文章编号：1003-0077(2018)12-0125-07

基于自注意力机制的阅读理解模型

张浩宇，张鹏飞，李真真，谭庆平

（国防科技大学 计算机学院，湖南 长沙 410072）

摘　要：机器阅读理解是自然语言处理领域一项得到广泛关注与研究的任务。该文针对中文机器阅读理解数据集DuReader，分析其数据集的特点及难点，设计了一种基于循环神经网络和自注意力机制的抽取式模型 Mixed Model。通过设计段落融合等策略，该文提出的模型在 DuReader 测试集上达到了 54.2 的 Rouge-L 得分和 49.14 的 Bleu-4 得分。

关键词：机器阅读理解；深度学习；自注意力机制

中图分类号：TP391　　**文献标识码**：A

Self-attention Based Machine Reading Comprehension

ZHANG Haoyu, ZHANG Pengfei, LI Zhenzhen, TAN Qingping

(School of Computer Science, National University of Defense Technology, Changsha, Hunan 410072, China)

Abstract: Machine reading comprehension has attracted concerns in the field of Natural Language Processing. To deal with the Chinese machine reading comprehension data set —DuReader, this paper presents an extractive language model called Mixed Model with multiple strategies including recurrent neural network, paragraph fusion and self-attention mechanism. The proposed method achieves a Rouge-L score of 54.2 and a Bleu-4 score of 49.14 on the DuReader test set.

Keywords: machine reading comprehension; deep learning; self attention mechanism

0　引言

深度学习在近年来得到了广泛的研究和发展。基于深度神经网络模型在计算机视觉、语音识别、自然语言处理等领域的困难任务上取得了接近甚至超越人类的水平。

机器阅读理解（Machine Reading Comprehension，MRC），是在限定上下文（context documents）的前提下为给定问题找到答案的一类任务，是自然语言处理（NLP）中的核心任务之一，近年来得到了广泛的研究。[1]

机器阅读理解任务可以形式化的描述为，对于给定问题 q 及其对应的文本形式的候选文档集合 d＝{d1, d2, ..., dn}，要求参评阅读理解系统自动对问题及候选文档进行分析，输出能够满足问题的文本答案 a。对于人工合成任务以及完形填空任务来说，给定三元组(q, d, a)，要求答案 a 必须出现在文档 d 中即 a⊂d。

1　数据集

由于开放域的阅读理解任务难度较大，且神经网络模型训练需要的标注数据量较大，为了控制任务的难度并保证数据量，很多机构发布了阅读理解相关的大规模标注数据集，推动了机器阅读理解任务研究的进展。

完形填空类型的机器阅读理解，是将问题中需要回答的单词以占位符代替。给定上下文，需要将问题句子中被替换的单词或者实体词预测补全出来，一般要求这个被抽掉的单词或者实体词是在文章中出现过的。数据集 CNN/Daily Mail（Hermann

收稿日期：2018-06-26　**定稿日期**：2018-08-13

基金项目：国家重点研发计划(2016YFB0200401)；新世纪优秀人才，国防科技卓越人才(2017-JCJQ-ZQ-013)；湖南省人才计划(2017RS3045)

et al. 2015)[2]，通过以新闻作为上下文、抽取新闻故事的摘要作为问题的方式构造。Hill 等在 2016 年提出的 CBT(The Children's Book Test)[3]数据集则以儿童故事集作为数据来源，通过筛选命名实体、动词、通用名词以及副词等四种不同类型的答案词控制任务的难度以及侧重点，并为每个样本提供了若干个候选答案，进一步降低了数据集的整体难度。

Rajpurkar 等在 2016 年以维基百科的 536 篇文章作为上下文，发布了 SQuAD(Stanford Question Answering Dataset)[4]数据集并得到了广泛的关注。SQuAD 与之前的完形填空类阅读理解数据集如 CNN/Daily Mail，CBT 等最大的区别在于：SQuAD 中的答案不再是单个实体或单词，而可能是一段连续的短语，这使得其答案更难预测。

随着对阅读理解研究的深入，2017 年以来一些难度更高的数据集被发布。Joshi 等在 2017 年发布了 TriviaQA[5]，该数据集样本中句法结构较复杂、具有信息冗余，需要复杂推理才能得到正确答案，而且答案不一定是上下文中的连续子文本串。Deepmind 在 2017 年也发布了 NarrativeQA[6]数据集，包含了来自于书本和电影剧本的 1 500 多个完整故事，且问题通常比较复杂，需要通过综合上下文不同位置的信息进行深层推理得到答案。

百度公司主办的 2018 机器阅读理解技术竞赛(2018 NLP Challenge on Machine Reading Comprehension)所使用的数据集为 2017 年提出的 DuReader[1]。该数据集从百度搜索和百度知道中抽取回答或者搜索结果作为上下文，并使用众包的方式生成答案。数据集分为训练集、验证集以及测试集 1 和测试集 2，其中测试集 1 和测试集 2 参赛者无法获得正确答案，提交预测后由百度后台给出成绩。成绩的判别方式是与正确答案之间的平均 Rouge-L，若样本中存在多个正确答案，则以最高的 Rouge-L为准。

相比以前的 MRC 数据集，DuReader 的难度提高了很多，根据 DuReader 论文中的相关数据，以及我们在比赛中对数据集的分析，这个数据集具有以下特点。

(1) 难度高。所有的问题、原文都来源于实际数据(百度搜索引擎数据和百度知道问答社区)，答案是由人类回答的。相比于 SQuAD 这种答案是原文中连续片段的阅读理解数据集，DuReader 数据集中许多样本，只通过抽取原文中连续片段的方式无法得到正确答案甚至是意思接近的答案，要正确回答这些问题只有在理解问题和上下文的基础上，通过生成与 Abstroct 中相矛盾式模型来生成答案。对于其他的答案不来源于上下文的数据集如 MS-MARCO，DuReader 数据集中通过计算人工答案和文档的最小编辑距离来判断两个数据集回答问题的困难度。编辑距离越大，对文档的编辑修改就更多，回答问题的复杂度也就越高，而 DuReader 的平均上下文/答案编辑距离要远远的超过 MS-MARCO。

此外，难度高还表现在，DuReader 中的答案平均长度很长，其答案的平均长度要远远超过 SQuAD 数据集中的样本，后者的答案一般不超过 15 个单词。而 DuReader 的答案长度分布为如图 1 所示的长尾分布，其答案的平均长度是 69。答案长度和编辑距离的数据说明，模型需要预测出足够接近的答案是很困难的。

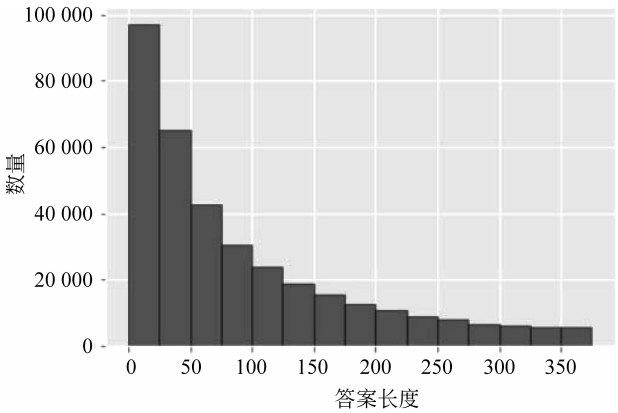

图 1 DuReader 数据集答案长度分布

(2) 问题类型多样。数据集中包含大量的之前很少研究的是非和观点类的样本。之前的数据集中大多问题的类型都是事实型，也就是面向事实给出实体或者描述来回答问题。DuReader 中的数据根据问题是否具有标准答案，分为事实类型和观点类型；根据问题本身的分类，分为实体、描述以及是非类的问题。这种问题类型的多样性，对模型的训练带来困难。如同一个问题和上下文，对观点类的问题来说，有些样本包含的多个答案本身可能互相矛盾(同时包含正向观点和负向观点)，而其他的相似样本可能只包含正向或者负向观点的答案，因此模型对这类问题常常可能给出与正确答案相似度很低的预测。从我们对模型最终性能的分析来看，观点类和是非类问题的效果是最差的，也是影响模型整体性能的关键点。

(3) 数据量大且来源复杂。数据集中每个问题都对应多个答案，总共包含 200k 个问题、1 000k 个

上下文和 420k 个答案，是目前最大的中文 MRC 数据集。这些数据，从来源上来看分为两类，百度知道以及百度搜索，这两类数据的分布有较大的不同。首先，这两类数据的上下文中噪音都比较多，这里的噪音包括但不限于特殊符号、无用的标点重复、俚语流行词和惯用表达以及广告。无用重复和广告类型的噪音对于抽取式模型（从原文中抽取连续字段作为答案）的影响较为严重，因为许多原文中的正确答案中间包含无用的广告链接或者标点重复，极大的降低了模型的 Bleu-4 和 Rouge-L 指标值。特殊符号和俚语流行词类型的噪音，可能对中文分词造成影响导致分词失败，把错误传播到后续的处理过程中。

2　相关研究

近年来，尤其是大规模标注数据集被频繁的发布以及 Word2Vec 等分布式词向量被提出之后，各种不同的基于神经网络的模型在机器阅读理解任务上不断的刷新最好成绩，而其中一个关键的因素就是不同形式注意力机制的引入，为文档和问题的高层次表示的学习提供了有效的双向信息交互。已有的工作包括 Wang 和 Jiang 等在 2016 年提出的 Match-LSTM[7]，Lee 等在 2016 年提出的 RaSoR，Seo 等在 2017 年提出的 BiDAF[8]，Hu 等 2017 年提出的 MnemonicReader[9]，Wang 等在 2017 年提出的 R-net[10] 等。

目前已有的方法，通常网络的第一层是利用词向量得到问题和文档词级别的语义表示，这种语义表示可认为是整个模型唯一的输入语义信息。其中一些已有模型，在分布式词向量的基础上，增加了字向量/字符向量，POS/NER/TF 等特征以及 Allen 实验室在 2018 年提出的上下文向量 ELMo[11]。对语义输入层的改变，多是为了解决输入语义的偏差，比如多义词、拼写错误、不同词态、低频词导致单独的词向量信息无法正确指导模型学习到正确的高层次表示。

在语义表示之后，通常会利用若干层 RNN 来单独建模问题和文档的上下文语义，虽然 Transformer 在机器翻译领域已经证明单独利用注意力机制即可编码序列语义获得不错的效果，但在阅读理解任务上尚未有类似工作。

在单独编码问题和文档的上下文语义表示后，通常会有一到多层的交互层，用于对问题和文档的语义信息通过注意力机制进行交互从而学到基于问题的文档表示和基于上下文的问题表示。一些综述类的文章中，根据注意力机制作用的方向将已有模型分为一维注意力模型和二维注意力模型（单向注意力和双向注意力）。前者通常只利用匹配函数计算上下文中每个单词与问题整体语义的匹配度，并以此决定答案位置；而后者通常通过匹配函数计算上下文中每个单词与问题中每个单词的语义相关度并形成一个二维相似度矩阵，而后利用该矩阵计算高层语义表示。也有工作[9] 利用深度强化学习模拟阅读理解中的推理过程进行信息交互。Huang 等人在 FusionNet 的工作中对阅读理解模型中问题文档语义交互的过程进行了探究。

在学习到问题和文档高层语义表示之后，通常使用一个解码模块得到答案。由于已有的数据集中答案大多为上下文中的连续片段或单个单词，解码模块通常是由 PointerNetwork[12] 衍生而来。

在 2018 机器阅读理解技术竞赛中，主办方百度提供了 Match-LSTM 和 BiDAF 作为基准模型，其中 BiDAF 在 SQuAD 数据集上集成模型的 EM 为 73.74，增加了 Self-Attention 和 ELMo 模块后的 BiDAF 模型达到了 81 的 EM 成绩。

3　模型与方法

3.1　主要问题及解决思路

由于 seq2seq 等生成式模型训练成本更高，且在阅读理解任务中的性能未知，所以此次比赛我们的基本思路是基于百度提供的 BIDAF 的基线模型使用抽取式的模型来预测答案，在抽取式模型中预测答案的偏差可能由下面几个过程共同导致。

生成答案的偏差。在获取分词后的 DuReader 数据集之后，使用生成的 span_answer（即上下文中的连续片段）作为标签来训练抽取式模型，我们对该标签的长度分布（图 2）和与正确答案的相似度进行了分析，来确定抽取式模型的性能上限。

由图 2 可知，生成的答案和正确答案的长度分布（图 1）基本一致，而由表 1 中数据，search 和 zhidao 源中的样本其生成的标签答案和正确答案的平均 match-score 分别是 0.79 和 0.92，已经是一个足够高的性能了。但是，由于生成式的标签对于每个样本只产生一个最接近其中一个正确答案标签，所以对于观点类和是非类问题，这种方式产生的标签大大减少了训练信息，可能导致训练效果下降。

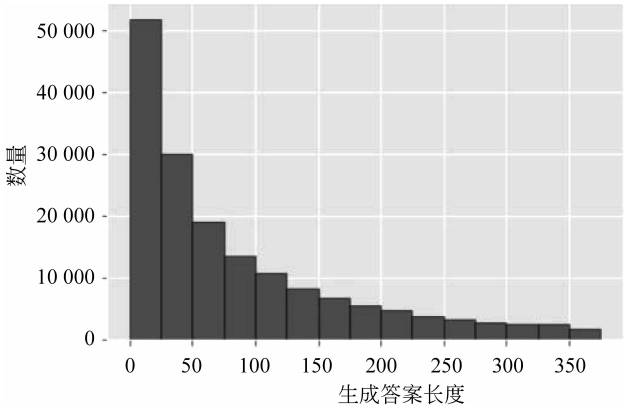

图2 生成答案的长度分布

表1 生成答案与人工答案匹配度

数据源	匹配度
Search	0.79
Zhidao	0.92

上下文选择导致的偏差。由于DuReader中的上下文长度较长、文档数量较多，为了能够让神经网络模型能够处理，默认选择最多5个上下文文档，且每个文档选择长度不超过500的最相关段落(goldenparagraph)。在训练时最相关段落是所有段落中与正确答案recall最小的段落，但是验证和测试时无法这样选择。DuReader在验证和测试时选择了与问题的recall最小的段落作为最相关段落，考虑到语料中的噪音问题，这样的计算方式是不合理的，可能造成所选上下文中并没有足够的信息来得到答案。尤其是对于来源于search的样本，存在部分内容与问题很符合，但是段落中并不包含与答案相关的有用信息的情况。另一种情况是，使用与问题的recall来选择上下文段落，可能选择到的是标题段落，导致所选的上下文段落长度不足因而缺少足够信息，我们分别统计了测试集1中searchdomain和zhidaodomain的所有样本，发现有591个和14个样本中所有上下文段落长度均小于50。Searchdomain的数据噪音更多，受上下文段落选择的影响更大。DuReader论文中的相关数据，也说明正确选择上下文段落能够大大提高模型效果。

针对这一问题，我们对searchdomain的数据，在验证和测试时采用一种启发式的方法选择上下文段落，从而获取更加富集语义的信息，如式(1)所示。

$$P = \sum c_i p_i \tag{1}$$

其中，p_i为从给出段落中抽取的与问题最相关段落，c_i为对应段落所占的权重。为了方便模型处理，根据计算出的答案长度分布，我们将上下文段落大小固定设置为500，通过将段落融合处理，可以让模型自动学习出不同给出段落中的相同特征。我们通过计算问题与给出段落的语义相关度计算出c_i。由于此次比赛给出的段落包含了排序信息，而搜索引擎的排序信息已经包含了对问题与段落的语义相关度信息，因此我们以搜索引擎的排序作为上下文信息蕴含质量的标准，将search数据中的上下文依次填入直到长度超过500。后续的实验证明，通过采用段落融合的方法，searchdomain数据在训练集和验证集上的Rouge-L指标接近，表明了这种上下文选择方式的有效性。

3.2 模型框架

模型采用BiDAF＋Self-Attention的框架，模型名称为Mixed。如图3所示。Mixed模型分为如下几层。

(1) 段落选取层：对于多个给出段落的阅读理解比赛，最佳段落的选择对最终结果的影响很大。

我们测试在给出人工最佳段落时，在测试集合上可以达到67%的分数。最佳段落的选取可以通过计算段落与问题之间的语义相关度来选取，Baseline系统通过计算词共项出现的比例来选取最佳段落。我们经过测试，发现通过这种方式选出的段落缺乏其他给出段落的信息，最终采用了段落融合的方式，增加了段落的上下文信息，提高了结果的Rough-L值，具体可以参考实验部分Dev-Paragraph模块对整体性能的影响。

(2) 模型输入层：模型输入为问题序列$q=\{q_1,\cdots,q_J\}$，以及上下文序列$d=\{d_1,\cdots,d_T\}$。对DuReader中每个问题具有多个上下文文档的，将n个上下文文档在解码层之前单独处理并单独学习其表示，并对多个文档同时进行解码预测答案的位置(答案位置无法跨文档)。

(3) 词向量层：使用不同语料预训练的Glove词向量(见实验小节)，将输入序列中的每个单词映射为词向量$D_i\in R^d$，$Q_j\in R^d$。

(4) 上下文编码层：利用双向LSTM，对问题和答案进行编码，得到每个单词的上下文感知编码$h_i\in R^{2d}$，$u_j\in R^{2d}$。

(5) 交互层：利用c2q和q2c的双向注意力机制，计算出上下文中每个单词的问题感知编码$g_i\in$

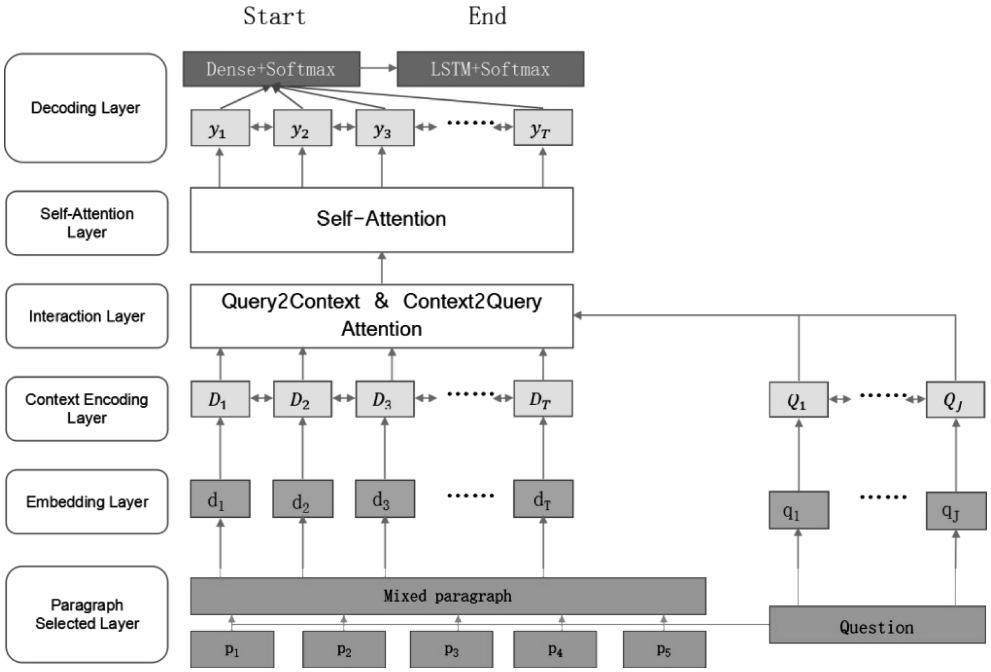

图 3　Mixed 模型结构

R^{8d}，计算过程如 BiDAF 中的 Attention-Flow 层。

(6) 自注意力层：利用自注意力机制，以交互层的输出 $g=\{g_1,\cdots,g_T\}$ 作为 memory 和 input 计算注意力权值 c_t，计算过程如式(2)所示。而后拼接输入和加权输出得到最终输出 $y_i=g_i\oplus c_i$，其中⊕表示向量拼接操作。这一层可以认为是带着问题把上下文再读一遍的过程，在许多阅读理解模型中利用了类似的结构，如式(2)所示。

$$s_j^i = V^T \text{relu}(w_g^j g_j + w_g^i g_i)$$
$$a_j^i = \frac{\exp(s_j^i)}{\sum_{j=1}^{n}\exp(s_j^i)}$$
$$c_i = \sum_{j=1}^{n} s_j^i g_j \tag{2}$$

(7) 解码层：解码层利用 LSTM 获取蕴含问题信息的文档整体表示，而后利用全连接和 Softmax 分类输入答案开始位置，利用 LSTM 和 Softmax 分类输出答案结束位置。

模型损失函数是答案开始和结束预测位置分布和正确位置的负对数似然之和，在训练时反向传播每个 minibatch 样本的平均损失。使用 Adam 优化器进行端到端的学习。在模型推理(inference)时，通过动态规划遍历每个答案开始点和结束点，找到其概率相乘最大的一组作为预测答案。

3.3　其他尝试

在实验中，我们曾经如下几个方向进行了尝试，但由于效果不好或者时间不足的原因，没有增加到最终的模型中，可能在后续的工作中会继续尝试。

(1) 人工特征。由于我们在模型中并未使用字向量，我们为上下文增加了五种人工抽取的特征，希望提高输入信息的表示能力。问题匹配特征用来说明上下文中的当前单词是否在问题中出现过；其他文档词共现特征用来计算本段落中的某一单词在其他段落中出现的次数；问题范围匹配特征以一定长度的窗口计算窗口内文档与问题的编辑距离和 Jaccard 距离。对于问题匹配特征和其他文档词共现特征我们还提取了相应的字级别的特征。但是增加这些特征之后模型的性能并没有明显提升，因此最终提交结果的模型中并没有使用这些人工特征。

ELMo 上下文向量，这是一个我们很希望尝试的方向，但在比赛结束前 Allen 实验室并没有公开 ELMo 的训练代码，因此最终没有增加这一模块。

(2) 答案的 paraphrase。我们通过分析查看训练集和验证集结果发现，抽取式的答案中会有很多与答案无关片段，在蕴含信息相同的情况下降低了预测答案的 Rouge-L 得分，因此可以通过 seq2seq 或者 sequence-tagging 的模型，对答案进行缩写或者改写。

4 实验

4.1 模型训练

在模型训练和推理时，我们对分词后上下文中的特殊字符等进行了筛选，删除了重复标点、URL、乱码、HTML代码等特殊字符。我们使用了DuReader语料和中文维基百科语料上预先训练的GloVe词向量，在预训练词向量时，使用jieba分词。

最终提交的模型是5个模型进行Ensemble后的结果。模型集成的方式是，单模型分别在DuReader训练集上进行训练，并将预测的answer_probs进行平均得到最终预测。

代码基于Tensorflow1.7，并在具有单个TitanXP的服务器上进行训练，单模型训练时占用显存约10.8GB，通常训练7个Epoch至收敛，约需要15小时。

模型训练过程中，使用的超参数设置如表2所示。

表2 超参数设置

超参数	值
Word_embedding_dim	300
Learning_rate	0.001
Dropout_keep_prob	0.7
Batch_size	32
Epochs	10
LSTM_hidden_size	150
Adam_gamma	0.2

由于最终提交的模型需要为Yes/No类型问题提供态度选项，我们通过一个简单的以预测答案词频为特征的分类器来计算答案态度。

4.2 实验结果及分析

我们在实验的每步记录了增加/减少的模块，以及对应的验证集/测试集1的Rouge-L性能，如表3所示，此处的验证集性能是由第三方脚本计算得到，因此可能不够准确。通过对该表的消融分析(AblationAnalysis)能够验证框架中每个模块的作用。从表中可以看出，增加自注意力机制(Self-Attention)、使用预训练的词向量(Pretrained-word-embeddings)以及验证和测试时改变选择相关段落的方式(+Dev-Paragraph)是对模型性能提高影响最大的几个模块。此外，由于搜索引擎给出的网络文本信息中存在的大量噪音(表4)，因此除了在分词前进行预处理，基于正则化规则的方式删除上下文和预测答案中的特殊符号和文本也能有效提高性能。

表3 Ablation Analysis

模型	验证集	测试集
Baseline	37.3	43.5
+Self-Att	39.2	45.1
+Pretrained-word-emb	40.7	46.7
+AttitudePred@Yes/No	41.0	47.3
+Dev-Paragraph	43.4	50.7
+Bilinear-Att	44.0	/
+Pretrained-wiki-word-embeddings	44.7	52.8
+Remove-answer-special-chars	45.6	/
Mixed(Ensemble)	46.7	54.5

表4 文本噪声示例

	示例
重复链接	`jingyan.baidu.com`
字符乱码	`f牌a`
标识符	`\r`,`\a`,`\u`
分词错误	`法爱情`,`贞高绝俗`
无意义英文字母/数字	`dttn`,`soxsok`,`carranza`,`33621105402`

我们对最终单模型在验证集上预测span_answer的准确匹配(Exact Match，EM)比率进行了测试，在zhidao数据上准确率达到了11.4%，在search数据上准确率达到了5.2%，总体准确率为8.1%。该数据说明在DuReader数据集难度下，抽取式的模型本身性能还不足。

最终Mixed(Ensemble)模型在验证集、测试集1和测试集2上的综合性能，以及在测试集2上每个类别问题的性能如表5所示(验证集性能计算由第三方脚本提供)。

表5 模型性能

问题类型	验证集	测试集
Search	41.9	48.7
Zhidao	51.5	59.8
Entity	50.7	55.5
Description	46.0	54.0
Yes/No	36.8	50.9
Total	46.7	54.2

4.3 模型性能及可扩展性分析

本文最终的 Ensemble 模型在高性能计算平台上采用分布式训练，训练时间需要 24 小时。随着互联网文本数据的不断增长，人们对机器阅读能力期望的不断提升，我们也需要设计更大复杂的网络模型。得益于神经网络的高度可并行性，通过高性能计算技术能够大幅缩短训练时间，提升迭代效率。我们针对数据特点进行压缩提升读取速度，采用分布式并行化处理提升模型运行效率，采用成熟的软件接口来增加可移植性。随着高性能计算技术的不断发展，我们能够构建更加复杂的网络模型，不断提升机器阅读理解能力。

我们在设计模型时，考虑了模型在不同应用场景下的可扩展性。如本文的模型还能方便的扩展到其他阅读理解场景中，如医疗数据文本分析处理、社交网络问答数据处理等场景。

5 总结

我们对 DuReader 数据集进行了分析，并以 BiDAF 模型为基础不断改进，最终实现了一个在 DuReader 上拥有不错性能的模型 Mixed Model，在最终测试集 2 的排行榜上 Rouge-L 和 Bleu-4 分别位列第 9 名和第 8 名。受限于时间和模型实现速度，在比赛结束的时候还有很多希望尝试的方向没有进行尝试，比如上下文向量、抽取式答案的改写以及生成式的模型，之后会向这些方向探索。

参考文献

[1] He W, et al. DuReader: a Chinese Machine Reading Comprehension Dataset from Real-world Applications [DB/OL]. arXiv preprint arXiv:1711.05073, 2018.

[2] Hermann K M, et al. Teaching machines to read and comprehend[C]//Proceedings of Advances in Neural Information Processing Systems, 2015:1693-1701.

[3] Hill F, et al. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations[C]//Proceedings of Under Review of ICLR, July 2016: 1 13.

[4] Rajpurkar P, et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016:2383-2392.

[5] Joshi M, et al. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension [DB/OL]. arXiv preprint arXiv:1705.03551, 2017.

[6] Kočisky, Tomáš, et al. The narrativeqa reading comprehension challenge[J]. Transactions of the Association of Computational Linguistics, 2018, 6: 317-328.

[7] Wang S, Jiang J. Machine comprehension using match-lstm and answer pointer[DB/OL]. arXiv preprint arXiv:1608.07905.2016.

[8] Seo M, et al. Bi-Directional Attention Flow for Machine Comprehension[C]//Proceedings of the International Conference on Learning Representations, 2017: 1-12.

[9] Hu M, Peng Y, Qiu X. Mnemonic Reader for Machine Comprehension[DB/OL], arXiv preprint arXiv: 1705.02798.2017.

[10] Wang W, et al. Gated self－matching networks for reading comprehension and question answering[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics :189-198.

[11] Peters M E, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[12] Vinyals O, Fortunato M, Jaitly N. Pointer networks [C]//Proceedings of the International Conference on Neural Information Processing Systems. MIT Press, 2015.

张浩宇(1993—)，博士研究生，主要研究领域为基于知识图谱的问答，机器阅读理解。
E-mail: zhanghaoyu10@nudt.edu.cn

李真真(1991—)，博士研究生，主要研究领域为信息抽取，知识图谱构建。
E-mail: lizhenzhen14@nudt.edu.cn

张鹏飞(1991—)，博士研究生，主要研究领域为基于网络文本的问答，机器阅读理解。
E-mail: zhangpengfei09@nudt.edu.cn