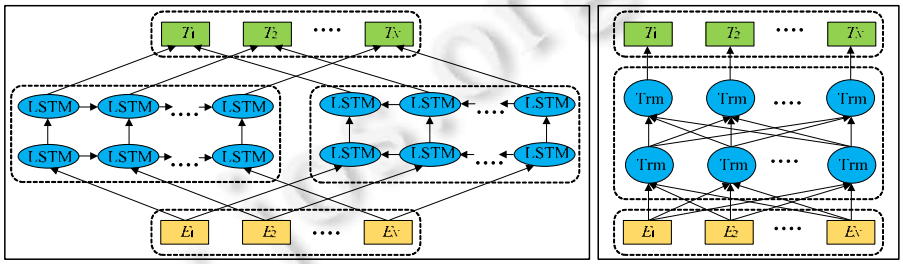
(2) ELMo 上下文向量

Peters 等人[60]认为,一个好的词嵌入应该包括两个部分:一是包含诸如语法和语义等复杂特征;二是能够识别这个单词在不同上下文中的不同使用意义,即一词多义的区分.除此之外,Peters 等人还认为,词表征应该结合

模型的所有内部状态,因为他们发现:高层次的 LSTM 倾向于捕捉上下文相关的信息,而低层次的 LSTM 倾向于捕捉语法相关的信息.因此,不同于 CoVe 模型,Peters 等人采用了耦合双向 LSTM 语言模型(biLM)来生成预训练的上下文词向量(如图 2(a)所示).这种模型的特点是每一个单词的表征都是整个输入句子的函数,同时其包含了高层次和低层次的信息.最后,Peters 等人通过实验表明:在 SQuAD 数据集上,使用结合了 ELMo 和 GloVe 的词向量给下游任务带来的性能比单独使用 GloVe 词向量提升了 4.7%.

(3) BERT 上下文向量

在 ELMo 模型提出不久后,Devlin 等人[14]就发现了 ELMo 存在的两个潜在问题:一是 biLM 模型并不是完全双向的,即句子从左到右和从右到左的 LSTM 过程并不是同时进行的;二是传统语言模型的数学原理决定了它的单向性,对于完全双向的 Bi-LSTM 来说,只要层数增加,就会存在预测单词"自己看见自己"的问题.通过建立双向 Transformer 架构,加以采用遮蔽语言模型以及连续句子预测来解决上述问题,Devlin 等人提出了具有划时代意义的 BERT 预训练模型(如图 2(b)所示).该模型进一步增强了词向量模型的泛化能力,充分描述了字符级别、单词级别和句子级别的关系特征.BERT 预训练模型的提出,使得机器阅读理解进入了一个新的阶段.关于 BERT 预模型及其优势将会在第 2.5 节详细加以介绍.



(a) ELMo模型架构 (b) BERT模型架构

Fig.2 Pre-training model architectures of ELMo and BERT

图 2 ELMo 和 BERT 的预训练模型架构

2.1.4 特征嵌入(feature embedding)

特征嵌入本质上就是将单词在句子中的一些固有特征表示成低维度的向量,包括单词的位置特征(position)[14]、词性特征(POS)、命名实体识别特征(NER)、完全匹配特征(em)以及标准化术语频率(NTF)[61]等等,一般会通过拼接的方式将其与字符嵌入、词嵌入、上下文嵌入一起作为最后的词表征,例如,将 GloVe 词向量 $GloVe(w)$、BERT 上下文向量 $BERT_w$ 以及上述特征向量 f_w 拼接作为输入,则可表示为

$$\tilde{w}=[GloVe(w);CoVe(w);BERT_w;f_w].$$

特征嵌入很好地保留了单词在原始句子中的一些特性,例如位置特征可以弥补词袋模型无顺序的缺点等等,这对于下游任务中模型性能的提升有着一定的辅助作用.

2.2 编码层(encoder layer)

编码层的目的是将已经表示为词向量的 Tokens(词的唯一标记单位)通过一些复合函数进一步学习其内在的特征与关联信息,机器阅读理解中常用循环神经网络(recurrent neural networks,简称 RNNs)及其变体对问题和段落进行建模编码,也有一些模型使用卷积神经网络(convolutional neural network,简称 CNN)进行特征提取,例如 QANet[62].下面,本小节将通过形式化的方式简单回顾两种神经网络在 MRC 领域的应用.

2.2.1 循环神经网络(RNN/LSTM/GRU)

循环神经网络[63]是神经网络的一种,主要用来处理可变长度的序列数据.不同于前馈神经网络,RNNs 可以利用其内部的记忆来处理任意时序的输入序列,这使得它更容易处理机器阅读理解数据集中的问题和段落序列.具体地,我们假设问题 Q 和段落 C 是一个 Tokens 序列(即已经通过预训练的词嵌入或上下文嵌入模型表示成词向量)$x=x_1,x_2,\ldots,x_n\in\mathbb{R}^d$,传统的 RNN 模型通过隐状态 $h_t=f(h_{t-1},x_t;\Theta)\in\mathbb{R}^h$ 可以表达出 $x_{1:t}$ 的上下文信息,如公

式(2)所示,其中,$W^{hh}\in\mathbb{R}^{h\times h}$,$W^{hx}\in\mathbb{R}^{h\times d}$,$b\in\mathbb{R}^h$ 是需要学习的参数:

$$h_t=\tanh(W^{hh}h_{t-1}+W^{hx}x_t+b) \tag{10}$$

为了优化传统 RNN 模型的性能(例如解决 RNN 出现的梯度消失问题),研究者们提出了许多 RNN 的变体,其中比较著名且常用的变体有长短期记忆网络(long short-term memory,简称 LSTM)[64]和门控循环单元(gated recurrent unit,简称 GRU)[65].在 MRC 甚至整个 NLP 领域的应用中,LSTM 是最具有竞争性且使用最为广泛的 RNN 变体,因此本文简单回顾 LSTM 的工作原理.

LSTM 模型通过更新每一时刻的细胞状态来实现序列编码,细胞单元的输入为当前时刻的输入 x_t、上一时刻的隐状态 h_{t-1} 和上一时刻的细胞状态 c_{t-1},输出为当前时刻的隐状态 h_t 和当前时刻的细胞状态 c_t.细胞单元主要由输入门 i_t、遗忘门 f_i 以及输出门 o_t 组成,通过一系列公式(公式(3)~公式(8))实现细胞状态的更新,其中,$W^{ih},W^{fh},W^{oh},W^{gh}\in\mathbb{R}^{h\times h}$,$W^{ix},W^{fx},W^{ox},W^{gx}\in\mathbb{R}^{h\times d}$,$b^i,b^f,b^o,b^g\in\mathbb{R}^h$ 为需要学习的参数:

$$i_t=\sigma(W^{ih}h_{t-1}+W^{ix}x_t+b^i) \tag{11}$$

$$f_t=\sigma(W^{fh}h_{t-1}+W^{fx}x_t+b^f) \tag{12}$$

$$o_t=\sigma(W^{oh}h_{t-1}+W^{ox}x_t+b^o) \tag{13}$$

$$g_t=\tanh(W^{gh}h_{t-1}+W^{gx}x_t+b^g) \tag{14}$$

$$c_t=f_t\odot c_{t-1}+i_t\odot g_t \tag{15}$$

$$h_t=o_t\odot\tanh(c_t) \tag{16}$$

在 MRC 任务中,研究者们常用双 LSTM(BiLSTM)模型对问题 Q 和段落 C 进行编码,如公式(17)、公式(18)所示:

$$\overrightarrow{h_t}=f(\overrightarrow{h_{t-1}},x_t;\overrightarrow{\Theta}),t=1,...,n \tag{17}$$

$$\overleftarrow{h_t}=f(\overleftarrow{h_{t-1}},x_t;\overleftarrow{\Theta}),t=n,...,1 \tag{18}$$

这种 BiLSTM 分为前向传播(从左到右)和反向传播(从右到左),我们将双向模型得到的隐状态结果进行拼接,得到 $h_t=[\overrightarrow{h_t};\overleftarrow{h_t}]\in\mathbb{R}^{2d}$.这种表示方法可以有效地将上下文的左右信息进行编码,从而成为神经阅读理解模型中编码层常用的方法.

2.2.2 卷积神经网络(CNN)

虽然 RNN 模型是机器阅读理解任务中编码层主要采用的方法,但 Yu 等人[62]认为:使用 RNN 模型对问题 Q 和段落 C 进行编码时,会导致模型训练和推理的速度变得非常缓慢,这使得模型无法应用于更大的数据集,同时无法应用于实时系统中.因此,Yu 等人摒弃了 RNN 模型,引入速度较快的卷积神经网络[66]对文本序列进行编码.他们利用 CNN 模型善于提取文本局部特征的优点,同时采用自注意力机制[39]来弥补 CNN 模型无法对句子中单词的全局交互信息进行捕捉的劣势.实验结果表明:在不失准确率的情况下,Yu 等人提出的模型在训练时间上较先前的模型快了 3~13 倍[47,67].但总体而言,CNN 模型在 MRC 任务中仍使用较少,因此有关 CNN 模型对文本序列进行编码的具体细节方面本文不再叙述,可参考文献[62,68].

除了上述两种常见的编码模型外,最近的研究还发现,使用基于自注意力机制的 Transformer 架构对序列进行编码可以获得更快的速度以及更好的效果.我们将在第 2.3 节详细介绍注意力机制有关的模型架构.

2.3 交互层(interaction layer)

交互层是整个神经阅读理解模型的核心部分,它的主要作用是负责段落与问题之间的逐字交互,从而获取段落(问题)中的单词针对于问题(段落)中的单词的加权状态,进一步融合已经被编码的段落与问题序列.

2.3.1 注意力机制(attention mechanism)

交互层主要采用注意力机制,在自然语言处理领域,该机制最早由 Sutskever 等人[69]在 2014 年应用于 Sequence-to-Sequence 模型(Seq2Seq);随后,Luong 等人[70]和 Bahdanau 等人[71]将其应用于机器翻译领域并获得极大的成功;这之后,注意力机制被广泛应用于各种 NLP 任务中[72−74],包括机器阅读理解任务.

如图 3 所示:在 MRC 任务中,一般使用注意力机制来融合“段落和问题”的信息.图 3 左为以段落-问题为例

的注意力矩阵表示,图 3 右为注意力矩阵中每一行注意力值(即段落中每一单词对问题的注意力)的详细计算方法,具体地,可以分为以下 3 步.

(1) 将段落 C 中的单词 C_j,j=1,…,m 和问题 Q 中的每一个单词 Q_1,…,Q_i,…,Q_n 进行相似度计算,得到权重 S_{j1},…,S_{ji},…,S_{jn},其中,相似度函数可以有多种选择,常用的有点积 dot、双线性映射 bilinear 以及多层感知机 MLP:

$$S_{ji}=\begin{cases} f_{dot}(C_j,Q_i)=C_j^TQ_i \\ f_{bilinear}(C_j,Q_i)=C_j^TWQ_i \\ f_{MLP}(C_j,Q_i)=V^T\tanh(W^CC_j+W^QQ_i) \end{cases} \tag{19}$$

(2) 使用 Softmax 函数对权重进行归一化处理,得到α_{j1},…,α_{ji},…,α_{jn}:

$$\alpha_{ji}=\mathrm{softmax}(S_{ji})=\frac{\exp(S_{ji})}{\sum_{i=1}^{n}\exp(S_{ji})} \tag{20}$$

(3) 将归一化后的权重和相应的问题 Q 中的单词 Q_i 进行加权求和,得到序列 $\hat{C}_1$,...,$\hat{C}_j$,...,$\hat{C}_m$,即问题-感知的段落表示:

$$\hat{C}_j=Attention(C_j,Q)=\sum_{i=1}^{n}\alpha_{ji}Q_i \tag{21}$$

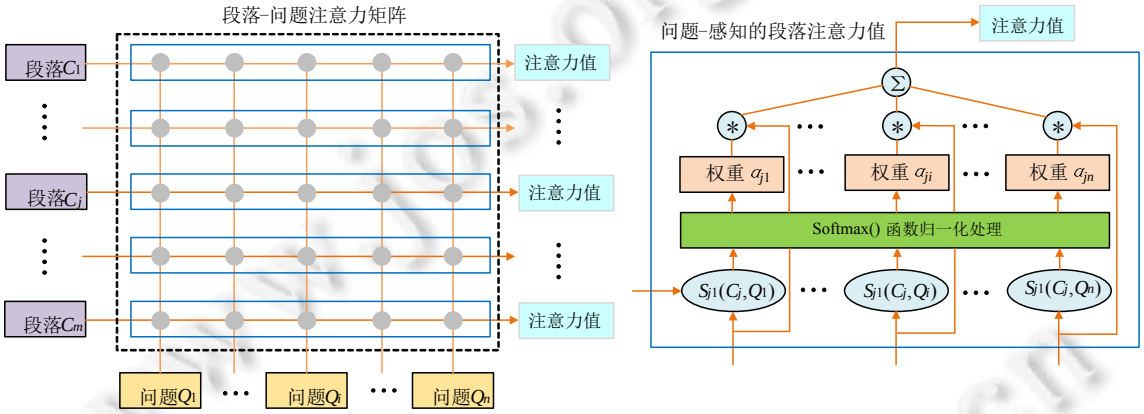


Fig.3 Attention mechanism in MRC tasks (C2Q)

图 3 MRC 任务中的注意力机制(C2Q)

当段落和问题序列通过注意力机制后,神经阅读理解模型就能学习到两者之间单词级别的权重状态,这大大提高了最后答案预测或生成的准确率.

2.3.2 自注意力机制(self-attention mechanism)

当注意力机制在 NLP 领域取得巨大成功后,Vaswani 等人[67]猜测:既然注意力机制能让模型学习到两个序列的权重状态,那是否可以将一个序列自己与自己进行注意力学习,进而学习句子内部的词依赖关系,捕获句子的内部结构,以此来替代 RNN 模型对序列进行编码.于是,Vaswani 等人提出了自注意力机制(self-attention mechanism),并基于多头自注意力机制(multi-head self-attention)提出了 Transformer 模型架构,该架构用注意力机制完全替代了 RNN 模型,使得整个模型的参数大大减少并且弥补了 RNN 模型并行性差的缺点.事实证明,Transformer 模型是将来 MRC 任务甚至整个 NLP 领域的趋势,我们将在第 2.5 节中继续对此加以讨论.

首次将自注意力机制运用于机器阅读理解任务的是 Wang 等人[75]在 2017 年提出的 R-Net 模型,他们认为:段落中的单词与单词之间能够通过注意力机制实现单词匹配(aligned),进而聚合来自段落不同部分的信息.Wang 等人将段落的自注意力机制形式化表示成如下公式,其中,S_{jt} 表示段落中第 j 个单词对整个段落 C_1,…,C_t,…,C_m 的注意力值.经过自注意力机制计算后,得到融合了段落自身加权表示的自感知的段落表示 $\hat{C}_1$,...,$\hat{C}_j$,...,$\hat{C}_m$:

$$S_{jt} = f_{MLP}(C_j, C_t) = V^T \tanh(W^C C_j + W^{\tilde{C}} C_t) \tag{22}$$

$$\alpha_{jt} = \text{softmax}(S_{jt}) = \frac{\exp(S_{jt})}{\sum_{t=1}^{m} \exp(S_{jt})} \tag{23}$$

$$\hat{C}_j = Attention(C_j, C) = \sum_{t=1}^{m} \alpha_{jt} C_t \tag{24}$$

最后,将单词与相应的注意力值按顺序进行拼接,作为 BiLSTM 的输入.除此之外,将自注意力机制应用于问题和段落拼接后的向量,不仅可以使模型学习到了问题和段落内部的融合信息,还同时学习到了问题和段落之间的交互信息[76].本文认为:注意力机制的另一个优点是大大增强了模型的可解释性,能够让研究者们清楚地看到单词之间的关联程度.

2.4 输出层(output layer)

输出层主要用来实现答案的预测与生成,根据具体任务来定义需要预测的参数.

(1) 针对抽取式任务,神经阅读理解模型需要从某一段落中找到一个子片段(span or sub-phrase)来回答对应问题,这一片段将会以在段落中的首尾索引的形式表示,因此,模型需要通过获取起始和结束位置的概率分布来找到对应的索引[77].具体来讲,模型需要通过公式(25)、公式(26)生成答案的起止索引,其中,$W^{(start)}$和 $W^{(end)}$是需要训练的参数:

$$P^{(start)}(i) = \frac{\exp(P_i W^{(start)} Q)}{\sum_u \exp(P_u W^{(start)} Q)} \tag{25}$$

$$P^{(end)}(i) = \frac{\exp(P_i W^{(end)} Q)}{\sum_u \exp(P_u W^{(end)} Q)} \tag{26}$$

(2) 针对完形填空任务,神经阅读理解模型需要从若干个答案选项中选择一项填入问句的空缺部分,因此,模型首先需要计算出段落针对问题的注意力值,然后通过获取选项集合中候选答案的概率预测出正确答案,如公式(27)~公式(29)所示,其中,W、$W_a^{(answer)}$ 和 $W_{\tilde{a}}^{(answer)}$ 是需要训练的参数,$\mathcal{A}$是候选答案集合:

$$\alpha_i = \frac{\exp(P_i W Q)}{\sum_t \exp(P_t W Q)} \tag{27}$$

$$u = \sum_i \alpha_i P_i \tag{28}$$

$$P(Y = a \mid p, q) = \frac{\exp(W_a^{(answer)} u)}{\sum_{\tilde{a} \in \mathrm{A}} \exp(W_{\tilde{a}}^{(answer)} u)} \tag{29}$$

(3) 针对多项选择任务,神经阅读理解模型需要从 k 个选项中选出正确答案,因此,模型可以先通过 BiLSTM 将每一个答案进行编码得到 a_i,之后与 u 进行相似度对比,预测出正确的答案,如公式(27)、公式(28)、公式(30)所示,其中,W 和 W^{answer} 是需要训练的参数:

$$P(Y = i \mid p, q) = \frac{\exp(a_i W^{(answer)} u)}{\sum_{t=1,\ldots,k} \exp(a_t W^{(answer)} u)} \tag{30}$$

(4) 针对生成式任务,由于答案的形式是自由的(free-form),可能在段落中能找到,也可能无法直接找到而需要模型生成,因此,模型的输出不是固定形式的,有可能依赖预测起止位置的概率(与抽取式相同),也有可能需要模型产生自由形式的答案(类似于 Seq2Seq).

(5) 针对会话类和多跳推理任务,由于只是推理过程与抽取式不同,其输出形式基本上与抽取式任务相同,有些数据集还会预测“是/否”、不可回答[27]以及“能否成为支持证据”[34]的概率.

(6) 针对开放域的阅读理解,由于模型首先需要根据给定问题,从例如 Wikipedia 上检索多个相关文档(包含多个段落),再从中阅读并给出答案[78]:

$$f(a|q) = retrieve(d_1, d_2, \ldots, d_n|q) + read(a_1, a_2, \ldots, a_n|q) + rank \tag{31}$$

2.5 BERT预训练模型

在谷歌公司提出 BERT 模型之前,就已有学者考虑使用高质量的预训练模型来提升后续任务的性能,例如 ELMo 和 GPT 模型[79].但是由于 ELMo 仍然采用 LSTM 作为编码器(如图 2(a)所示),导致网络训练速度较慢,尤其是需要用到海量未标记语料来训练模型时;而 GPT 模型虽然采用了可以并行处理序列的 Transformer 架构,大大提升了训练效率,但是由于其为单向编码结构,导致序列中每一个 Token 只能通过自注意力机制注意到先前的 Token.因此,Devlin 等人[14]提出了将上述两个模型结合的 BERT 预训练模型.他们采用遮蔽语言模型(masked LM)的方法来解决完全双向编码机制隐含的"自己看见自己"的问题,同时,采用连续句子预测(next sentence prediction)的方法将模型的适用范围从单词级别扩展到句子级别,这两项创新也使得 BERT 预训练模型能够充分利用并挖掘海量的语料库信息[80],从而大幅度提升包括机器阅读理解在内的 11 项下游任务的性能.接下来,我们将详细阐述 BERT 预训练模型所用到的 3 类技术并分析其优势.

2.5.1 Transformer 架构

Transformer 架构[67]由 6 个相同的编码-解码模块组成,其中,每个编码模块包括了自注意力模块(self-attention)和前向神经网络模块(feed forward neural network),每个解码模块包括了自注意力模块、编码-解码注意力模块(encoder-decoder attention)和前向神经网络模块,而其中,自注意力模块采用了多头注意力机制(multi-headed attention),即采用 h 个不同的自注意力进行集成,多次并行地通过缩放点积(scaled dot-product)来计算注意力值,如公式(32)、公式(33)与如图 4 所示,其中,d_k 为 K 的维度:

$$Attention(Q,K,V)=\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{32}$$

$$MultiHead(Q,K,V)=Concat(head_1,...,head_h)W^O,\text{其中},head_i=Attention(QW_i^Q,KW_i^K,VW_i^V) \tag{33}$$

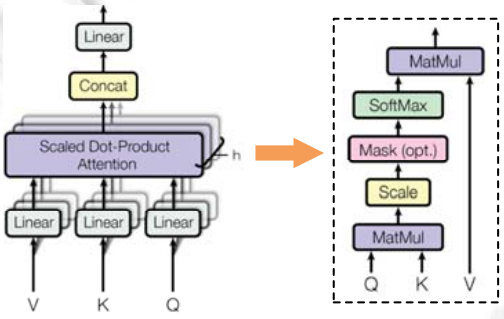


Fig.4 Multi-headed attention and scaled dot-product attention

图 4 多头注意力机制和缩放点积注意力

Transformer架构最早是为了解决序列转换或神经机器翻译问题[81]而设计的架构,在Transformer架构之前,NLP 领域中多数基于神经网络的方法是依赖于 RNN 或其变体对句子进行序列编码.尽管 RNN 结构在序列建模方面非常强大,但其序列性的特点也限制了模型的训练速度.而 Transformer 架构摒弃了 RNN 结构,采用了自注意力机制作为编码模块,这个变化为模型带来了两个优势:(1) 由于 Transformer 架构不需要循环处理单词序列,因此训练速度比 RNN 结构快很多;(2) 采用自注意力机制不仅能够让句子中单词与单词之间产生关联,还能通过权重系数计算出哪些单词之间的关联性更大,提高了模型的可解释性.

Transformer 架构另一个里程碑式的创新之处在于:为基于海量未标记语料训练的预训练模型的构建提供了支持,进而使研究者们只需在对应的下游任务中微调预训练模型就能达到较好的效果.其中最具代表性的应用就是通过基于 Transformer 架构的预训练模型来提升词表达能力:通过自注意力机制,可以在一定程度上反映出一句话中不同单词之间的关联性以及重要程度,再通过训练来调整每个词的重要性(即权重系数),由此来获得每个单词的表达.由于这个表达不仅仅蕴含了该单词本身,还动态地蕴含了句子中其他单词的关系,因此相比于普通的词向量,通过上述预训练模型得到的上下文词表达更为全面.

2.5.2 遮蔽语言模型

Devlin 等人认为:虽然完全双向模型的性能必定比单向模型(GPT)或不完全双向模型(ELMo)更好,而一旦采用了完全双向模型,随着网络层数的增加,势必会出现“自己看见自己”的问题,这就使模型失去了意义(我们的目标是通过训练学习到词与词之间蕴含的未知关系).针对上述问题,Devlin 等人受完形填空任务[82]的启发,提出了采用 Masked LM 的方式来训练模型.他们将输入 Tokens 中的 15%进行随机遮蔽处理,用[MASK]标记替代;进一步地,他们为了解决预训练模型中若完全使用[MASK]标记则会导致后续任务不能很好地进行模型微调的问题(因为后续任务微调中并不会出现[MASK]这一标记),进行了如图 5 所示的改变.

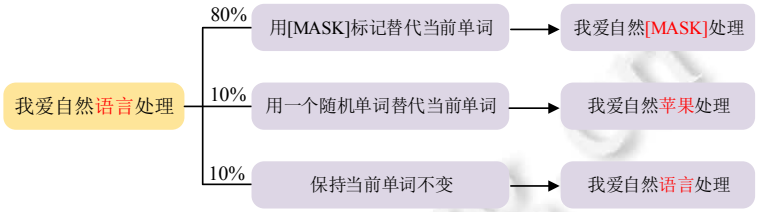


Fig.5 Masked LM in BERT

图 5 BERT 中的遮蔽语言模型

通过上述变化,使得 Transformer 架构不知道哪个单词需要被预测,哪个单词已经被替换.因此,BERT 不仅解决了完全双向模型“自己看见自己”的问题,还“被迫”地保证了每一个输入 Token 都能保持分布式的上下文表征状态.

2.5.3 连续句子预测

连续句子预测任务主要是为了让模型能够学习连续的文本片段之间的关系,以加入句子级别的表征能力.具体地说,对于每次从训练集中选取的两个连续句子 *A* 和 *B*,Devlin 等人将句子 *B* 做以下处理:在训练时,输入模型的第 2 个句子 *B* 会以 50%的概率从全部文本中随机选取替换,剩下 50%的概率选取第 1 个句子 *A* 的后续的文本,即保持原句不变.例如:

(1) 输入=[CLS]男子去[MASK]商店[SEP]他买了一升[MASK]牛奶[SEP],标签=IsNext;

(2) 输入=[CLS]男子去[MASK]商店[SEP]我爱自然[MASK]处理[SEP],标签=NotNext.

使用连续句子预测来训练 BERT 模型,可以很大程度上提高模型在类似机器阅读理解(MRC)以及自然语言推理(NLI)这些需要句间理解能力的任务上的表现性能.

综上所述,本文认为:BERT 模型的提出,对于机器阅读理解任务来说,除了为建立更高性能的模型提供新的思路以外,更是证明了一个好的预训练模型在 MRC 任务中的重要性.因此,未来 MRC 模型的建立可以从以下两方面展开.

(1) 加入上下文嵌入作为表征.将 BERT 预训练模型得到的词的上下文表征结合静态词嵌入方法,共同作为嵌入层的结果,以此来提高模型性能,这也是目前 BERT 模型最广泛的使用方法;

(2) 优化 BERT 模型.由于单一的 BERT 模型对于需要复杂推理的任务处理起来相对薄弱,因此我们可以在 BERT 模型的基础上进行结构优化,提高模型对问题与段落内在关系的推理能力,由此来处理更为复杂、推理难度更大的 MRC 数据集.

3 数据集及神经网络模型相关研究

机器阅读理解的研究之所以能够在近期发展如此快速且成功,主要有以下两个原因:大规模的阅读理解数据集的发布以及端到端神经阅读理解模型的构建.两者共同推进着 MRC 系统的发展,一方面,大规模 MRC 数据集的发布能够很好地适应神经阅读理解模型的训练,因此激励研究者对模型进行不断的创新;另一方面,模型性能的不断提升也促进更具有挑战性、更符合人类自然语言习惯的数据集的建立.图 6 所示为神经网络时代下的

MRC 主要数据集以及模型发展时间线(蓝色为数据集,绿色为模型),我们可以看到:在最近 3 年的时间里,MRC 领域的发展就已经取得了惊人的成绩(所有数据集和模型都已经以论文形式发表在国际会议或 arXiv 上,截至日期为 2019.7.18).

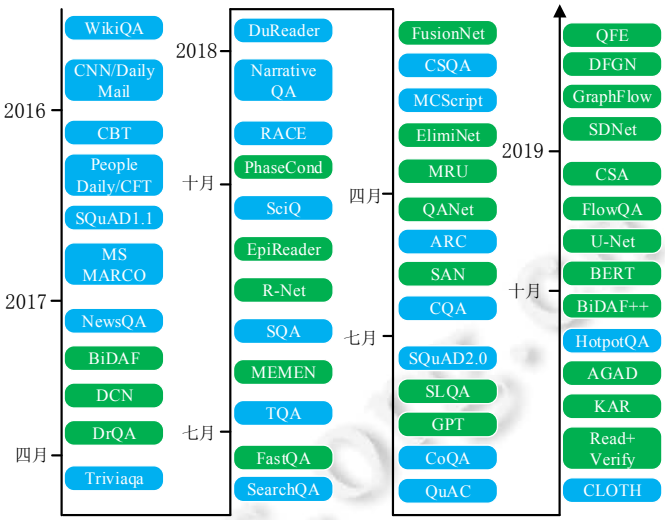


Fig.6 Recent development of datasets and models in neural reading comprehension

图 6 近年来神经阅读理解数据集和模型的发展

3.1 数据集分类与分析

从 2015 年至今,国内外已经公布了许多专门用于机器阅读理解的数据集,本文选取最具有代表性的 24 个中英文数据集进行对比分析,各个数据集的属性见表 2.

Table 2 Comparison of some properties of main MRC datasets

表 2 机器阅读理解数据集的属性对比

名称	类别	语言	#问题	#段落	问题来源	段落来源	不可回答	是/否类
CNN/DailyMail[1]	完形填空	英文	1.38M	312K	生成	新闻	–	–
CBT[25]	完形填空	英文	688K	108	生成	少儿图书	–	–
CLOTH[26]	完形填空	英文	99K	7.1K	英语考试	英语考试	–	–
PeopleDaily/CFT[35]	完形填空	中文	100K	28K	生成	新闻/童话	–	–
TQA[23]	多项选择	英文	26.3K	1 076	理科课程	理科课程	–	√
SciQ[20]	多项选择	英文	13.7K	N/A	众包	科学考试	–	–
MCScript[24]	多项选择	英文	14K	2 100	众包	日常语料	–	√
RACE[22]	多项选择	英文	100K	28K	英语考试	英语考试	–	–
ARC[21]	多项选择	英文	7 787	N/A	科学考试	科学考试	√	–
WikiQA[16]	抽取式	英文	3 047	N/A	必应日志	维基百科	–	–
SQuAD1.1[13]	抽取式	英文	108K	536	众包	维基百科	–	–
NewsQA[17]	抽取式	英文	120K	12.7K	众包	新闻	√	–
SearchQA[19]	抽取式	英文	140K	6.9M	网页	网页	–	–
TriviaQA[15]	抽取式	英文	95.9K	663K	网页	维基/网页	–	–
SQuAD2.0[18]	抽取式	英文	151K	505	众包	维基百科	√	–
SQA[30]	多轮对话	英文	17.6K	6 066	众包	维基百科	–	√
CSQA[29]	多轮对话	英文	1.6M	200K	众包/生成	知识图谱	√	√
CQA[31]	多轮对话	英文	34.7K	N/A	众包/生成	Web 知识库	–	–
CoQA[27]	多轮对话	英文	127K	8K	众包	多领域	√	√
QuAC[28]	多轮对话	英文	98.4K	13.6K	众包	维基百科	√	√
DuReader[36]	生成式	中文	200K	1M	百度日志	网页	√	√
MS MARCO[33]	生成式	英文	100K	200K	必应日志	网页	√	√
NarrativeQA[32]	生成式	英文	46K	1.5K	众包	书籍/电影	–	–
HotpotQA[34]	多跳推理	英文	112K	N/A	众包	维基百科	–	√

接下来,本文将分别从类别、问题来源、段落来源以及数据集难度这 4 个维度出发,对 MRC 数据集的发展进行归纳分析.

3.1.1 数据集类别分析

目前,国内外学者构建的机器阅读理解数据集大致可以被分为六大类,即完形填空、多项选择、抽取式、多轮对话、生成式以及多跳推理,每一类数据集的任务定义已在第 1.2 节中给出描述,本小节不再重复.除了上述六大类数据集以外,本节还阐述了面向开放域 Open-Domain 的机器阅读理解.接下来,本小节将按类别归纳每个数据集的特点与不足.

(1) 完型填空

Hermann 等人[1]在 2015 年提出了第一个大规模的 MRC 数据集 CNN/Daily Mail,该数据集的提出,成为了机器阅读理解进入神经网络时代的标志(尽管 WikiQA[16]的提出更早一些,但其只有 3 047 条问题数据,因此并不能真正意义上称为大规模),可想而知该数据集在 MRC 领域的地位.Hermann 等人认为:一个完备的有监督的 MRC 数据集应该只能依赖(C,Q,A)三元组而不再依赖额外方法(例如语法信息、共现统计信息等)获取的知识,即仅通过段落和问题能够直接找到答案.因此,Hermann 等人通过实体替代和置换的生成方法从 312K 条新闻数据中构建了 1.38M 条填空语句,同时采用匿名标记来表示被替代和置换的实体,以阻止外部知识的干扰.Hill 等人[25]认为:CNN/Daily Mail 使用匿名标记进行实体替代和置换这一方法虽然可以保证数据集问答对的纯粹性,即无需依赖外部知识辅助,但标记化的答案会降低数据集所需要的推理能力,不符合人类的真实问答行为.因此,他们提出了 CBT 数据集.该数据集以叙事结构清晰的 108 本儿童读物为段落源,分别通过文本段落末尾句子中取出一个命名实体和随机抽样同词性词的方法来生成 688K 条问题以及对应的 10 个候选答案.CBT 数据集至今为止仍然是用于测试模型对完形填空类问答性能的非常重要的标准数据集之一.之后,Cui 等人[35]结合了 CNN/Daily Mail 和 CBT 数据集的特点,针对中文语料提出了 People Daily/CFT 数据集,从而填补了中文完形填空数据集的空白.到了近期,Xie 等人[26]发现:利用生成的方法产生问答集会导致问题缺乏目的性,即空白处过于琐碎与段落整体关联性不强,他们认为,完形填空数据集应该是严谨有逻辑的,因此他们收集中国各地高考和中考的英语考试试题,通过抽取、清洗等方法最终获得包含 7 131 篇完形填空段落和 99K 个问题的数据集 CLOTH.本文认为:由于完形填空问题并不符合人们日常对话习惯,因此更适合作为试题或测验.同时,一个高质量的具有挑战性的完形填空数据集应该具备以下两点:一是留白处(问题)不仅需要和该句上下文有关联,还需要与同一段落中其他留白处(问题)有一定的逻辑关系;二是候选答案的特征需要有一定关联,例如词性或语义.

(2) 多项选择

Kembhavi 等人[23]收集了 1 076 个中学理科课程,提出了包含有 26 260 个多项选择问题的多项选择类数据集 TQA.与其他数据集最大的不同之处在于:该数据集不仅包含文本类问题,还包含一部分图像类问题.同时,他们尝试加入“是/否类问题”,这大大提高了数据集的难度.Welbl 等人[20]认为:一个高质量的多项选择数据集的选项应该具有很强的迷惑性(plausible),需要模型具备信息抽取、整合、理解以及常识推理能力才能选出正确答案.因此,他们首先利用一个文本过滤器从四到八年级的科学考试中筛选出符合要求的段落,然后通过 Amazon Mechanical Turk 众包服务平台生成一个大规模多项选择类数据集 SciQ.尽管 SciQ 是一个结构化程度很高的领域数据集,但该数据集中的答案都可以在段落中被找到,因此在挑战性方面没有 TQA 高.与此同时,Lai 等人[22]也针对提高机器的推理能力的需求,提出了基于 28K 篇中国初高中学生英语考试段落的 RACE 数据集.该数据集的问题超过 100K,并且都是由专家提出,以此保证了该数据集的推理难度.与 SciQ 数据集类似,Ostermann 等人[24]也采用众包的方法从日常语料(众包工作者根据主题创作的故事)中获取了包含 14K 个问题的多项选择数据集 MCScript,但该数据集选取了开放域语料,同时包含了 29%的“是/否类问题”,因此相对来说数据集难度有所提高,然而提高程度并不明显.目前,在多项选择类数据集中,最具有挑战性的数据集是 Clark 等人[21]在 2018 年提出的 ARC 数据集.他们认为:现阶段多项选择类数据集之所以难度不大,是因为其答案都能够通过基于检索或词共现的方法获取,他们把该类问题归为简单集(easy set).之后,又加入了无法用上述两种方法获取答案的问题,将其归为挑战集(challenge set).挑战集中的答案不会在段落中出现,而是需要模型结合外部知识进行推理做出

选择.图 7 给出了在 ARC 挑战集上的问题类型分布,可以看出,几乎没有一种类型的问题可以直接通过段落检索而得到正确答案.因此,ARC 挑战集也成为了至今多项选择类数据集中难度最大的数据集之一.截至本文撰写时,ARC 挑战集榜单上的第一名也仅仅达到了 44.71%的准确率,未来仍然有很大的研究空间.

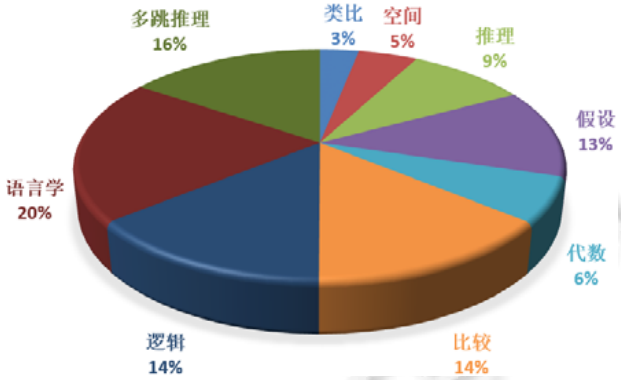


Fig.7 Distribution of question types in ARC challenge set

图 7 ARC 挑战集中问题类型的分布

(3) 抽取式

抽取式 MRC 数据集由于其段落的多领域性、问题的多变性以及答案的自由性,使其成为近 3 年来学者们研究频率最高的数据集.早在 2015 年,Yang 等人[16]就将必应查询语句视为问题 *Q*,将维基百科中包含答案的句子视为答案 *A*,以此构建了抽取式数据集 WikiQA,但该数据集有两个不足之处:一是查询语句存在表达不规范的问题;二是该数据集本质上是句子选择任务,即只需选出包含答案的句子,并不能称为抽取式.随后,Rajpurkar 等人[13]采用众包服务模式构建了第一个真正意义上的大规模抽取式 MRC 数据集 SQuAD1.1(为了区分 Rajpurkar 等人在 2018 年提出的 SQuADRUn 数据集,我们将前者称为 SQuAD1.1,后者称为 SQuAD2.0),正如 Rajpurkar 等人所说,SQuAD1.1 与 CNN/Daily Mail 和 CBT 最大的不同之处在于:前者的问题是通过段落生成的,并且答案包括了非实体词汇以及长片段;而后两者的问题是直接留白于段落,且答案是简单的实体单词.

本文总结了 SQuAD1.1 数据集的两个重要意义:一是相比于完形填空与多项选择类数据集,SQuAD1.1 更加贴近人类自然语言,这使得接下来两年抽取式数据集成为了 MRC 领域的主流数据集;二是该数据集的成功,表明了采用众包服务模式来生成问答集是一种快速且有效的方法,可以有效降低构建高质量数据集的难度,因此越来越多的高质量数据集被提出,直接推动了神经网络模型的发展.

在 SQuAD1.1 之后,研究者们采用各种方法、技巧用以丰富抽取式数据集的形式,增加数据集的难度,例如,Trischler 等人[17]将众包服务模式运用于 CNN/Daily Mail 数据集的源段落,构建了基于新闻数据的抽取式数据集 NewsQA;除此之外,Trischler 等人为了和 SQuAD1.1 数据集有所区分,在答案中加入了一定比例的“*null*”答案用以增加该数据集的难度,NewsQA 也成为了较早通过改变答案类型来增加难度的数据集.Joshi 等人[15]认为:现有的数据集都是在源段落上运用各种技术(生成、截取和众包)获得问答集,但这会导致问答对本身就隐含了段落的信息,潜在地降低了数据集的难度.因此,Joshi 等人利用逆向思维,首先在 quiz-league 网站收集问答对;之后,通过在 Web 和 Wikipedia 中查找段落证据来构建(*Q*,*A*,*C*)三元组,最终构建了包含 95.9K 个问答对、663K 个三元组的数据集 TriviaQA.类似地,Dunn 等人[19]也采用相同的方法在 J-Archive 网站上获取问答对,然后利用 Google 搜索获取段落,构建了包含 6.9M 个段落片段以及 140K 个问答对的抽取式数据集 SearchQA.随后,Rajpurkar 等人[18]发现,现有的包括 SQuAD1.1 在内的抽取式数据集的难度已经无法满足性能快速提升的神经阅读理解模型.其原因主要是答案种类太过于简单,虽然 NewsQA 尝试加入一定比例的“*null*”答案来增加数据集的难度,但 Rajpurkar 等人认为:上述改变太小,不足以“难倒”现有的神经阅读理解模型.因此,他们在提出的 SQuAD2.0 数据集中除了加入“不可回答问题”之外,还加入了众多负面实例,例如在问题中使用反义词、否定词、

实体替换、互斥词等等,这个变化有效提高了数据集的难度,使 Baseline 的 EM 值从 68.0%下降到了 59.2%.

(4) 多轮对话

多轮对话的流程如图 8 所示.我们期望机器能和人类一样完成一个完整的会话过程,这就要求研究人员构建更为完善的、复杂的会话数据集.Iyyer 等人[30]最早将多轮对话阅读理解任务定义为序列问答(sequential QA,简称 SQA),他们通过众包服务模式,将 WikiTableQuestions 数据集中的问题分解成若干相互关联的序列,构建了一个简单的会话数据集;之后,Saha 等人[29]以及 Talmor 等人[31]分别从知识图谱(KG)和网页知识库(Web KB)中提取三元组和 SPARQL 查询语句,构建了 CSQA 数据集和 CQA 数据集.但上述数据集是采用众包平台结合半自动生成的方法来获取问答对,因此,对话在表达的自然性上有所欠缺.到了 2018 年中期,Reddy 等人[27]提出了一个包含了 7 个不同领域的 127K 个问答对的高质量多轮对话数据集 CoQA,CoQA 比起之前的数据集有以下 3 个优势:首先,CoQA 的每一个对话框中的对话流包含了大量共指关系(coreference),例如 he、him、she、it、they 等等,这使得 CoQA 比 SQA、CSQA、CQA 等多轮对话数据集更加符合人类自然语言习惯,同时给神经网络模型在语用推理方面提出了更高的要求;其次,CoQA 还加入了约 1/3 的摘要式答案(abstractive),例如依赖段落中出现的与问题相关的实体的计数或枚举,而不仅仅是从段落中截取一段跨距;最后,“是/否类问题”在 CoQA 中也占了 19.8%,而这个比例在之前的数据集中是远远没有达到的.以上 3 个优点让 CoQA 成为了目前评价模型处理多轮对话类任务的性能的首选数据集.此外,Choi 等人[28]在几乎同一时间提出了与 CoQA 相似的 QuAC 数据集,与 CoQA 不同的是,QuAC 在根据段落获取问答流时,提问者只被允许看到段落的标题以及首段落,由此展开一系列的提问,这使得每个问题蕴含有更多的人为思考以避免过度依赖段落,这样更贴近人类自然语言对话.QuAC 另一个创新点是提出了一个新的评价指标 HEQ(human equivalence score),用来评价模型的输出答案是否与人类一致.该指标又细分为 HEQ-Q 和 HEQ-D:前者用于评价问题的一致性,后者用于评价整轮对话的一致性.

本文认为:在现有的多轮对话数据集中,每一个对话框中的对话流之间规律性太强,例如在 CoQA 中,一个问题一般与前几个问答对关联性较大,而和其相距较远的问答对关联性较弱,很少有后面的问题需要利用最初问答对的情况(即“对话反转”问题),这会为神经阅读理解模型在训练时提供一个潜在的特征.在未来,研究者们可以加入更丰富的“对话反转”问题,以此来构建更具挑战性的多轮对话数据集.

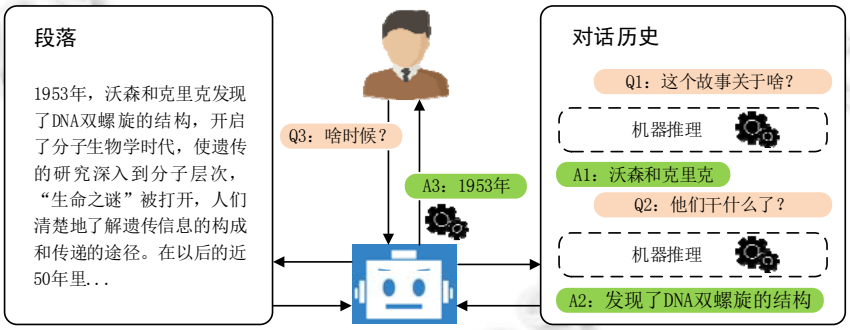


Fig.8 Schematic diagram of multiple-roundconversation

图 8 多轮对话流程示意图

(5) 生成式

生成式数据集的特点在于问题的答案是自由形式的,即答案完全由人工编辑生成.这就要求机器拥有更强的推理能力,能够理解问题与段落中相关句子的逻辑关系并给出正确答案,而不是简单的文本匹配.Nguyen 等人[33]在 2016 年首次提出了生成式数据集 MSMARCO,他们提出该数据集的目的是希望机器能够在若干摘要段落中找到与问题相关的信息并进行推理生成正确的答案,然而,MS MARCO 数据集中的答案虽然是人工编辑的,但它们和段落中的片段之间还原度非常大,因此研究者们发现:若采用和解决 SQuAD1.1 数据集的方式,即预测答案在段落中的位置,就可以获得理想的结果,这说明该数据集仍然缺乏一定的难度.之后,Kocisky 等人[32]基于书本故事和电影脚本提出了著名的生成式数据集 NarrativeQA,该数据集要求模型只有阅读完整的段落或摘

要后才能推理出正确的答案而无法简单地寻找匹配片段.由于这些答案是人工编辑生成的,因此模型必须理解段落的意义才能生成与人工编辑答案最为接近的答案.与 MS MARCO 和 NarrativeQA 类似,He 等人[36]基于百度搜索和百度知道提出了面向中文领域的大规模生成式数据集 DuReader.该数据集中的答案同样是人为编辑生成的,并且还包含了“是否类”和“选项类”的问题;除此之外,DuReader 有时候需要机器概括多个段落或文档才能生成最终答案(即加入了一些多跳推理的问答),加大了该数据集的难度.

(6) 多跳推理

从图 6 我们可以发现,多跳推理类别数据集是最近 1 年才兴起的,其代表为 Yang 等人[34]在 2018 年 9 月 25 日提出的 HotpotQA 数据集.与其他数据集的区别是:(1) HotpotQA 中问题的答案需要模型从多个支持段落中找到相关句子并进行推理得到;(2) 增加了新的评价指标 $F1^{(sup)}$,用以评价模型找到正确支持证据的能力,这一点非常重要,因为现有模型最大的不足就是缺乏可解释性,即知道答案却不知如何知道答案,提供支持证据可以让机器更加接近人类的思维.因此在未来的研究中,多跳推理任务必然会成为国内外机器阅读理解研究的趋势.

(7) 开放域阅读理解

面向开放域的阅读理解(open-domain QA)可以理解为利用例如 Wikipedia、Freebase KB[83]等知识库作为知识源来解决开放领域的阅读理解任务[84,85],而上述六大类别的数据集可以被认为是 Open-Domain QA 的衍生子任务.在早期较为著名的工作有 Microsoft' AskMSR[86]和 IBM's DeepQA[87],其中,AskMSR 本质上是一个依靠数据冗余来实现开放域检索的搜索引擎,因此并不能称为阅读理解模型;DeepQA 同时利用非结构化和结构化的数据来产生候选答案或对证据进行投票,但其仍然属于传统的检索系统.之后,Chen 等人[78]针对开放域阅读理解任务提出了 DrQA.他们采用文档检索结合神经网络的方法构建了 QA 模型,将 Wikipedia 作为知识源,利用 SQuAD 数据集来训练模型,并在 TREC[88]、WebQuestions[89]和 WikiMovies[90]这 3 个数据集上进行测试,实验证明 DrQA 是一种有效的针对开放域问答的阅读理解模型.

3.1.2 问题来源分析

我们可以从表 2 中看出,早期数据集的问题来源主要通过基于规则的生成方法(如 CNN/Daily Mail 和 CBT)或从用户的检索日志中直接抓取(如 WikiQA 和 MS MARCO).而上述两种方法虽然简单但却有着各自的缺点:前者虽然可以产生数量庞大的问题条目,但是由于基于规则的生成方法具有一定的局限性[91],使其与人类的自然语言表达有一定的差距;后者虽然是人类在搜索引擎中的真实查询语句,但问句中含有大量不规范的表达,从而降低了数据集的质量.也有数据集采用直接提取格式化文档中的问题部分作为问题集合的方法(例如 TQA, ARC 和 CLOTH),但这些数据集需要有相对客观可靠的段落来源,例如课程或考试.随着众包群智服务模式的发展[92],研究者们开始将问答对(Q-A pair)的创建视为一项任务分发给众包工作者(crowdworkers)来完成,这不仅可以获得大规模的问答对数据,还能显著提高数据集的质量.因此在中后期,数据集的构建方法逐渐从合成或直接抓取方法转向众包模式,例如,SQuAD 数据集使用基于 AMT 的 Daemo 平台[93],CoQA 使用基于 AMT 的 ParlAI MTurk API[94]等等.通过表 2 我们可以猜测,众包模式必然是未来建立高质量大规模 MRC 数据集的主流方法.

3.1.3 段落来源分析

通过对比表 2 的内容,我们发现完形填空和多项选择类数据集中的段落大都是来源于完整的、结构化的垂直领域数据,例如新闻、少儿图书以及学科考试等[95−98].这是因为,对于完型填空类数据集来说,空缺的位置需要与上下文有着一定的逻辑关系(否则与随机猜词没有区别),因此,采用新闻、童话故事这些叙事完整、逻辑鲜明的语料来构建该类型数据集最为合适;而对于一个高质量的多项选择类数据集来说,不仅问题和答案需要与段落具有较强的相关性,而且答案选项之间也需要有一定的迷惑性(似是而非),因此,该类型数据集大都直接取自严谨的课程测试或科学考试,这种方式既方便了数据集的建立又保证了数据集的质量.

抽取式与多轮会话类数据集中的段落则大多数来源于百科类知识库,例如维基百科或 Web 知识库.这是因为,抽取式与多轮会话类数据集中的问题和答案的形式更为自由,更符合人类在阅读理解领域的问答习惯.此时,仅有垂直领域的段落已经无法满足上述类型数据集的需求.因此,面向开放域的百科类数据或者 Web 数据更适合抽取式和多轮会话类数据集的建立.此外,如果没有众包平台的发展,仅仅把段落来源从垂直领域延伸到开

放域,也很难在短时间内构建高质量的抽取式与多轮会话类数据集.如上一小节所示:由于众包工作者是现实中的自然人,在激励的驱动下,能够针对任何形式的段落按照任务发布者的要求用自然语言生成高质量的问题和答案,从而使研究者们能够在短时间内从开放域中获得高质量的问答对.

3.1.4 数据集难度分析

通过观察图 6 和表 2 我们发现:随着时间的发展,机器阅读理解数据集变得越来越具有挑战性,越来越符合人类的自然语言习惯.

在 2017 年以前提出的数据集中,所有问题的答案都可以在段落中被找到,因此我们所建立的模型只需要考虑如何提高捕获问题和段落之间的关联特征的能力,而在答案推理能力层面则不需要注入太多精力.事实上,研究者们也发现:如果数据集中所有答案都可以在段落中找到,那么针对该数据集的模型的性能很快就可以逼近甚至超越人类水平,例如,SQuAD1.1 数据集从 40.4%EM 值的 Baseline 模型到 82.744%EM 值的 QANet 模型仅仅使用了 2 年(人类 82.304%).但现实中,这些“高性能”的模型泛化能力却很弱,因此提出更高难度、更符合人类语言习惯的数据集迫在眉睫.

在 2017 年到 2018 年之间,研究者们主要从变化答案类型的角度来增加数据集的挑战性,例如:NewsQA 通过设置 9.5%的“*null*”答案来增加数据集的难度,同时降低单词匹配类型答案的比例(相比 SQuAD1.1 下降 7.2%),增加推理和综合类型答案的比例(相比 SQuAD1.1 提高了 5.4%和 8.8%);而 MCScript 通过设置 29%的“Yes/No”答案来增加数据集的难度,在这种情况下,模型不仅需要具备更强的逻辑推理能力,还需要具备一定的辨识能力,而不是盲目地从段落中找错误的答案.

然而研究者们很快就发现,仅仅改变答案的构成并不能很有效地增加数据集的难度.这是因为,只要在构建模型时加入简单的判断函数,经过权重的训练就能提升对“不可回答问题”与“是/否类问题”的回答准确率.因此,研究者们尝试从改变数据集类型的角度出发,提出了会话、多跳推理类别的阅读理解数据集.

(1) 相比于之前的单轮对话类型数据集(例如完形填空、多项选择、抽取式和生成式等),多轮会话类型数据集有以下两个优势.

1) 多轮对话形式比单轮问答更符合人类日常语言习惯,问题和答案类型更为丰富,不仅可以包含上述的“不可回答问题”和“是否类问题”,其答案还可以是对段落的抽象与总结;

2) 由于在同一个对话框里,后面的问题答案与前面的问题答案具有很强的关联性,例如 CoQA 中(见表 1)的后续问题包含了大量的指示代词,这就要求模型具备共指消解[99]的能力.

以上两点极大地提高了数据集对模型推理能力的要求,从而使此类数据集更具有挑战性,例如,目前 QuAC 数据集榜单上性能最好的模型仅达到 68.0%*F*1 值,与人类的 81.1%还有很大差距;

(2) 多跳推理类型数据集的出现,使 MRC 任务的难度又提升了一个级别,模型无法从单一的段落得到问题的答案,而是需要从多个段落中进行链式推理才能得到正确的答案.该类数据集起到了真正考验模型的推理能力的作用,截至本文撰写日期,HotpotQA 数据集榜单第一的模型(未发表)也只达到了 67.92%的 Joint *F*1 值,而人类则达到了 82.55%.因此本文认为:在未来,多跳推理类型数据集必然是学者们研究的重点.

3.2 神经网络模型分析

机器阅读理解数据集难度的不断提升,也推动着神经阅读理解模型的发展.本文选取了 27 个具有代表性的神经阅读理解模型,并按时间线顺序进行了归纳对比,见表 3,其中,

(1) 所有模型都是原论文中的基本模型;

(2) 应用场景包含了原论文中提到的数据集以及特定数据集发布时使用该模型作为 Baseline 的场景;

(3) 上下文在这里指的是预训练词向量的上下文嵌入,不包括网络结构中对问题和文章的上下文编码;

(4) CNN 在这里特指架构中使用的神经网络模型,不包含 Char-Level 嵌入中使用的 CNN.

Table 3 Comparison of neural reading comprehension models

表 3 神经阅读理解模型对比

名称	词向量表示			神经网络模型					应用场景					
	单词	上下文	特征	LSTM	GRU	CNN	Att	Self-Att	完型	多项	抽取	对话	生成	多跳
BiDAF[47]	√	–	√	√	–	–	√	–	√	√	√	–	–	–
DCN[100]	√	–	–	√	–	–	√	–	–	–	√	–	–	–
DrQA[78]	√	–	√	√	–	–	√	–	–	–	√	√	–	–
FastQA[101]	√	–	√	√	–	–		–	–	–	√	–	–	–
MEMEN[102]	√	–	√	√	√	–	√	–	√	–	√	–	–	–
R-Net[75]	√	–	–	–	√	–	√	√	–	–	√	–	–	–
EpiReader[103]	√	–	–	–	√	√	√	–	√	–	–	–	–	–
PhaseCond[104]	√	–	–	√	√	–	√	√	–	–	√	–	–	–
FusionNet[105]	√	√	√	√	√	–	–	–	–	–	√	–	–	–
ElimiNet[106]	√	–	–	–	√	–	√	–	–	√	–	–	–	–
MRU[107]	√	–	–	√	–	–	√	–	–	√	√	–	√	–
QANet[62]	√	–	√	–	–	√	√	√	–	–	√	–	–	–
SAN[108]	√	–	√	√	√	–	√	√	–	–	√	–	–	–
GPT[109]	√	–	√	–	–	–	√	√	√	√	√	–	–	–
SLQA[110]	√	√	√	√	–	–	√	√	–	–	√	–	–	–
Read+Verify[111]	√	√	√	√	–	–	√	√	–	–	√	–	–	–
KAR[112]	√	–	√	√	–	–	√	√	–	–	√	–	–	–
RMR+[113]	√	√	√	√	–	–	√	√	–	–	√	–	√	–
BiDAF++[114]	√		√	√	–	–	√	√	–	–	√	√	–	–
U-Net[76]	√	√	√	√	–	–	√	√	–	–	√	–	–	–
BERT[14]	√	√	√	–	–	–	–	√	√	√	√	√	√	√
FlowQA[46]	√	√	√	√	√	–	√	√	–	–	–	√	–	–
CSA[115]	√	√	√	√	–	√	√	√	–	√	–	–	–	–
GraphFlow[116]	√	√	√	√	–	–	√	√	–	–	–	√	–	–
SDNet[117]	√	√	√	√	√	–	√	√	–	–	–	√	–	–
DFGN[118]	√	√	√	√	–	–	√	–	–	–	–	–	–	√
QFE[119]	√	–	–	–	√	–	√	√	–	–	–	–	–	√

3.2.1 词向量表示

从表 3 我们可以发现:单词级别的嵌入,是神经阅读理解模型中词向量表示的基础,是每一个模型必要的输入表示.目前,大多数神经阅读理解模型使用预训练的 GloVe 词向量作为单词级别的嵌入.这是因为,GloVe 词向量不仅包含单词的局部信息,同时还包含整个语料库的统计学信息,因此对单词的表示较为全面.但也有少数研究者尝试使用其他单词级别的嵌入方法来替代 GloVe,例如:Pan 等人[102]注意到,同一单词在不同语境下应该有不同的向量表示方法,他们将单词的 POS 和 NER 特征融入到 Word2Vec 词嵌入方法中,以此来获得语境化的词向量,这也使得 MEMEN 模型成为较早融入上下文嵌入思想的神经阅读理解模型,但该方法只是简单地将单词的一些语法语义特征当作输入来辅助 Word2Vec 训练词向量,并未实现真正意义上的上下文嵌入;Trischler 等人[103]认为,在完形填空任务中使用何种单词级别的嵌入并不会对后续推理产生很大影响,因此他们简单地按照均匀分布模型将词嵌入随机初始化,最终模型的结果仍然取得了当时的 State-of-the-art;Chen 等人[120]通过实验证明了使用 Fasttext 词向量替代 GloVe 可以使模型在 SQuAD1.1 数据集上的性能提升 1%.

随着 CoVe、ELMo 等上下文嵌入方法的提出,研究者们尝试将预训练的上下文嵌入作为输入词表征的一部分,用以提高词向量的表达能力.通过消融实验(ablation study),研究者们发现,使用上下文表征在一定程度上可以提升模型的性能.例如:Huang 等人[105]提出的 FusionNet 模型使用了 CoVe 上下文嵌入后,在 SQuAD1.1 数据集上 EM 值提升了 1.2%;Wang 等人[110]提出的 SLQA 模型使用了 ELMo 上下文嵌入后,在 SQuAD1.1 数据集上 EM 值提升了 2.4%;Hu 等人[111]提出的 Read+Verify 模型使用了 ELMo 上下文嵌入后,在 SQuAD2.0 数据集上 EM 值提升了 4.5%.上下文嵌入促进模型性能大幅度提升,是在 BERT 预训练模型提出后,我们可以在 SQuAD 和 CoQA 的 leaderboard 上看到:在原有的模型上加入 BERT 上下文嵌入后,其性能获得了大幅度的提升.例如,Zhu 等人[117]通过消融实验发现:若把 SDNet 模型中的 BERT 去掉,其在 CoQA 数据集上的 $F1$ 值将会下降 7.15%.因此,未来在模型中加入上下文嵌入以提高模型性能是必然趋势.

在早期,研究者们主要利用 1D CNN 模型[48,68]获取的字符级别(char-level)的嵌入来解决数据集中单词的

OOV(out-of-vocab)问题.之后,研究者们陆续将单词的其他特征加入词向量中,例如单词的 POS、NER、NTF 以及 em 特征.消融实验结果表明:当在模型的输入向量中加入这些单词更细粒度的特征信息后,可以在一定程度上提高模型的性能.例如:Seo 等人[47]通过实验表明,采用字符级别的嵌入可以使 BiDAF 模型在 SQuAD1.1 数据集上的 EM 值提高 2.7%;同样是 SQuAD1.1 数据集,Chen 等人[78]通过实验表明,在 DrQA 模型上加入 em 特征可以提高 1.5%的 EM 值,加入 POS、NER 以及 NTF 这些特征后可以提高 0.8%的 EM 值.

模型的不同词向量表示对性能的改变见表 4.

Table 4 Effect of word representations on model performance (portion)

表 4 词向量对模型性能的影响(部分)

模型	*F*1 值变化
GloVe→Fasttext[120]	+1%(SQuAD1.1)
FusionNet+CoVe[105]	+1.2%(SQuAD1.1)
SLQA+ELMo[110]	+2.4%(SQuAD1.1)
Read+Verify+ELMo[111]	+4.5%(SQuAD2.0)
SDNet−BERT[117]	−7.15%(CoQA)
BiDAF−Char−Embedding[47]	−2.7%(SQuAD1.1)
DrQA−em[78]	−1.5%(SQuAD1.1)
DrQA−feature[78]	−0.8%(SQuAD1.1)

3.2.2 神经网络模型

目前,几乎所有的神经阅读理解模型的核心部分都是在 RNNs、CNN 以及注意力机制的基础上构建而成(包括 Transformer 架构).通过表 3 我们可以看到:编码方面,除了 GPT 和 BERT 模型是采用 Transformer 架构以外,其他模型都是采用 LSTM、GRU 或者 CNN.事实上,目前大部分神经网络模型都是在本文第 2 节所归纳的架构基础上构建的.在编码层,这些模型或使用 BiLSTM 或使用 BiGRU,结构类似区分度较低;在交互层,研究者们主要的工作在于如何在灵活使用注意力机制或自注意力机制的同时加入一些新的模块,使模型的性能有所提升.因此,本小节不再赘述模型的基础架构,按时间顺序重点介绍表中模型的创新点,以期为后续的模型研究提供一些思路.

Seo 等人[47]在总结了前人在阅读理解中使用注意力机制[1,38,121−124]的研究工作之后提出了 BiDAF 模型,该模型是奠定神经阅读理解模型架构的重要模型之一.Seo 等人首次在 BiDAF 模型中引入了“双向注意力机制”,在段落对问题的注意力(C2Q)的基础上加入了问题对段落的注意力(Q2C),用来表示段落中哪一个单词与问题中的单词相关性最大,如图 9 所示(其中左图即图 3).他们通过消融实验表明:加入 Q2C 后,可使模型在 SQuAD1.1 数据集上的 *F*1 值提高 3.6%;同时,BiDAF 模型可以通过修改输出层的编码以适应不同类型的数据集.

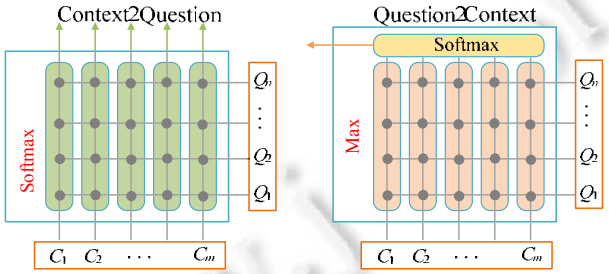


Fig.9 Bi-attention mechanism in BiDAF model

图 9 BiDAF 的双向注意力机制

Xiong 等人[100]结合了 Maxout 网络[125]和 Highway 网络[126]的优点,与 BiDAF 几乎同一时间提出了基于 HMN(highway maxout network)的 DCN 模型,并通过实验证明,HMN 模块可以使模型在 SQuAD1.1 数据集上的 *F*1 值提高 5.5%.然而,由于加入 HMN 模块会使模型变得相当复杂,并且在性能上的收益并不可观,因此在之后的模型中很少被采用.之后,Chen 等人[78]提出了 DrQA 模型,该模型主要面向开放域阅读理解,先通过 Document

Retriever 从维基百科中检索出问题相关段落,然后再通过 Document Reader 从段落中找到对应答案,利用 Wikipedia 作为知识源,然后在 SQuAD1.1 数据集上进行训练,可以比 BiDAF 模型提高 $F1$ 值 1.7%.特别地,DrQA 较早引入了 POS、NER、TF 等特征嵌入,并且获得了成功.除此之外,有关其他 Open-Domain QA 的相关模型详见第 3.1.7 小节.Weissenborn 等人[101]认为,现有的神经阅读理解模型过于复杂,计算量太大,而模型实际只需要具备两个模块:一是对文本序列的建模,二是段落和问题的交互.因此,他们提出了 FastQA 模型.顾名思义,该模型是轻量级的.Weissenborn 等人将 RNN 编码后的段落和问题进行轻量级的融合,最后通过定向搜索(beam-search)抽取答案.虽然 FastQA 模型获得了接近当时的 SOTA 结果且训练速度较快,但本文认为:过度简化模型,尤其是丢弃注意力机制,会导致模型无法更深入地融合段落和问题的潜在语义信息.不过,从提高模型的训练速度、减少模型的训练参数的角度出发来优化模型,不失为一种思路.

在神经阅读理解模型发展过程中,另一个重要模型是 Wang 等人[75]提出的 R-Net 模型.他们首次在模型中加入了自注意力机制,如图 10 所示,通过在第 2 个交互层中计算段落自己与自己的注意力值,学习已经融合了问题信息的段落内部单词之间的权重分布.这一做法获得了极大的成功,使得 R-Net 模型在 SQuAD1.1 数据集上达到了当时 SOTA88.17%的 $F1$ 值.R-Net 模型的成功,使得自注意力机制成为了后续构建神经阅读理解模型不可缺少的一部分,我们可以从表 3 看到:几乎所有在 R-Net 之后的神经阅读理解模型,都会使用自注意力机制.

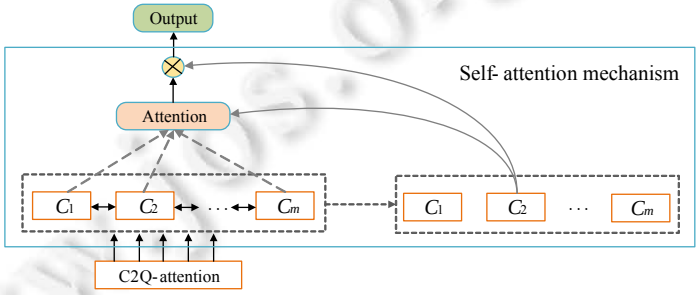


Fig.10 Self-attention mechanism in R-Net model

图 10 R-Net 的自注意力机制

之后,Yu 等人[62]针对使用 RNN 模型进行编码会导致模型训练速度变得非常缓慢的问题,提出了基于 CNN 和自注意力机制的 QANet 模型.他们表示:训练速度的提升,允许我们在同样的时间内使用更多的训练数据,从而提高模型的泛化能力.接着,他们又通过将英文数据集翻译成 k 个法文版本,然后再将每个法文数据集翻译回 k 个英文,最终得到 k^2 个英文扩展数据集,用以训练 QANet 模型并获得了当时的 SOTA 分数.Wang 等人[91]通过将交互层进行粒度划分,提出了基于分层注意力融合网络的 SLQA 模型,他们利用不同粒度的注意力机制来分重点捕获问题和段落之间的不同粒度的关联信息,大大提高了信息的融合度,从而提升了模型的性能.然而本文认为:虽然上述模型的性能表现优异,但模型结构过于复杂,导致训练速度较慢(使用 Tesla M40 单卡需要训练约 20 小时).如何优化模型加速训练,是未来对 SLQA 模型提出的新要求.Hu 等人[111]针对数据集中如何准确识别“不可回答问题”进行了深入研究,并提出了 Read+Verify 模型.他们利用两种独立的辅助损失函数:Span 损失和 No-Answer 损失,对模型进行优化,同时采用验证机制来识别似是而非的答案,进而辅助“不可回答问题”的判断,最终使模型在 SQuAD2.0 上获得了 74.3%的 $F1$ 值.Wang 等人[112]尝试在学习过程中加入知识库来提高模型的性能,并以此提出了基于 WordNet 知识库的 KAR 模型.然而,该模型在 SQuAD1.1 数据集上性能提升的效果并不理想.但本文认为:KAR 模型引入知识库的思想具有一定的创新性,或许未来能够结合该思想突破模型的性能瓶颈.Sun 等人[76]提出的 U-Net 模型通过设置一个通用节点(universal node)来连接问题和段落,用以在整个训练过程中学习来自问题和段落的全局信息.与其他模型不同的是,U-Net 从编码层到输出层都将问题和段落视为一个整体.消融实验表明:这个通用节点确实学习到了一些全局信息,使模型性能提高了 2.4%.当 U-Net 模型提出不久后,Google 就提出了 BERT 预训练模型,有关 BERT 模型的细节见第 2.5 小节,这里不再赘述.

Parikh 等人[106]针对多项选择数据集,基于人类会主动排除错误答案的思维,提出了结合排除法和选择法的ElimiNet 模型.模型使用了一个消除门(elimination gate)电路来决定哪些答案是错误的,并给予一定概率放弃与该错误答案相关的文本,最终在 RACE 数据集上,较基线模型提升了 0.4%.Chen 等人[115]认为,问题的信息是作为提取候选答案的高层特征信息的关键.因此将段落、问题和候选答案视为三维空间,并采用了 CNN-MaxPooling 动态提取相邻空间的注意力(spatial attentions).该模型相比于 ElimiNet 在 RACE 上的性能提升了 6.4%(single model).Tay 等人[107]认为,将段落按需分成多个子段落可以对后续的推理操作有所帮助,因此提出了 MRU 编码单元(multi-range reasoning units).MRU 采用了称为"合约与扩展(contract-and-expand)"的操作,将段落按合约分成 1、2、4、8 等子段落,之后进行扩展,并输入全连接网络,最后,通过设置一个门向量来输出相关度最大的信息.Tay 等人结合了双向注意力机制后,在 RACE、SearchQA 以及 NarrativeQA 上都取得了不错的性能.Hu 等人[113]针对不同层次的注意力缺失和冗余的问题,提出了重注意力机制(re-attention mechanism),用来获取注意力之间的注意力;同时,结合强化学习提出了 RMR 模型[127];之后,他们又基于 RMR 模型进一步加入了知识净化技术(knowledge distillation),将集成模型的知识迁移到单一模型上,并在 NarrativeQA 上取得了当时的 SOTA 分数.

Huang 等人[46]分析了多轮会话数据集的特点,认为:一个模型如果被期望在该类数据集中有好的性能表现,就需要具备记忆对话框中对话历史的能力.他们提出了对话框中的对话流顺序与段落内容顺序具有相似的进行趋势的假设,即每一轮问题的答案与段落位置顺序有较大的关联,并基于该假设提出了基于集成流层(integration-flow layer)的 FlowQA 模型.IF 层对问题和段落的融合信息进行两次 reshape 操作,并用一个 GRU 网络记忆对话历史和段落内容顺序的交互信息.最终,FlowQA 模型获得了 CoQA 数据集上当时 SOTA 的 75%的 $F1$ 值以及在 QuAC 数据集上当时 SOTA 的 64.1%的 $F1$ 值.Zhu 等人[117]受 Huang 等人[105]提出的 FusionNet 中历史词(history-of-word)的启发,结合 BERT 模型提出了 SDNet,并在 CoQA 数据集上斩获了当时 SOTA 的 76.6%的 $F1$ 值.Chen 等人[116]首次将图神经网络(graph neural network,简称 GNN)运用于会话类数据集,他们首先在每一轮对话上动态构建了问题-感知的段落语境图,之后,建立了一个流机制来建模语境图序列的依赖关系,提出了 GraphFlow,并在 CoQA 较 FlowQA 和 SDNet 分别提升了 2.3%和 0.7%.本文认为:虽然 GraphFlow 并没有很显著的性能提升,但他们创新性地将 GNN 用于会话类数据集,为以后解决其他 MRC 任务提供了新的思路.

与会话类数据集相似,已发表的针对于多跳推理 HotpotQA 数据集的神经网络模型目前只有少数,因此本文选取了 Leaderboard(distractor setting)上已发表的前两个模型进行分析.Xiao 等人[118]受人类在回答问题时会按步推理的启发,提出了动态融合图网络(dynamically fused graph network,简称 DFGN)模型.该模型从给定问题中的实体出发,通过动态地探索由段落中实体构建的实体图,逐步从段落中找到相关支持实体,最终在 HotpotQA 数据集上获得了 59.82%的 Joint $F1$ 值.Nishida 等人[119]则将重心放在如何更加精准地获取 HotpotQA 的支持证据上,他们认为:现有的方法只是独立地评估每个句子的重要性,以此选出支持证据,因此模型无法考虑到句子之间的依赖关系以及问题句子中的重要信息.他们提出了 QFE 模型(query focused extractor),QFE 能够通过 RNN 和注意力机制从段落中抽取与问题相关的支持证据.该模型虽然在 HotpotQA 数据集上只获得了 59.61%的 Joint $F1$ 值(榜单第九),但在支持证据 Sup $F1$ 值上却达到了 84.49%(榜单第二),说明 QFE 模型确实起到了一定的作用.

本文认为:虽然包括 ARC、SquAD、CoQA 以及 HotpotQA 等在内的各类数据集榜单纪录不断地被加入 BERT 架构的神经阅读理解模型所刷新,但 BERT 预训练模型仍存在一些缺陷,例如模型本身的预训练需要喂入海量数据(33 亿词量),同时,研究者或机构需要具备相当先进的硬件设备条件(谷歌公司使用 64 个 TPU 训练了约 4 天,之后用 1 024 块 TPU 将时间缩短至 76 分钟[128]),因此对于一般的研究人员或机构而言,首要考虑的还是如何优化神经网络模型的架构,从而进一步提高基于 BERT 的模型架构微调后的性能.

为了更加清晰地显示每一种模型的创新点以及在数据集上的性能,本文将其归纳为表 5.

Table 5 The innovation and performance of models

表 5 模型创新点与性能

模型	创新点	性能	模型	创新点	性能
BiDAF[47]	双向注意力机制	77.3 *F*1 (SQuAD1.1)	U-Net[76]	通用节点记录信息	72.6 *F*1 (SQuAD2.0)
DCN[100]	HMN 模块	82.8 *F*1 (SQuAD1.1)	BERT[14]	预训练模型	91.8 *F*1 (SQuAD1.1)
DrQA[78]	Wikipedia 知识源	79.0 *F*1 (SQuAD1.1)	ElimiNet[106]	引入排除法	44.5 Acc (RACE)
DrQA	Wikipedia 知识源	25.4 EM (TREC)	CSA[115]	空间卷积-池化	50.9 Acc (RACE)
DrQA	Wikipedia 知识源	36.5 EM (WikiMov.)	MRU[107]	MRU 单元	50.4 Acc (RACE)
FastQA[101]	轻量级模型	77.1 *F*1 (SQuAD1.1)	MRU	MRU 单元	19.8 Bleu-4 (Narra.)
R-Net[75]	自注意力机制	88.2 *F*1 (SQuAD1.1)	RMR+[113]	重注意力、知识净化	27.5 Bleu-4 (Narra.)
QANet[62]	CNN+自注意力	87.8 *F*1 (SQuAD1.1)	FlowQA[46]	引入对话流	75.0 *F*1 (CoQA)
SLQA[110]	分层注意力融合	82.8 *F*1 (SQuAD1.1)	GraphFlow[116]	引入图神经网络	77.3 *F*1(CoQA)
Read+Ver[111]	独立辅助损失函数	74.3 *F*1 (SQuAD2.0)	DFGN[118]	动态融合实体图	59.82 *F*1 (HotpotQA)
KAR[112]	引入知识库辅助	83.5 *F*1 (SQuAD1.1)	QFE[119]	聚焦提取支持证据	59.61 *F*1 (HotpotQA)

3.2.3 应用场景

从表 3 我们可以看到:在前中期,研究者们所提出的神经阅读理解模型主要应用于抽取式数据集,而很少应用于完形填空数据集.本文认为主要原因是:与完形填空数据集相比,抽取式数据集更加符合人类的自然语言习惯,更适合评价神经阅读理解模型的性能,虽然所提出的神经阅读理解模型可以通过改变输出层的预测函数来应用于完形填空数据集(如第 2.4 节所示),但研究者们仍然更倾向于抽取式数据集.相比于抽取式数据集,研究者们对多项选择和生成式数据集的研究热度相对有所降低,但我们可以发现:这些年来,仍有部分研究者针对这两类数据集提出了性能更好的模型.随着多轮会话与多跳推理数据集的兴起,研究者们开始构建适合于该类数据集的神经阅读理解模型.虽然近半年来已有不少模型被提出,但大部分成果仍未被转化为文献形式,因此本文无法在此进行归纳分析,读者可参考 CoQA 与 HotpotQA 的 Leaderboard 获得更多了解.

4 总结与研究展望

机器阅读理解是当今计算机自然语言处理领域的核心难点问题,其解决具有重要的理论意义和良好的应用前景.尽管基于神经网络的机器阅读理解在近几年来发展迅速,人们在构建各种各样的大规模 MRC 数据集的同时,性能更高的神经阅读理解模型也被不断提出,两者相辅相成,共同推进着 MRC 领域的发展,然而使机器达到真正的人类阅读理解水平,研究者们还有很长的路要走.本文在第 1 节详细总结了机器阅读理解的发展历程和任务定义.在第 2 节归纳介绍了神经阅读理解模型框架以及最新的 BERT 预训练模型.在第 3 节归纳了近年来该领域的主流数据集以及神经阅读理解模型,并详细分析了各自的优点与不足.总的来说,目前机器阅读理解任务仍处于研究探索阶段,各方面还存在许多问题与挑战,诸如:

(1) 模型缺乏深层次的推理能力

早期的数据集中存在的问题仍然没有得到真正的解决,即使更具挑战性的数据集正在被不断地提出.针对 SQuAD1.1 数据集,虽然现有的模型已经获得了超过人类水平的性能,但仔细研究后发现,这些模型仍然会犯很多低级的错误,比如:模型无法理解“BdefeatedA”就是“B won”的意思,且涉及“比较类”问题就常会出错等等[129].这表明,现有模型仅仅是做到了更完善、更复杂的浅层次匹配,对于段落与问题之间内在蕴意的深层推理能力仍非常薄弱.

(2) 模型的鲁棒性与泛化能力太差

Jia 等人[130]通过对抗性实验发现:如果我们在段落末尾加入一些无关句子(distracting sentence),这些句子与问题会有一些单词重叠但不影响答案的正确性,这时,当前的模型性能就会急剧下降近一半;而当这些句子是不符合语法的单词序列时,模型的性能会进一步下降.这说明,现有的模型鲁棒性太差,一旦数据集带有噪声,其性能就会急剧下降,导致无法将该模型部署到实际应用中.除此之外,我们如果将已经训练好的模型应用于其他不同文本来源和构造方法的数据集,其性能也会急剧下降,这表明,现有模型的泛化能力太差.

(3) 对于模型来说,是表征重要还是架构重要

通过对神经阅读理解模型的归纳分析后我们发现:为了更好地捕捉段落和问题的相似度,研究者们提出了越来越复杂的注意力机制.这样做确实可以在一定程度上提高模型的性能,但 Devlin 等人[14]却表示:在海量文本语料库上预训练一个架构简单的深度语言模型(BERT),即使不需要对段落和问题进行任何融合操作,也可以通过参数微调(fine-tuned)获得非常好的性能.然而通过分析经典数据集的 Leaderboard(SquAD,ARC,CoQA 以及 HotpotQA)后我们发现:虽然基于 BERT 所构建的模型能够在榜单上达到非常优异的排名,但这类模型或未能形成文献形式而被发表,或在消融实验时被发现去掉 BERT 后性能会大幅度下降[117,118].那么我们究竟该如何平衡两者之间的关系,在利用好 BERT 的同时思考如何优化网络架构,从而进一步提升基于 BERT 的模型的性能,而不是仅仅把 BERT 当成提高模型性能的唯一方法?这在未来仍然是一个问题.

(4) 模型的可解释性太差

现有模型对最后答案的预测并没有提供充分的理论依据,即目前端到端神经网络的黑盒模型弊端在神经阅读理解模型中仍然存在,这会降低模型使用者对其的信任程度,从而难以在例如医学、法律这些敏感领域进行实际应用部署.

因此,将来的研究工作可以从以下几个方面展开.

(1) 构建更贴近人类自然语言习惯的数据集

提问者进行问题生成时可以脱离已有的段落内容,避免问题潜在地模仿段落句子结构或重用段落的单词内容,这样做不仅能增加数据集的难度,还能使数据集更符合人类问答习惯;此外,尽管现有数据集中已有生成式类型(例如 NarrativeQA[32]和 DuReader[36]),但其仍缺乏考验机器抽象能力的问题,因此,研究者们可以在构建数据集时加入更多抽象式问题,例如主题凝练.

(2) 构建兼具速度与性能的模型

现有模型的主要研究点在于如何提高模型在特定数据集上的性能,而忽视了模型的训练速度,这会导致一个现象:人们倾向于付出巨额的 GPU 或 TPU 资源来训练非常复杂的模型,以期在性能上有质的突破,但实际上并未达到预期效果.虽然已有部分模型从加快训练速度的角度考虑来构建模型,例如 FastQA[101]、QANet[62],但基于 RNNs 的架构仍然是目前构建神经网络模型的主流方法.未来我们可以在构建模型时重点考虑训练速度,例如考虑合理地将 LSTM 替换为 Transformer 或者 CNN 架构,这样可以提高训练速度,允许我们在相同时间内训练更多的数据集,进而提升模型的可扩展性.

(3) 在训练中融入对抗实例,以提高模型的鲁棒性与泛化能力

未来研究中,我们需要考虑如何在训练过程中加入对抗实例以提高模型的鲁棒性,从而使模型在具有噪声的数据集上也能保持一定的性能;此外,如何将迁移学习(transfer learning)和多任务学习(multi-task learning)应用到神经网络模型中,构建跨数据集的高性能模型,也是未来的研究方向.

(4) 提高模型的可解释性

未来研究中,我们可以在构建数据集时加入支持证据,让模型在每一次预测时提供相关证明;此外,尝试在构建模型时加入原理生成模块(rationales generating),让模型在预测答案之前优先给出对应的理由,也是未来的研究方向.

致谢 斯坦福大学陈丹琦博士的博士论文《Neural Reading Comprehension and Beyond》给我们以启发,在此表示衷心的感谢.

References:

[1] Hermann KM, Kociský T, Grefenstette E, *et al*. Teaching machines to read and comprehend. In: Proc. of the Neural Information Processing Systems. 2015. 1693−1701.

[2] Lehnert WG. The process of question and answering [Ph.D. Thesis]. Yale University, 1977.

[3] Hinton GE, Osindero S, Teh YW, *et al*. A fast learning algorithm for deep belief nets. Neural Computation, 2006,18(7):1527−1554.

[4] Salton G, Mcgill MJ. Introduction to modern information retrieval. In: Proc. of the Introduction to Modern Information Retrieval. 1983.

[5] Kronenfeld DB, Schank RC, Abelson RP, *et al*. Scripts, plans, goals, and understanding: An inquiry into human knowledge structures. Language, 1978,54(3).

[6] Berant J, Chou AK, Frostig R, *et al*. Semantic parsing on freebase from question-answer pairs. In: Proc. of the 2013 Conf. on EMNLP. 2013. 1533−1544.

[7] Hirschman L, Light M, Breck E, *et al*. Deep read: A reading comprehension system. In: Proc. of the 37th Conf. on ACL. 1999. 325−332.

[8] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Proc. of the Int'l Conf. on Machine Learning. 2006.

[9] Richardson M, Burges CJC, Renshaw E. MCTest: A challenge dataset for the open-domain machine comprehension of text. In: Proc. of the 2013 Conf. on EMNLP. 2013. 193−203.

[10] Narasimhan K, Barzilay R. Machine comprehension with discourse relations. In: Proc. of the IJCNLP. 2015. 1253−1262.

[11] Sachan M, Dubey KA, Xing EP, *et al*. Learning answer-entailing structures for machine comprehension. In: Proc. of the IJCNLP. 2015. 239−249.

[12] Wang H, Bansal M, Gimpel K, *et al*. Machine comprehension with syntax, frames, and semantics. In: Proc. of the IJCNLP. 2015. 700−706.

[13] Rajpurkar P, Zhang J, Lopyrev K, *et al*. SQuAD: 100000+ questions for machine comprehension of text. In: Proc. of the 2016 Conf. on EMNLP. 2016. 2383−2392.

[14] Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. 2018. https://arxiv.org/abs/1810.04805

[15] Joshi M, Choi E, Weld DS, *et al*. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In: Proc. of the 55th Conf. on ACL. 2017. 1601−1611.

[16] Yang Y, Yih W, Meek C, *et al*. WikiQA: A challenge dataset for open-domain question answering. In: Proc. of the 2015 Conf. on EMNLP. 2015. 2013−2018.

[17] Trischler A, Wang T, Yuan X, *et al*. NewsQA: A machine comprehension dataset. In: Proc. of the 2nd Workshop on Representation Learning for NLP. 2017. 191−200.

[18] Rajpurkar P, Jia R, Liang P. Know what you don't know: Unanswerable questions for SquAD. In: Proc. of the 56th Conf. on ACL. 2018. 784−789.

[19] Dunn M, Sagun L, Higgins M, *et al*. SearchQA: A new Q&A dataset augmented with context from a search engine. arXiv: Computation and Language, 2019. https://arxiv.org/pdf/1704.05179.pdf

[20] Welbl J, Liu NF, Gardner M, *et al*. Crowdsourcing multiple choice science questions. In: Proc. of the Workshop on Noisy User-generated Text (W-NUT). 2017. 94−106.

[21] Clark P, Cowhey I, Etzioni O, *et al*. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv: Artificial Intelligence, Springer-Verlag, 2018.

[22] Lai GK, Xie QZ, Liu HX, *et al*. RACE: Large-scale reading comprehension dataset from examinations. In: Proc. of the 2017 Conf. on EMNLP. 2017. 785−794.

[23] Kembhavi A, Seo M, Schwenk D, *et al*. Are you smarter than a sixth grader? Textbook question answering for multimodal machine comprehension. In: Proc. of the IEEE Conf. on CVPR. IEEE Computer Society, 2017. 5376−5384.

[24] Ostermann S, Modi A, Roth M, *et al*. MCScript: A novel dataset for assessing machine comprehension using script knowledge. In: Proc. of the Language Resources and Evaluation. 2018. 3567−3574.

[25] Hill F, Bordes A, Chopra S, *et al*. The goldilocks principle: Reading children's books with explicit memory representations. In: Proc. of the ICLR. 2016.

[26] Xie Q, Lai G, Dai Z, *et al*. Large-scale cloze test dataset created by teachers. In: Proc. of the 2018 Conf. on EMNLP. 2018. 2344−2356.

[27] Reddy S, Chen D, Manning CD, *et al*. CoQA: A conversational question answering challenge. 2018. https://arxiv.org/abs/1808.07042

[28] Choi E, He H, Iyyer M, *et al*. QuAC: Question answering in context. In: Proc. of the 2018 Conf. on EMNLP. 2018. 2174−2184.

[29] Saha A, Pahuja V, Khapra MM, *et al*. Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. In: Proc. of the AAAI. 2018. 705−713.

[30] Iyyer M, Yih W, Chang M, *et al*. Search-based neural structured learning for sequential question answering. In: Proc. of the 55th Conf. on ACL. 2017. 1821−1831.

[31] Talmor A, Berant J. The Web as a knowledge-base for answering complex questions. In: Proc. of the NAACL. 2018. 641−651.

[32] Kociský T, Schwarz J, Blunsom P, *et al*. The NarrativeQA reading comprehension challenge. Trans. of the Association for Computational Linguistics, 2018, 317−328.

[33] Nguyen T, Rosenberg M, Song X, *et al*. MS MARCO: A human-generated machine reading comprehension dataset. In: Proc. of the 31st Conf. on NIPS. 2017.

[34] Yang Z, Qi P, Zhang S, *et al*. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In: Proc. of the 2018 Conf. on EMNLP. 2018. 2369−2380.

[35] Cui Y, Liu T, Chen Z, *et al*. Consensus attention-based neural networks for Chinese reading comprehension. In: Proc. of the 26th COLING. 2016. 1777−1786.

[36] He W, Liu K, Liu J, *et al*. DuReader: A Chinese machine reading comprehension dataset from real-world applications. In: Proc. of the Workshop on Machine Reading for Question Answering. 2017. 37−46.

[37] Sukhbaatar S, Szlam A, Weston J, *et al*. End-to-end memory networks. In: Proc. of the 29th Conf. on NIPS. 2015. 2440−2448.

[38] Wang S, Jiang J. Machine comprehension using match-LSTM and answer pointer. In: Proc. of the ICLR. 2017.

[39] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. In: Proc. of the 31st Conf. on NIPS. 2017. 5998−6008.

[40] Papineni K, Roukos S, Ward T, *et al*. Bleu: A method for automatic evaluation of machine translation. In: Proc. of the 40th Conf. on ACL. 2002. 311−318.

[41] Lin C. ROUGE: A package for automatic evaluation of summaries. In: Proc. of the 42th Conf. on ACL. 2004. 74−81.

[42] Denkowski MJ, Lavie A. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In: Proc. of the Workshop on Statistical Machine Translation. 2011. 85−91.

[43] Gardner M. A survey of the current state of reading comprehension. 2016. http://matt-gardner.github.io/paper-thoughts/2016/12/08/reading-comprehension-survey.html

[44] Arivuchelvan KM, Lakahmi K. Reading comprehension system—A review. Indian Journal of Science, 2017,14 (1):83−90.

[45] Lai TM, Bui T, Li S, *et al*. A review on deep learning techniques applied to answer selection. In: Proc. of the 27th COLING. 2018. 2132−2144.

[46] Huang HY, Choi E, Yih WT. FlowQA: Grasping flow in history for conversational machine comprehension. arXiv: Artificial Intelligence, 2018. https://arxiv.org/pdf/1810.06683.pdf

[47] Seo MJ, Kembhavi A, Farhadi A, *et al*. Bidirectional attention flow for machine comprehension. In: Proc. of the ICLR. 2017.

[48] Kim Y. Convolutional neural networks for sentence classification. In: Proc. of the 2014 Conf. on EMNLP. 2014. 1746−1751.

[49] Bengio Y, Ducharme R, Vincent P, *et al*. A neural probabilistic language model. Journal of Machine Learning Research, 2003,3(6):1137−1155.

[50] Mikolov T, Chen K, Corrado GS, *et al*. Efficient estimation of word representations in vector space. 2013. https://arxiv.org/abs/1301.3781

[51] Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. In: Proc. of the 27th Conf. on NIPS. 2013. 3111−3119.

[52] Pennington J, Socher R, Manning CD. GloVe: GlobalVectors for word representation. In: Proc. of the 2014 Conf. on EMNLP. 2014. 1532−1543.

[53] Bojanowski P, Grave E, Joulin A, *et al*. Enriching word vectors with subword information. Trans. of the Association for Computational Linguistics, 2017,5:135−146.

[54] Lai S, Liu K, He S, *et al*. How to generate a good word embedding. IEEE Intelligent Systems, 2016,31(6):5−14.

[55] Bakarov A. A survey of word embeddings evaluation methods. arXiv: Computation and Language, 2018. https://arxiv.org/abs/1801.09536

[56] Artetxe M, Labaka G, Lopezgazpio I, *et al*. Uncovering divergent linguistic information in word embeddings with lessons for intrinsic and extrinsic evaluation. In: Proc. of the Conf. on CoNLL. 2018. 282−291.

[57] Zhao JQ, Gui XL. Deep convolution neural networks for twitter sentiment analysis. IEEE Access, 2018,6:23253−23260.

[58] Naili M, Chaibi AH, Ben Ghezala HH. Comparative study of word embedding methods in topic segmentation. Procedia Computer Science, 2017,112:340−349.

[59] Mccann B, Bradbury J, Xiong C, *et al*. Learned in translation: Contextualized word vectors. In: Proc. of the 31st Conf. on NIPS. 2017. 6294−6305.

[60] Peters ME, Neumann M, Iyyer M, *et al*. Deep contextualized word representations. In: Proc. of the NAACL-HLT. 2018. 2227−2237.

[61] Wu HC, Luk RW, Wong K, *et al*. Interpreting TF-IDF term weights as making relevance decisions. ACM Trans. on Information Systems, 2008,26(3).

[62] Yu AW, Dohan D, Luong M, *et al*. QANet: Combining local convolution with global self-attention for reading comprehension. In: Proc. of the ICLR. 2018. http://export.arxiv.org/abs/1804.09541v1

[63] Williams RJ, Zipser D. A learning algorithm for continually running fully recurrent neural networks. Neural Computation, 1989, 1(2):270−280.

[64] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997,9(8):1735−1780.

[65] Cho K, Van Merrienboer B, Gulcehre C, *et al*. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proc. of the 2014 Conf. on EMNLP. 2014. 1724−1734.

[66] Lecun Y, Bengio Y. Convolutional networks for images, speech, and time series. In: The Handbook of Brain Theory and Neural Networks. MIT Press, 1998.

[67] Cui Y, Chen Z, Wei S, *et al*. Attention-over-attention neural networks for reading comprehension. In: Proc. of the 55th Conf. on ACL. 2017. 593−602.

[68] Zhang X, Zhao JJ, Lecun Y, *et al*. Character-level convolutional networks for text classification. In: Proc. of the 29th Conf. on NIPS. 2015. 649−657.

[69] Sutskever I, Vinyals O, Le QV, *et al*. Sequence to sequence learning with neural networks. In: Proc. of the 28th Conf. on NIPS. 2014. 3104−3112.

[70] Luong T, Pham H, Manning CD, *et al*. Effective approaches to attention-based neural machine translation. In: Proc. of the 2015 Conf. on EMNLP. 2015. 1412−1421.

[71] Bahdanau D, Cho K, Bengio Y, *et al*. Neural machine translation by jointly learning to align and translate. In: Proc. of the ICLR. 2015.

[72] Rocktaschel T, Grefenstette E, Hermann KM, *et al*. Reasoning about entailment with neural attention. In: Proc. of the ICLR. 2016.

[73] Rush AM, Chopra S, Weston J, *et al*. A neural attention model for abstractive sentence summarization. In: Proc. of the 2015 Conf. on EMNLP. 2015. 379−389.

[74] Yin W, Schutze H, Xiang B, *et al*. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. Trans. of the Association for Computational Linguistics, 2016,4(1):259−272.

[75] Wang WH, Yang N, Wei FR, *et al*. Gated selfmatching networks for reading comprehension and question answering. In: Proc. of the 55th Conf. on ACL. 2017. 189−198.

[76] Sun F, Li L, Qiu X, *et al*. U-Net: Machine reading comprehension with unanswerable questions. arXiv: Computation and Language, 2018. https://arxiv.org/abs/1810.06638

[77] Vinyals O, Fortunato M, Jaitly N, *et al*. Pointer networks. In: Proc. of the 29th Conf. on NIPS. 2015. 2692−2700.

[78] Chen D, Fisch A, Weston J, *et al*. Reading Wikipedia to answer open-domain questions. In: Proc. of the 55th Conf. on ACL. 2017. 1870−1879.

[79] Radford A, Narasimhan K, Salimans T, *et al*. Improving language understanding by generative pre-training. 2018. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

[80] Zhu Y, Kiros R, Zemel RS, *et al*. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proc. of the IEEE ICCV. 2015. 19−27.

[81] Kalchbrenner N, Espeholt L, Simonyan K, *et al*. Neural machine translation in linear time. arXiv: Computation and Language, 2016. https://arxiv.org/abs/1610.10099

[82] Taylor WL. “Cloze procedure”: A new tool for measuring readability. Journalism Bulletin, 1953,30(4):415−433.

[83] Bollacker K, Evans C, Paritosh P, *et al*. Freebase: A collaboratively created graph database for structuring human knowledge. In: Proc. of the Int’l Conf. on Management of Data. 2008. 1247−1250.

[84] Bordes A, Usunier N, Chopra S, *et al*. Large-scale simple question answering with memory networks. arXiv: Learning, 2015. https://arxiv.org/abs/1506.02075

[85] Fader A, Zettlemoyer LS, Etzioni O, *et al*. Open question answering over curated and extracted knowledge bases. In: Proc. of the Knowledge Discovery and Data Mining. 2014. 1156−1165.

[86] Brill ED, Dumais ST, Banko M, *et al*. An analysis of the AskMSR question-answering system. In: Proc. of the 2002 Conf. on EMNLP. 2002. 257−264.

[87] Ferrucci DA, Brown EW, Chucarroll J, *et al*. Building Watson: An overview of the DeepQA project. AI Magazine, 2010,31(3): 59−79.

[88] Baudis P, Sedivý J. Modeling of the question answering task in the YodaQA system. In: Proc. of the Cross Language Evaluation Forum. 2015. 222−228.

[89] Berant J, Chou AK, Frostig R, *et al*. Semantic parsing on freebase from question-answer pairs. In: Proc. of the 2013 Conf. on EMNLP. 2013. 1533−1544.

[90] Miller AH, Fisch A, Dodge J, *et al*. Key-value memory networks for directly reading documents. In: Proc. of the 2016 Conf. on EMNLP. 2016. 1400−1409.

[91] Riloff E, Thelen M. A rule-based question answering system for reading comprehension tests. In: Proc. of the Workshop on Reading Comprehension (NAACL/ANLP 2000). 2000. 13−19.

[92] Mason WA, Suri S. Conducting behavioral research on Amazon’s mechanical Turk. Behavior Research Methods, 2012,44(1):1−23.

[93] Gaikwad S, Morina D, Nistala R, *et al*. Daemo: A self-governed crowdsourcing marketplace. In: Proc. of the 28th ACM UIST. 2015. 101−102.

[94] Miller AH, Feng W, Batra D, *et al*. ParlAI: A dialog research software platform. In: Proc. of the 2017 Conf. on EMNLP. 2017. 79−84.

[95] Svore KM, Vanderwende L, Burges CJ, *et al*. Enhancing single-document summarization by combining RankNet and third-party sources. In: Proc. of the 2007 Conf. on EMNLP. 2007. 448−457.

[96] Woodsend K, Lapata M. Automatic generation of story highlights. In: Proc. of the 48th Conf. on ACL. 2010. 565−574.

[97] Clark P, Etzioni O, Khot T, *et al*. Combining retrieval, statistics, and inference to answer elementary science questions. In: Proc. of the AAAI. 2016. 2580−2586.

[98] Schoenick C, Clark P, Tafjord O, *et al*. Moving beyond the turing test with the Allen AI science challenge. Communications of the ACM, 2017,60(9):60−64.

[99] Zhang JP, Chapman WW, Crowley RS. Coreference resolution: A review of general methodologies and applications in the clinical domain. Journal of Biomedical Informatics, 2011,44(6):1113−1122.

[100] Xiong C, Zhong V, Socher R, *et al*. Dynamic coattention networks for question answering. In: Proc. of the ICLR. 2017.

[101] Weissenborn D, Wiese G, Seiffe L, *et al*. Making neural QA as simple as possible but not simpler. In: Proc. of the Conf. on CoNLL. 2017. 271−280.

[102] Pan B, Li H, Zhao Z, *et al*. MEMEN: Multi-layer embedding with memory networks for machine comprehension. arXiv: Artificial Intelligence, 2017. https://arxiv.org/pdf/1707.09098.pdf

[103] Trischler A, Ye Z, Yuan X, *et al*. Natural language comprehension with the EpiReader. In: Proc. of the 2016 Conf. on EMNLP. 2016. 128−137.

[104] Liu R, Wei W, Mao W, *et al*. Phase conductor on multi-layered attentions for machine comprehension. arXiv: Computation and Language, 2018. https://arxiv.org/pdf/1710.10504.pdf

[105] Huang H, Zhu C, Shen Y, *et al*. FusionNet: Fusing via fully-aware attention with application to machine comprehension. In: Proc. of the ICLR. 2018.

[106] Parikh S, Sai AB, Nema P, *et al*. ElimiNet: A model for eliminating options for reading comprehension with multiple choice questions. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence (IJCAI). 2018. 4272−4278.

[107] Tay Y, Tuan LA, Hui SC, *et al*. Multi-range reasoning for machine comprehension. arXiv: Computation and Language, 2018. https://arxiv.org/abs/1803.09074v1

[108] Liu X, Shen Y, Duh K, *et al*. Stochastic answer networks for machine reading comprehension. In: Proc. of the 56th Conf. on ACL. 2018. 1694−1704.

[109] Radford A, Narasimhan K, Salimans T, *et al*. Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf

[110] Wang W, Yan M, Wu C, *et al*. Multi-granularity hierarchical attention fusion networks for reading comprehension and question answering. In: Proc. of the 56th Conf. on ACL. 2018. 1705−1714.

[111] Hu M, Wei F, Peng Y, *et al*. Read+Verify: Machine reading comprehension with unanswerable questions. arXiv: Computation and Language, 2018. https://arxiv.org/pdf/1808.05759.pdf

[112] Wang C, Jiang H. Exploring machine reading comprehension with explicit knowledge. arXiv: Artificial Intelligence, 2018. https://arxiv.org/pdf/1809.03449.pdf

[113] Hu M, Peng Y, Huang Z, *et al*. Attention-guided answer distillation for machine reading comprehension. In: Proc. of the 2018 Conf. on EMNLP. 2018. 2077−2086.

[114] Yatskar M. A qualitative comparison of CoQA, SQuAD 2.0 and QuAC. arXiv: Computation and Language, 2018. https://arxiv.org/pdf/1809.10735.pdf

[115] Chen Z, Cui Y, Ma W, *et al*. Convolutional spatial attention model for reading comprehension with multiple-choice questions. In: Proc. of the National Conf. on Artificial Intelligence. 2019.

[116] Chen Y, Wu LF, Zaki MJ. GRAPHFLOW : Exploiting conversation flow with graph neural networks for conversational machine comprehension. 2019. https://graphreason.github.io/papers/13.pdf

[117] Zhu C, Zeng M, Huang X. SDNet: Contextualized attention-based deep network for conversational question answering. arXiv: Computation and Language, 2019. https://arxiv.org/pdf/1812.03593.pdf

[118] Xiao Y, Qu Y, Qiu L, *et al*. Dynamically fused graph network for multi-hop reasoning. arXiv: Computation and Language, 2019. https://arxiv.org/abs/1905.06933

[119] Nishida K, Nishida K, Nagata M, *et al*. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. arXiv: Computation and Language, 2019. https://arxiv.org/abs/1905.08511

[120] Chen D, Bolton J, Manning CD, *et al*. A thorough examination of the CNN/daily mail reading comprehension task. In: Proc. of the 54th Conf. on ACL. 2016. 2358−2367.

[121] Weston J, Chopra S, Bordes A, *et al*. Memory networks. In: Proc. of the ICLR. 2015.

[122] Bahdanau D, Cho K, Bengio Y, *et al*. Neural machine translation by jointly learning to align and translate. In: Proc. of the ICLR. 2015.

[123] Kadlec R, Schmid M, Bajgar O, *et al*. Text understanding with the attention sum reader network. In: Proc. of the 54th Conf. on ACL. 908−918.

[124] Xiong C, Merity S, Socher R, *et al*. Dynamic memory networks for visual and textual question answering. In: Proc. of the Int'l Conf. on Machine Learning. 2016. 2397−2406.

[125] Goodfellow IJ, Wardefarley D, Mirza M, *et al*. Maxout networks. ICML, 2013,28(3):1319−1327.

[126] Srivastava RK, Greff K, Schmidhuber J, *et al*. Training very deep networks. In: Proc. of the 29th Conf. on NIPS. 2015. 2377−2385.

[127] Hu M, Peng Y, Huang Z, *et al*. Reinforced mnemonic reader for machine reading comprehension. In: Proc. of the Int'l Joint Conf. on Artificial Intelligence (IJCAI). 2018. 4099−4106.

[128] You Y, Li J, Reddi S, *et al*. Large batch optimization for deep learning: Training BERT in 76 minutes. arXiv: Machine Learning, 2019. https://arxiv.org/abs/1904.00962

[129] Chen DQ. Neural reading comprehension and beyond [Ph.D. Thesis]. Stanford University, 2018. https://cs.stanford.edu/~danqi/papers/thesis.pdf

[130] Jia R, Liang P. Adversarial examples for evaluating reading comprehension systems. In: Proc. of the 2017 Conf. on EMNLP. 2017. 2021–2031.

顾迎捷(1992－),男,学士,主要研究领域为机器阅读理解,自然语言处理.

桂小林(1966－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为物联网,云计算,大数据分析与隐私保护,信息安全.

李德福(1996－),男,学士,主要研究领域为机器阅读理解,自然语言处理.

沈毅(1994－),男,学士,主要研究领域为机器阅读理解,自然语言处理.

廖东(1995－),男,学士,主要研究领域为大数据分析,边缘计算.