# 通过多文档精排与融合的开放域问答任务增强实现

李 博,朱天佑,刘俊健,吕宏伟,陈振宇

(国家电网有限公司 大数据中心,北京 100053)

**摘 要:** 开放域问答(OpenQA)是自然语言处理中的一项具有挑战性的任务,传统的机器学习和深度学习技术通常用于从原始语料库中检索与问题相关的候选文档片段以进行答案提取。然而,当前方法检索的候选文档片段往往包含大量的噪声以及与问题无关的信息,并且主流的OpenQA模型在准确响应需要多个文档片段作为相关证据的问题方面存在不足。鉴于此,提出通过多文档精排与融合增强开放域问答的方法(RFMD),该方法在检索阶段设计了基于Transformer的文档精排模块,以减少候选文档中的噪声信息;在阅读理解阶段,RFMD采用以文本生成为中心的问答模块,通过构建跨文档片段的全局注意力机制,整合多个相关文档片段的信息,准确回答需要多个文档片段作为支持证据的问题。RFMD在NaturalQuestions和TriviaQA数据集上的EM得分分别达到45.8%和63.4%,验证了该模型在OpenQA任务中的有效性和优越性。

**关键词:** 开放域问答;预训练模型;生成模型;相似度分数;Prompt设计

**DOI:** 10. 11907/rjdk. 231986 开放科学(资源服务)标识码(OSID):

**中图分类号:** TP301.6 **文献标识码:** A **文章编号:** 1672-7800(2024)009-0082-08

# Open-Domain Question Answering Task Enhanced by Multiple Documents Refinement and Fusion

LI Bo, ZHU Tianyou, LIU Junjian, LYU Hongwei, CHEN Zhenyu

(*Big Data Center, State Grid Corporation of China, Beijing* 100053, *China*)

**Abstract:** Open-domain question answering (OpenQA) is a challenging task in natural language processing, the conventional machine learning and deep learning techniques are commonly used to retrieve many candidate document fragments related to the question from the raw corpus for answer extraction. However, the candidate document fragments retrieved by current methods tend to include considerable noise and irrelevant information to the question, and the previous OpenQA model falls short in accurately responding to questions that necessitate multiple document fragments as correlative evidence. Therefore, this paper proposes an open-domain question answering method based on refinement and fusion of multiple documents(RFMD). Specifically, RFMD designs a Transformer-based document refinement module during the retrieval stage to reduce noise information in the candidate documents. In the reading comprehension stage, RFMD employs a text generation-focused question answering module. By constructing a global attention mechanism across document fragments, it integrates information from multiple relevant document fragments to accurately answer questions that require multiple document fragments as supporting evidence. RFMD achieves EM scores of 45.8% and 63.4% on the NaturalQuestions and TriviaQA datasets respectively, verifying the effectiveness and superiority of the model in OpenQA tasks.

**Key Words:** open-domain question answering; pre-training model; generative model; similarity score; Prompt design

**收稿日期:** 2023-09-19

**扫描二维码阅读全文:**

**基金项目:** 国家电网有限公司大数据中心自建科技项目(SGSJ0000YFJS2200047)

**作者简介:** 李博(1982-),男,硕士,国家电网有限公司大数据中心高级工程师,研究方向为图像智能识别、自然语言处理技术、知识图谱技术及电力专业应用;朱天佑(1994-),男,博士,国家电网有限公司大数据中心工程师,研究方向为电力大数据、人工智能;刘俊健(1994-),男,硕士,国家电网有限公司大数据中心助理工程师,研究方向为电力大数据、人工智能;吕宏伟(1985-),男,硕士,国家电网有限公司大数据中心工程师,研究方向为电力大数据、人工智能;陈振宇(1985-),男,博士,国家电网有限公司大数据中心高级工程师,研究方向为电力大数据、人工智能。本文通讯作者:朱天佑。

## 0　引言

开放域问答(Open-domain Question Answering, Open-QA)旨在通过检索大规模非结构化语料库(如维基百科)回答自然语言形式的问题[1]。开放域问答可以应用于多种场景,包括搜索引擎[2]、虚拟助手[3]和智能客服[4]等。目前,主流的开放域问答方法是通过对大规模非结构化语料库进行信息检索,并建立机器阅读理解模型回答自然语言问题。与 GPT 系列的大型语言模型相比,其具备参数量小、易于部署、主要研究领域更专注等优点[5]。大型语言模型主要通过训练具有大量参数的神经网络而从大量数据中获取知识,由于依赖大量参数和训练数据,故难以在特定领域局部部署和应用。相比之下,针对于开放域问答领域的模型可以基于语料库实时更新模型参数,更适用于回答新知识、新领域的问题,因此具有重要研究意义。

## 1　相关工作

在主流的开放域问答任务研究中,解决问题通常分为两个阶段:信息检索和阅读理解。信息检索,即选择与问题相关的候选文档;阅读理解,即从候选文档中推理出问题答案。信息检索阶段旨在从非结构化知识库(原始语料)中抽取与问题最相关的候选文档,从而缩小问题答案的搜索空间。已有研究中采用的信息检索方法包括传统机器学习和深度学习方法。传统机器学习方法主要采用向量空间模型[6]和概率模型[7]从大型语料库中选择相关文档。在向量空间模型中,知识库中的原始语料文档和查询问题被编码为稀疏向量表示,其中每个向量的维度对应不同的术语。在概率模型中,单词之间的概率关系被集成到模型中。然而,这些检索方法主要基于稀疏表示检索相关候选文档,而稀疏表示的局限性是模型可能会忽略语义相关但与问题的词汇重叠度较低的文档[8]。近年来,深度学习方法在信息检索中得到了广泛应用,通过使用密集向量表征文档和查询问题,以解决传统方法中稀疏表示和词汇重叠度低的问题。例如,Huang 等[9]开发了一系列具有深层结构的潜在语义模型,将问题和知识库原始语料文档映射到一个公共低维空间中。Guu 等[10]通过无监督的方式预训练知识检索器,并通过微调开放域问答的任务证明其有效性。主流研究利用信息检索技术对知识库原始语料进行筛选,抽取出与查询问题相关的候选文档,并联合阅读理解模型进行答案抽取。然而,之前的工作没有对检索到的候选文档进行精细化筛选,导致训练过程中包含了大量噪声和冗余信息,这严重干扰了模型抽取答案的性能以及训练所使用的资源,因而有必要对检索出的候选文档进行重新筛选和精确排序[5]。

阅读理解是开放域问答的另一核心任务,其目标是从候选文档中推理出问题答案,该阶段的性能取决于问题的复杂性、候选文档质量和具体的阅读理解方法。传统阅读理解方法通常将查询问题类型格式化,并且问题答案通常来自于文档中的某个实体或名词短语。因此,传统的阅读理解方法高度依赖于命名实体识别技术(Named Entity Recognition, NER),且通常采用文本、单词短语或句法匹配等方法获取不同类型问题的答案[11]。传统方法的局限性在于其对问题类型有极大的限制且答案类型单调,无法回答复杂多跳的自然语言问题[4]。基于深度学习的方法通常采用从候选文档中预测问题答案跨度的方案,即将答案在文档中的开始位置和结束位置中的跨度内容作为问题的最终答案。例如,Chen 等[12]将查询问题和候选文档输入双向长短期记忆网络中,模型预测出答案范围;Karpukhin 等[13]使用 Bert 计算包含查询答案的段落及其 Token 跨度,并选择概率最高的跨度作为最终答案;Roberts 等[14]提出在不使用额外知识的情况下,利用语言模型进行开放域问答,然而此方法无法获取最新的外部知识,对于新领域类型问题无法作出准确回答。主流的阅读理解方法主要采用基于实体跨度提取答案范围的方法,然而此类方法无法处理问题答案存在于多个跨度和多个文档中的情况,导致模型性能有限。

综上所述,在开放域问答任务中,存在两个问题:①检索到的文档处理不够精细,噪声信息较多;②在面对复杂问题时,主流研究方法难以聚合多个文档相关证据以回答问题。为解决上述问题,提出了基于多文档精排与融合的开放域问答方法(Refinement and Fusion of Multiple Documents, RFDM)。与前人研究模型从候选文档中粗略地提取答案跨度的方法不同,该模型首先采用传统的 TF-IDF 方法对原始语料库进行检索以获取候选文档,之后采用语言模型对问题和文档进行嵌入表征,并基于向量相似性筛选出与问题语义相关但重叠度低的文档,这一过程可以减少问题答案的搜索空间兵降低候选语料的噪声。之后,基于问题和精筛后的候选语料库设计模型 Prompt,使得后续采用的生成模型能最大限度地理解 Prompt 不同字段的含义。最后,基于生成模型融合来自多个文档的证据信息,以解决问题形式复杂且问题答案证据存在于多个文档中的情况。如图 1 所示,模型首先通过精排模块从候选语料库中筛选低噪声、高相关的文档,并将其与问题和相似度得分组合构建模型 Prompt;然后,送入生成模型,生成模型通过融合多片段内容生成问题答案。

## 2　方法架构

基于多文档精排与融合的开放域问答模型(RFMD)架构如图 2 所示。RFMD 模型由 4 个模块组成,分别为原始语料检索模块、候选文档精排模块、Prompt 设计模块和基于生成模型的问答模块。

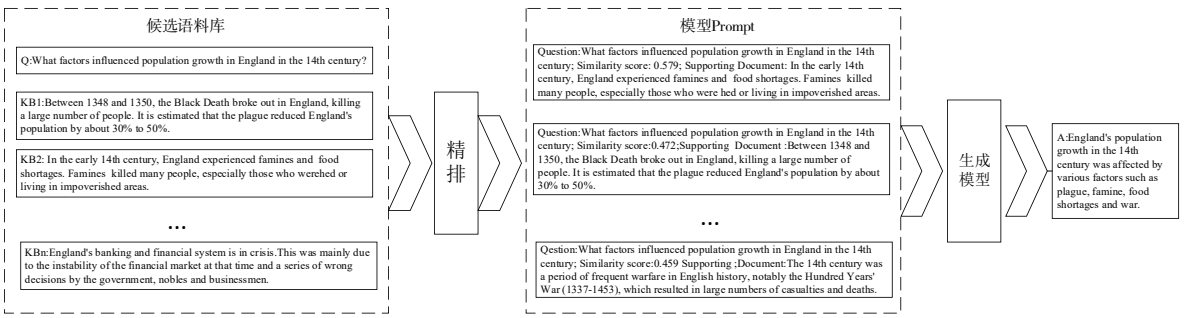



**Fig. 1 Document refinement and multi-fragment fusion case**

**图1 文档精排与多片段融合案例**

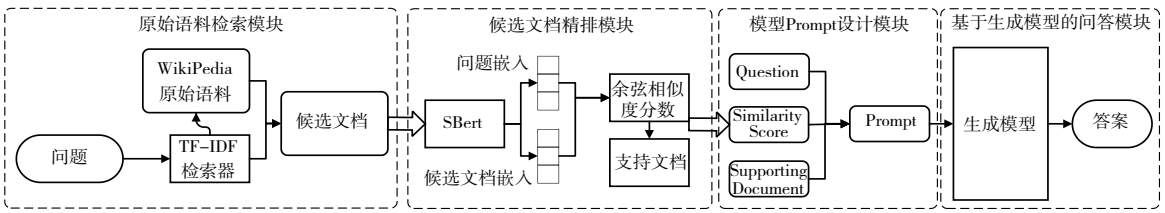



**Fig. 2 Architecture of open domain question answering method based on multi-document refinement and fusion**

**图2 基于多文档精排与融合的开放域问答方法架构**

原始语料检索模块负责从大规模知识库原始语料中抽取与问题相关的候选文档;候选文档精排模块负责从候选文档中提取与问题高度相关的支持文档;Prompt设计模块负责将问题与候选文档等内容设计为生成模型易于理解的输入形式;基于生成模型的问答模块利用生成模型以及模型Prompt从多个文档中提取问题相关证据信息并生成答案。

RFMD模型的研究重点在于对检索到的候选文档进行精排(精排后的文本被称为支持文档)并设计Prompt,采用生成模型生成问题答案的方式回答问题。

### 2.1 任务定义

给定一组问题$Q=\{q_1,\cdots,q_n\}$,表示共有$n$个问题需要被回答。原始语料文档$D=\{d_1,\cdots d_m\}$,$m$表示原始语料文档的数量,以及问题答案集合$A=\{a_1,\cdots,a_n\}$,$a_i$表示问题$q_i$的答案。

问题$q_i$的候选文档集被表征为$C^i=\{c_1^i,\cdots,c_k^i\}$,$k$被表征为每个问题被检索到$k$个与问题相关的候选文档,$C^i$表示第$i$个问题的候选文档。问题$q_i$的支持文档集被表征为$S^i=\{s_1^i,\cdots,s_u^i\}$,$u$表示问题$q_i$的支持文档总数量,$S^i$表示第$i$个问题的支持文档。问题$q_i$与支持文档的相似度得分被表征为$G^i=\{g_1^i,\cdots,g_u^i\}$,$G^i$表示第$i$个问题与其支持文档的相似度分数。

### 2.2 原始语料检索模块

在原始语料检索模块,首先基于Unicode文本标准化[15]处理知识库原始语料文档集和输入的问题,对HTML标签、停用词(如'a','an','the'等)和特殊符号(如'*','$'等)进行过滤。之后,使用式(1)的TF-IDF[16](词频—逆文档频率)算法计算查询问题中每个词汇在文档集中的重要性。首先通过分词将文档转化为单词列表,之后计算每个单词在文档中的出现频率,$TF(t)$表示单词$t$在每个文档$d_i$中出现的频率,如式(1)所示。

$$TF(t)=\frac{\text{单词}t\text{在文档中出现的次数}}{\text{文档}\mathrm{d_i}\text{中所有单词的总数}},\tag{1}$$

接着,计算每个单词$t$在文档集中出现的文档频率$IDF(t)$,见式(2),这里加1是为避免出现分母为0的情况。

$$IDF(t)=\log\frac{\text{文档集}D\text{的总数}}{\text{包含单词}t\text{的文档数}+1}\tag{2}$$

最后,基于TF($t$)和IDF($t$)计算每个单词的*TF-IDF*权重,如式(3)所示。

$$TF-IDF(t,d,D)=TF(t,d)*IDF(t)\tag{3}$$

对于每个文档$d_i$,将问题中每个*TF-IDF*的权重相加获得每个文档的最终权重。最后,将文档按其总体权重排序,并返回最相关的文档作为候选文档。

经过上述处理过程,该模块可以高效且快速地从原始语料$D=\{d_1,\cdots d_m\}$中为每条问题$q_i$检索出$top-k$条候选文档$C=\{c_1^i,\cdots,c_k^i\}$。

### 2.3 候选文档精排模块

尽管使用*TF-IDF*可以快速从原始语料中检索和获取候选文档,但是它们缺乏对自然语言详细处理的功能(如分词、词干提取和命名实体识别等)。因此,在精确筛选候选文档时,需要采取措施去除包含大量噪声和与问题相关度低的文档。RFMD模型使用了预训练语言模型Sentence-Bert[17]对候选文档进行精排。

候选文档精排模块将来自原始语料检索模块的候选文档$C=\{c_1^i,\cdots,c_k^i\}$传递给预训练语言模型Sentence-Bert(SBert)[17]。SBert模型是一种基于深度学习的文本嵌入技术,适用于文本相似度计算任务。具体而言,与传统的Word2vec[18-19]和Doc2vec[20]不同,SBert通过采用多层Transformer[21]编码和平均池化的方法对查询问题$q_i$、候选

文档 $C^i=\{c_1^i,\cdots,c_k^i\}$ 进行编码,并将每个查询问题和文档的编码压缩为固定长度的向量嵌入,如式(4)所示。

$$\hat{q}_i = AvgPool\left(SBert\{q_i^1,q_i^2,\cdots,q_i^n\}\right) \tag{4}$$
$$\hat{c}_k^i = AvgPool\left(SBert\{c_k^{i1},c_k^{i2},\cdots,c_k^{in}\}\right)$$

其中:$q_i^n$ 表示查询问题的第 $n$ 个 Token,$\hat{q}_i$ 表示查询问题的嵌入向量;$c_k^{in}$ 表示第 $i$ 个问题的第 $k$ 个候选文档的第 $n$ 个 Token,$\hat{c}_k^i$ 表示第 K 个文档的嵌入表征,候选文档集的表征为 $c_k^i=\{\hat{c}_1^i,\cdots,\hat{c}_k^i\}$。

获得查询 $q_i$ 和候选文档 $C^i=\{c_1^i,\cdots,c_k^i\}$ 的向量表征后,使用余弦相似度计算查询问题嵌入 $\hat{q}_i$ 和每个文档嵌入 $\hat{c}_k^i$ 的相似度得分 $g_k^i$,计算公式如式(5)所示。

$$g_k^i = \frac{q_i \cdot c_k^i}{\left|q_i\right|\left|c_k^i\right|} \tag{5}$$

该余弦相似度的取值范围为[−1, 1],分数越接近于 1 表示查询和文档相似度越高,分数越接近 −1 表示相似度越低。之后根据相似度分数,即 $G^i=\{g_1^i,\cdots,g_u^i,\ ...,\ g_k^i\}$,对候选文档进行精排,选取得分最高的前 $u$ 条文档作为查询的支持文档,表示为 $S^i=\{s_1^i,\cdots,s_u^i\}$。

### 2.4　Prompt 设计模块

该模块主要设计生成模型所需的 Prompt 输入。该生成模型将来自候选文档精排模块的支持文档 $S^i=\{s_1^i,\cdots,s_u^i\}$ 以及相似度得分 $G^i=\{g_1^i,\cdots,g_u^i\}$ 与问题 $q_i$ 进行连接,构建 Prompt,作为下一模块的输入。拼接格式为:"Question: … Similarity Score: … Supporting Document: …",Prompt 案例如式(6)所示。

$$P_u^i = Concat(Qustion{:}\ q_i;\ Similarity\ Score{:}\ g_u^i;\ Supporting\ Document{:}\ s_u^i) \tag{6}$$

其中,$i$ 表示第 $i$ 个问题,$u$ 表示第 $u$ 个支持文档,$P_u^i \in \mathbb{R}^k$,$k$ 为 $P_u^i$ 的长度。按照此种格式输入信息是因为生成模型是针对于文本到文本的任务而设计。将问题和支持文档拼接到一起,基于生成模型的问答模块可以同时获得问题信息以及支持文档的上下文信息。同时,将相似度得分进行拼接是为了指导生成模型从不同支持文档中提取答案的优先程度,即相似度得分越高,该支持文档包含正确答案的可能性越大。因此,在融合该文档相关内容时,生成模型将为高得分文档分配更高的融合权重。此外,Prompt 将问题与每个支持文档分别相连接的原因是因为生成模型对输入字符长度有限制。当支持文档字符较长时,如果将所有支持文档都直接连接到问题上,生成模型无法从足够多的文档中提取答案,进而可能导致生成模型生成的答案内容不完整。

### 2.5　基于生成模型的问答模块

该模块整体结构如图 3 所示。首先将 Prompt 设计模块的 $P_u^i$ 送入生成模型的多层 Transformer 编码器端进行编码,以获得输入 Prompt 的编码表示 $h_u^i$,如式(7)所示。

$$h_u^i = \text{Transformer} - Encoder(P_u^i) \tag{7}$$

其中,$h_u^i \in \mathbb{R}^{k\times d}$ 为每条 Prompt 的编码嵌入,$k$ 为 $P_u^i$ 的长度。之后,将获得的编码信息进行连接,形成全局特征矩阵 $H^i$,如式(8)所示。

$$H^i = Concat\{h_1^i, h_2^i, \cdots, h_u^i\} \tag{8}$$

其中,$H^i \in \mathbb{R}^{u\times k\times d}$。最后,利用生成模型的多层 Transformer 解码器端参考相似度分数,从多个支持文档中聚合相关证据信息,见式(9),并通过自注意力交叉和多头注意力机制提取多个文档之前的潜在相关特征,

$$\begin{aligned} Q^i &= W_Q Y^i \\ K^i &= W_k H^i \\ V^i &= W_v H^i \end{aligned} \tag{9}$$

其中,$Y^i$ 为上一层自注意力机制的结果,$W_Q$、$W_K$ 和 $W_V$ 为当前模块的可训练参数。问题答案的输出是通过将注意力权重作用于每个 Token 词上,生成问题答案。如式(10)所示。

$$\alpha_{i,j} = Q^i K^i \tag{10}$$
$$\tilde{\alpha}_{i,j} = \frac{\alpha_{i,j}}{\sum \exp(\alpha_{i,j})}$$
$$O_i = W_O \sum \tilde{\alpha}_{i,j} V_j^i$$

其中,$j$ 表示第 $j$ 个单词 Token。之后,将输出 $O_i$ 自回归解码生成问题 $q_i$ 的答案 $a_i$。

## 3　实验与结果分析

### 3.1　数据集与评价指标

本实验采用 NaturalQuestions(NQ)[22] 和 Trivia QA[23] 数据集验证 RFMD 模型的有效性。NQ 数据集由 Google AI Language 团队创建,其中包含了从 Google 搜索引擎中收集的真实用户查询和答案,这些答案来源(如维基百科、Freebase 等)可靠,并且经过人工审核和标注。实验中采用前人工作中经过精细化处理的 NQ 数据集,即丢弃超过 5 个

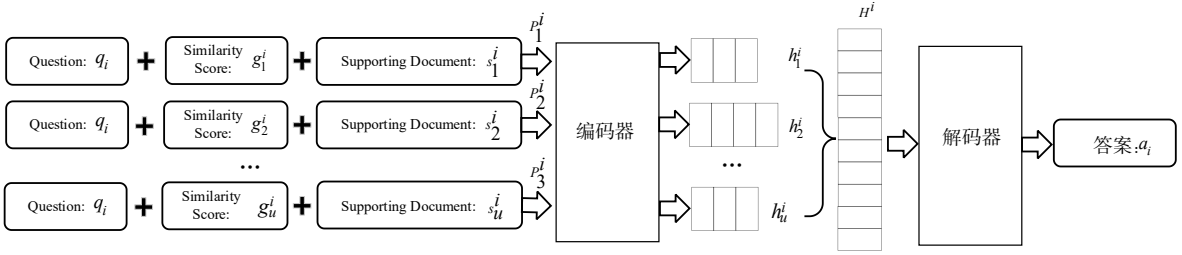



**Fig. 3　Question answer generation module structure**

**图 3　问题答案生成模块结构**

标记答案而获取的，数据集详情如表1所示[24]。Trivia QA数据集由卡耐基梅隆大学和麻省理工大学研究人员创建，其中的问题涵盖了广泛的主题，包括历史、科学、文化、娱乐等，答案来自于维基百科页面中的段落，并经过了人工标注。实验中使用经过过滤的Trivia QA数据集，详情如表1所示。与文献[10]相同，本实验采用的知识库原始语料文档来自于2018年12月20日的维基百科转储文档，该文档是由运营维基百科的维基媒体基金会提供的例行数据转储。

**Table 1 Details of the NQ and TriviaQA datasets**

**表1 NQ和TriviaQA数据集详情**

| 数据集 | 训练集 | 验证集 | 测试集 | 问题案例 | 答案案例 |
|---|---|---|---|---|---|
| NQ | 79168 | 8757 | 3610 | when is the next dead pool? | May 18, 2018 |
| Trivia QA | 78785 | 8837 | 11313 | The VS-300 was a type of what? | Helicopter |

实验中采用与前人研究相同的评价指标即标准精确匹配度(Exact Match, EM)，计算公式如(11)所示。

$$EM(q,d)=\begin{cases}1, & q=d\\ 0, & otherwise\end{cases} \tag{11}$$

其中，$q$表示生成的答案，$d$表示给定的问题标准答案，若完全匹配，则EM得分为1，否则得分为0。正如文献[25]中所述，预测答案与经过规范化后(删除冠词、标点符号和重复的空格)的可接受答案列表中的任何答案匹配，则认为答案预测正确。

### 3.2 实验环境及参数设置

本实验使用32G B显存的Tesla V100 GPU训练RFMD模型。基于生成模型的问答模块可采用T5[26]、BART[27]等作为生成模型，本文采用FLAN-T5[27]作为实验的生成模型，该模型采用类比推理和对抗样本生成等帮助RFMD模型作出决策。FLAN-T5能够将自然语言处理任务转化为文本到文本的预测问题，并且性能表现出色。FLAN-T5的强泛化能力使得其适用于开放域问答任务。本实验使用的预训练语言模型SBert和生成模型FLAN-T5来自于HuggingFace Transformers库，两个模型大小均采用Base。相似度分数使用来自Sklearn库的cosine_similarity函数计算。实验中使用Adam优化器对模型进行了优化，学习率设置为$10^{-4}$，Dropout率设置为0.1，候选文档个数设置为100，支持文档集总数设置为50，模型训练每条拼接文档长度设置为256。实验使用训练集微调RFMD模型，并保存在验证集上获得最高EM得分的模型参数，然后在测试集上进行测试，验证模型效果。

### 3.3 模型比较

为验证本文模型有效性，将所提出的RFMD模型的实验结果与Path Retriever[22]模型、Hard EM[28]模型、ORQA[24]模型、REALM[10]模型、BM25+BERT[24]模型和RECONSIDER[29]模型等6种基线模型进行比较。其中，Path Retriever模型、ORQA模型和REALM模型的主要工作是对问题文档检索模块进行处理；Hard EM模型、BM25+BERT和RECONSIDER模型的主要工作是对阅读理解模块进行处理。6种基线模型具体介绍如下：

(1)Path Retriever模型。该模型采用基于图的循环检索方法，其中问题和文档表示节点，其关系表示为边，通过维基百科图检索和注意力机制的推理路径选择文档片段回答开放域问题。

(2)Hard EM模型。该模型采用基于预先计算的特定任务的潜在学习变量预测，从预先计算的答案集合中预测最优可能的答案，其将该类任务转换为离散化的可能接近问题答案的方案。

(3)ORQA模型。该模型展示了从问答字符串中联合学习检索器和阅读理解器，通过无监督反向完形填空任务预训练检索器，降低检索后质量较差的样本文本数量。

(4)REALM模型。该模型对潜在的知识检索器进行增强语言模型预训练，其通过引入大量外部世界知识和填补空缺任务以训练检索器，以及建立外部语料库与问题所需知识的显示连接以回答问题。

(5)BM25+BERT模型。该模型是基于BM25进行文档检索、基于BERT生成文档跨度的开放域问答方法，即基于开始和结束位置获取问题答案。

(6)RECONSIDER模型。该模型提出了一种基于重新排序阅读理解模型提取出答案跨度的方法，其从段落内答案的跨度注释对较小的候选集执行以跨度为中心的重新排序。

### 3.4 实验结果

将上述6种基线模型与本文RFMD模型进行比较，基线模型在两个数据集上的结果均来自官方结果，'-'表示官方未给出在该数据集上的测试实验。具体效果比较如表2所示。

**Table 2 Comparison of experimental effects of different models on NQ and Trivia QA datasets**

**表2 不同模型在NQ和Trivia QA数据集上的实验效果比较**

| 模型 | EM评估指标(%) | |
|---|---|---|
| | NQ | Trivia QA |
| Path Retriever模型 | 31.7 | - |
| Hard EM模型 | 28.8 | 50.9 |
| ORQA模型 | 31.3 | 45.1 |
| REALM模型 | 40.4 | - |
| BM25+BERT模型 | 26.5 | 47.1 |
| RECONSIDER模型 | 45.5 | 61.7 |
| 去除精排模块 | 44.6 | 60.2 |
| RFMD模型 | **45.8** | **63.4** |

本文实验方法在NQ数据集上获得了45.8%的EM得分，在Trivia QA数据集获得了63.4%的EM得分，均优于对比基线模型。可以看出，相较于对检索模块的改进模型，即Path Retriever模型、ORQA模型和REALM模型，在NQ数据集上，本文RFMD模型分别提升了14.1，14.5和5.4个百分点；在Trivia QA数据集上，本文模型相对于基线模型

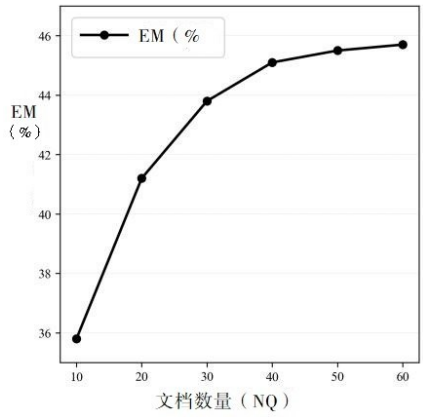
ORQA 提升了 18.3 个百分点,该结果展示了本文所提出的候选文档精排模块的有效性,即与其他检索或检索预训练模型相比,本文 RFMD 模型不需要对检索模块进行预训练,直接通过粗选和精排即可实现对原始语料集的有效筛选。相较于对阅读理解模块的改进,即 Hard EM 模型、BM25+BERT 模型和 RECONSIDER 模型,本文 RFMD 模型在 NQ 数据集上的 EM 得分分别提升了 17.0、19.3 和 0.3 个百分点,在 Trivia QA 数据集上分别提升了 12.5、16.3 和 1.7 个百分点,这表明了本文模型对 Prompt 设计和采用生成模型生成答案的有效性,即与从文档中提取候选片段,以及对候选片段排序等方法相比,本文模型采用生成方式融合多文档片段内容实现了对问题答案的有效生成。

此外,实验中将去除精排模块的 RFMD 模型与完整的 RFMD 模型进行了对比,在两个数据集上的性能分别相差 1.2 和 3.2 个百分点,验证了本文所提出的精排模块的有效性。所有基线模型中只有 RECONSIDER 模型采用精排的方式以帮助提升性能,与 RFMD 模型将检索的候选文档进行精排的方式不同,RECONSIDER 模型是将提取的答案跨度进行精排,RFMD 模型的性能效果优于 RECONSIDER 模型,在两个数据集上分别提升了 0.3 和 1.7 个百分点。实验中,其良好性能的获得得益于对候选文档的精确排序以及采用 Prompt 指导生成模型聚合多段落的相关证据生成问题的答案处理。

### 3.5　候选文档精排模块消融

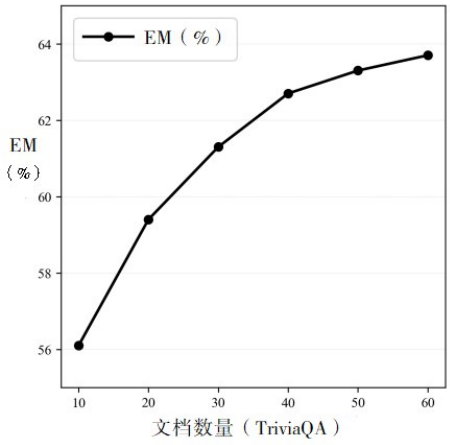
为了研究支持文档数量对模型性能的影响,本文对候选文档精排模块的支持文档数量进行了消融实验,即基于支持文档与问题相似度得分,选取得分最高的前不同数量的文档进行消融实验,实验结果如图 4 所示。

此消融实验设置的支持文档数量参数为[10,20,30,40,50,60]。从图 4 可以看到,在 NQ 和 Trivia QA 数据集上,随着问题支持文档数量的提升,RFMD 模型的性能也得到了较大提升。支持文档数量设置为 60 相对于设置为 10,RFMD 模型的 EM 得分在 NQ 数据集上提升了 10.2 个百分点,在 Trivia QA 数据集上提升了 7.4 个百分点。通过图 4 中两个数据集随着支持文档数量的增长情况可以总结出,在 10~20 和 20~30 的支持文档范围内,RFMD 模型的 EM 得分增长率较高。在支持文档数量为 50~60 时,RFMD 模型的 EM 得分增长率变低。随着支持文档数量的增加,RFMD 模型性能的增长率变低,这说明了对文档数量进行精排的重要性。其中增长率变低,是因为文档与问题相似度得分变低所导致,相似度得分低导致提取出的有用信息变得相对较少。因此,综合模型实验性能、训练时间消耗和内存占用,本文将支持文档数量超参数设置为 50。总体而言,RFMD 模型随着支持文档数量的提升,其 EM 得分同样得到提升,这进一步证明了实验中使用的 FLAN-T5 这种序列到序列的预训练语言模型在应对开放域问答问题上的有效性。




（a）　Performance of different document counts (NQ)

（a）　不同文档数量的性能(NQ)




（b）　Performance of different document counts (TriviaQA)

（b）　不同文档数量的性能(TriviaQA)

**Fig. 4　The effect of different number of question supporting documents on the final result**

**图 4　问题支持文档数量不同对最终结果的影响**

### 3.6　Prompt 设计消融

为验证本文所提出的 Prompt 设计模块的有效性,针对 Prompt 不同提示词进行消融实验,即分别删除“Similarity score”和“Supporting Document”,以及将“Question”和多个“Supporting Document”连接的组合方式,消融实验结果如表 3 所示。其中,W/O 表示 Prompt 未包含该部分。通过表 3 实验结果,可以验证本文提出的 Prompt 的有效性。其中,去除 Document 部分对 RFMD 模型效果影响最大,这验证了外部知识对于 RFMD 模型回答问题的重要性。

同时,去除“Similarity score”部分也对模型性能带来了

**Table 3　Ablation experiment of prompt model**

**表 3　模型 Prompt 消融实验**

| 模型 | EM 评估指标/% | |
|---|---|---|
| | NQ | TriviaQA |
| W/O Similarity score | 45.1 | 62.7 |
| W/O Supporting Document | 23.3 | 31.2 |
| Question + Supporting Document * u | 37.5 | 40.2 |
| RFMD 模型 | **45.8** | **63.4** |

影响，证明RFMD模型精排部分的相似度分数可以有效支持模型预测。通过实验发现，本文所提出的Prompt在性能上优于采用“Question”和多个“Supporting Document”组合输入生成模型的方式。这是由于生成模型对输入文档的长度有限制，导致生成模型只能从长度受限的文档中提取问题答案。通过上述消融实验，证明了本文所提出的Prompt的有效性。

### 3.7 结合支持文档的问答案例

本文测试了RFMD模型处理非训练集问题的鲁棒性，这决定了RFMD模型是否真正学习到了现实世界中人类回答问题、从大量文本中寻找答案的思维方式。测试方法是将自然语言问题以及通过检索器检索得到的与问题相关的支持文档及相似度得分，组合为Prompt形式，并将其输入RFMD模型生成答案。具体案例如图5所示。

通过图5可以得出，本文模型能够处理多样化的问题并生成符合预期的问题答案。此外，统计数据表明，本文模型的平均响应时间为每个问题0.49 s。这些实验结果还表明，本文模型具备了在大量支持文档中搜索问题答案的能力。

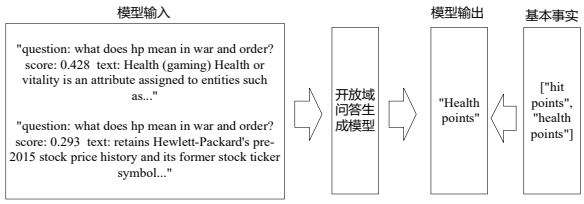



**Fig. 5 Question answer case of model with supporting documents**

**图5 模型结合支持文档回答问题案例**

### 3.8 与大语言模型比较

本文比较了RFMD模型与4种大语言模型（包括InstructGPT[30]、GPT-3[31]、LLaMA[32]和ChatGPT）在开放域问答任务上的性能表现。大语言模型详细介绍如下：

（1）InstructGPT模型。InstructGPT是基于GPT系列模型（自回归语言生成）的指令生成模型，它可以将自然语言指令转换为机器可执行的代码。该模型旨在使人类能够更轻松地与计算机进行交互。

（2）GPT-3模型。GPT-3是一种基于注意力机制的神经网络架构，可以应用于NLP中的各种任务，例如文本生成、自动问答和翻译等。它是目前最大和最先进的预训练语言模型之一，具有1 750亿个参数。

（3）LLaMA模型。LLaMA是一个开放且高效的大型基础语言模型，其数据集来源都是公开数据集，无任何定制数据集，保证了其工作与开源兼容和可复现，整个训练数据集在Token化之后大约包含1.4 T的Token。

（4）ChatGPT模型。ChatGPT是目前最先进的对话生成模型之一，也是一种通用语言模型，可以根据用户输入的文本进行回复，并且具有较高的自然度和连贯性。它可以用于多个自然语言处理任务，如文本分类、命名实体识别、问答和对话生成等。

由于大语言模型生成的内容通常具有极高可读性且对人类友好的特性，故其在回答问题时可能会包含大量的冗余信息和解释性文字，而不仅仅是直接并精确地提供问题的答案。为此，在对大语言模型进行评估时，本文采用与前人研究相同的一种“包含”形式的评价方法，即若大语言模型生成的内容片段中包含数据集答案列表中的任何一个答案，便认为它实现了精确匹配[33]。表4展示了大语言模型与本文所提模型的性能比较，实验结果表明，本文模型在NQ数据集上表现最优，但在Trivia QA数据集上性能表现不如ChatGPT模型。虽然如此，本文RFMD模型的效果仍具有竞争力。通过比较大语言模型的性能可以发现，ChatGPT模型的性能相较于其他3个大语言模型表现更优异，这证明了其在开放域问答任务方面的出色表现能力。此外，相比于大语言模型系列产品，本文提出的RFMD模型主要拥有以下优势：①可部署性强：本文模型相比于ChatGPT等模型具有参数量较小、所需存储空间少等优势，这使得本文模型更容易在不同的设备上部署，并可以更快地加载运行；②架构简单：本文模型采用了简单的Transformer架构，这使得其训练和部署过程都较位简单；与此相反，Chat GPT等模型具有较大的参数规模和复杂的架构，使得其训练和部署变得更加困难；③面向任务微调：RFMD模型在设计时考虑到了面向任务微调，因此它更适合进行特定任务的微调，使用微调技术，可以将RFMD模型应用于不同的开放域问答问题中，并且能够在不同的数据集上获得良好的性能表现；④事实性强：RFMD模型具有较高的准确性和事实性，相比其他模型，在使用多种候选文档时，RFMD模型能够更好地从中选择正确的答案，并且可以更准确地回答问题。

**Table 4 Performance comparison among the proposed model and the large language models**

**表4 本文模型与大语言模型性能比较**

| 模型 | EM评估指标/% | |
|---|---|---|
| | NQ | Trivia QA |
| InstructGPT | 19.5 | 57.4 |
| GPT-3 | 14.6 | 49.2 |
| LLaMA | 16.8 | 50.0 |
| ChatGPT | 32.8 | **64.1** |
| RFMD模型 | **45.8** | 63.4 |

## 4 结语

本文提出了基于多文档精排与融合的开放域问答方法，该方法包含原始语料检索、候选文档精排、Prompt设计和基于生成模型的问答4个模块。主要利用TF-IDF进行快速知识库原始语料检索，获取候选文档，利用Sbert对候选文档进行精排，并为生成模型设计Prompt，最后基于生成模型从多个支持文档中寻找并聚合相关证据生成问题答案。本文通过四级结构实现了对开放域问答任务的改进，并在NQ和Trivia QA数据集上验证了RFMD模型的有

效性。此外,通过消融实验证明了本文提出的精排模块和Prompt的有效性,并结合支持文档的问答案例研究证明了RFMD模型还具备响应的高效性和处理多类型问题的鲁棒性。将RFMD模型与大语言模型性能进行比较,证明了RFMD模型的优异性能。未来计划将检索和精排过程集成到生成模型中,实现端到端的开放域问答系统。

**参考文献:**


[1] LI D Q, LI M X, ZHANG X, et al. Research on open-domain question answering based on knowledge base[J]. Computer Knowledge and Technology: Academic Edition, 2020, 16(36): 179-181.
李东奇,李明鑫,张潇. 基于知识库的开放域问答研究[J]. 电脑知识与技术:学术版,2020,16(36):179-181.

[2] TONG G F. Research on open-domain knowledge question answering system based on knowledge base[D]. Nanjing: Nanjing University, 2018.
童国烽. 基于知识库的开放域知识问答系统研究[D]. 南京:南京大学,2018.

[3] WANG T B, HUANG R Y, ZHANG J P, et al. Design and implementation of a knowledge graph question answering system integrating machine reading comprehension [J]. Journal of Information Engineering University, 2021, 22(6): 709-715.
王天彬,黄瑞阳,张建朋,等. 融合机器阅读理解的知识图谱问答系统设计与实现[J]. 信息工程大学学报,2021,22(6):709-715.

[4] CHEN Z R, WANG X, WANG L, et al. A review of open-domain knowledge graph question answering research[J]. Journal of Computer Science and Exploration, 2021, 15(10): 1843-1869.
陈子睿,王鑫,王林,等. 开放领域知识图谱问答研究综述[J]. 计算机科学与探索,2021,15(10):1843-1869.

[5] ZHU F, LEI W, WANG C, et al. Retrieving and reading: a comprehensive survey on open-domain question answering[DB/OL]. https://arxiv.org/abs/2101.00774.

[6] SALTON G, WONG A, YANG C S. A vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613-620.

[7] AMATI G, VAN RIJSBERGEN C J. Probabilistic models of information retrieval based on measuring the divergence from randomness [J]. ACM Transactions on Information Systems (TOIS), 2002, 20(4): 357-3389.

[8] SHIRI A. Introduction to modern information retrieval[J]. Library Review, 2004, 53(9): 462-3.

[9] HUANG P S, HE X, GAO J, et al. Learning deep structured semantic models for web search using clickthrough data[C]// Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013: 2333-2338.

[10] GUU K, LEE K, TUNG Z, et al. Retrieval augmented language model pre-training[C]// International Conference on Machine Learning, 2020: 3929-3938.

[11] SANG E F, DE MEULDER F. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition[C]// Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, 2003: 142-147.

[12] CHEN D, FISCH A, WESTON J, et al. Reading Wikipedia to answer open-domain questions[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 1870-1879.

[13] KARPUKHIN V, OĞUZ B, MIN S, et al. Dense passage retrieval for open-domain question answering[C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020: 6769-6781.

[14] ROBERTS A, RAFFEL C, SHAZEER N. How much knowledge can you pack into the parameters of a language model?[C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020: 5418-5426.

[15] DAVIS M, DüRST M. Unicode normalization forms[EB/OL]. https://www.unicode.org/reports/tr15/.

[16] RAMOS J. Using TF-IDF to determine word relevance in document queries[C]// Proceedings of the First Instructional Conference on Machine Learning, 2003(1): 29-48.

[17] REIMERS N, GUREVYCH I. Sentence-BERT: sentence embeddings using siamese BERT-networks[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 3982-3992.

[18] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[C]// Proceedings of the 26th International Conference on Neural Information Processing Systems, 2013: 3111-3119.

[19] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [DB/OL]. https://arxiv.org/abs/1301.3781.

[20] LE Q, MIKOLOV T. Distributed representations of sentences and documents[C]//International Conference on Machine Learning, 2014: 1188-1196.

[21] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[DB/OL]. https://arxiv.org/abs/1706.03762v2, 2017.

[22] KWIATKOWSKI T, PALOMAKI J, REDFIELD O, et al. Natural questions: a benchmark for question answering research[J]. Transactions of the Association for Computational Linguistics, 2019, 7: 453-466.

[23] JOSHI M, CHOI E, WELD D S, et al. TriviaQA: a large scale distantly supervised challenge dataset for reading comprehension[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 1601-1611.

[24] LEE K, CHANG M W, TOUTANOVA K. Latent retrieval for weakly supervised open domain question answering[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019: 6086-6096.

[25] RAJPURKAR P, ZHANG J, LOPYREV K, 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.

[26] RAFFEL C, SHAZEER N, ROBERTS A, et al. Exploring the limits of transfer learning with a unified text-to-text transformer[J]. The Journal of Machine Learning Research, 2020, 21(1): 5485-5551.

[27] CHUNG H W, HOU L, LONGPRE S, et al. Scaling instruction-finetuned language models[J]. Journal of Machine Learning Research, 2024, 25(70): 1-53.

[28] MIN S, CHEN D, HAJISHIRZI H, et al. A discrete hard EM approach for weakly supervised question answering[DB/OL]. https://arxiv.org/pdf/1909.04849.

[29] IYER S, MIN S, MEHDAD Y, et al. RECONSIDER: improved re-ranking using span-focused cross-attention for open domain question answering[C]// Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021: 1280-1287.

[30] OUYANG L, WU J, JIANG X, et al. Training language models to follow instructions with human feedback[J]. Advances in Neural Information Processing Systems, 2022, 35: 27730-27744.

[31] BROWN T, MANN B, RYDER N, et al. Language models are few-shot learners[J]. Advances in Neural Information Processing Systems, 2020, 33: 1877-901.

[32] TOUVRON H, MARTIN L, STONE K, et al. Llama 2: open foundation and fine-tuned chat models[DB/OL]. https://arxiv.org/abs/2307.09288.

[33] REN R, WANG Y, QU Y, et al. Investigating the factual knowledge boundary of large language models with retrieval augmentation[DB/OL]. https://arxiv.org/pdf/2307.11019.


(责任编辑:孙　娟)