计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(6):293−300 [doi: 10.15888/j.cnki.csa.009150]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于关键短语抽取与答案过滤的问答对生成①

郭峥嵘[1], 郭躬德[1], 王　晖[2]

[1](福建师范大学 计算机与网络空间安全学院, 福州 350117)
[2](贝尔法斯特女王大学 电子电气工程和计算机科学学院, 贝尔法斯特 BT9 5BN)
通信作者: 郭躬德, E-mail: ggd@fjnu.edu.cn; 王　晖, E-mail: h.wang@qub.ac.uk

摘　要: 高质量的问答对有助于从文章中获取知识, 提高问答系统性能, 促进机器阅读理解, 在人类活动和人工智能领域中都起着较为重要的作用. 当前主要问答对生成方法依靠提供文章中的候选答案, 根据答案生成特定的问题. 然而一些候选答案可能会生成无法从文章中回答的问题, 或是生成问题的答案不再是候选答案, 造成问答对相关性差, 影响问答对的质量. 针对此问题, 本文提出了一个基于关键短语抽取与过滤生成问答对的方法. 该方法能够在输入文本中自动抽取适合生成问题的关键短语作为候选答案, 再根据候选答案在问题生成器和答案生成器中生成问答对, 并通过对比候选答案与生成答案的相似度过滤相关性低的问答对, 最终输出保证质量的问答对. 本方法在SQUAD1.1 和 NewsQA 数据集上进行了实验验证, 并人工检验了生成的问答对的质量, 结果表明该方法可以有效提高生成的问答对的质量.

关键词: 问答对; 候选答案; 关键短语抽取; T5 模型; 相似度过滤

引用格式: 郭峥嵘,郭躬德,王晖.基于关键短语抽取与答案过滤的问答对生成.计算机系统应用,2023,32(6):293–300. http://www.c-s-a.org.cn/1003-3254/9150.html

Question-answer Pair Generation Based on Key Phrase Extraction and Answer Filtering

GUO Zheng-Rong[1], GUO Gong-De[1], WANG Hui[2]

[1](College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China)
[2](School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT9 5BN, United Kingdom)

Abstract: High-quality question-answering plays an important role in human activities and artificial intelligence because it can help to obtain knowledge from articles, improve the performance of question-answering systems, and promote machine reading comprehension. The current mainstream question-answer pair generation methods usually rely on candidate answers in the provided article to generate specific questions based on these answers. However, some candidate answers may generate questions that cannot be answered from the article, or the answers to the generated questions are no longer the same as the candidate answers, which thus results in a poor correlation of the question-answer pairs and affects the quality of the question-answer pairs. In order to solve these problems, this study proposes a method to generate question-answer pairs based on key phrase extraction and filtering. The method can automatically extract key phrases suitable for generating questions from the input text as the candidate answers and then generate question-answer pairs by a question generator and an answer generator according to the candidate answers. Finally, the method outputs question-answer pairs with high quality by comparing the similarity between the candidate answers and the generated answers and filtering out those question-answer pairs that have a low correlation with the candidate answers. The proposed method is evaluated by experiments on SQUAD1.1 and NewsQA datasets, and the quality of generated question-answer pairs is manually checked. The results show that this method can effectively improve the quality of generated question-answer pairs.

① 基金项目: 国家自然科学基金 (61976053, 62171131); 福建省自然科学基金 (2022J01398)
收稿时间: 2022-12-06; 修改时间: 2023-01-19; 采用时间: 2023-02-03; csa 在线出版时间: 2023-04-25
CNKI 网络首发时间: 2023-04-26

Key words: questions-answer pair; candidate answer; key phrase extraction; T5 model; similarity filtering

问答对运用在许多的自然语言处理任务中, 如机器阅读理解, 自动问答系统, 机器人聊天系统等[1,2], 通过人工进行问答对标记需要消耗大量的时间与财力[3,4], 因此许多学者把研究重点放在从文章中自动抽取高质量的问答对上. 随着深度学习的发展, 目前主要的问答对生成工作[5–9] 是通过使用各种方法训练深度神经网络从文章中找到候选答案, 再根据候选答案生成问题. 然而这些方法通常需要复杂的规则和大量的数据训练模型[6,7,10], 且可能会出现候选答案与基于候选答案生成的问题的对应答案不一致或基于候选答案生成的问题无法从文中找到对应答案, 这种情况称为问答对相关性差[11].

针对上述问题, 本文通过分析候选答案与生成问题的关系以及如何确保问答对的质量, 提出了一种基于关键短语抽取与过滤生成问答对的方法. 本文的主要工作如下.

(1) 通过对 SQUAD1.1[12] 和 NewsQA[13] 中的大量文章抽取出的命名实体进行问题生成和依赖解析, 我们发现含有某些依赖标签如: 介词宾语, 形容词修饰语等的命名实体能够生成相关性较高的问题, 而含有另一些依赖标签如: 复合词, 占有修饰词等的命名实体虽然本身生成的问题相关性较低, 但经过一定的组合变化后也能够生成相关性较高的问题. 我们把能够生成相关性较高问题的短语称为文章的关键短语, 并提出一种从文章中抽取关键短语的方法.

(2) 为了进一步提高问答对质量, 我们提出一种问答对过滤方法. 我们将关键短语在问题生成器和答案生成器上生成的对应问答对组合成<关键短语, 问题, 答案>, 对其中关键短语和答案进行相似度过滤, 留下相似度较高或一致的问答对以确保质量.

1 相关工作

问答对生成的基础任务是问题生成. 问题生成任务[14–19] 是自然语言处理任务中长期被研究的一个任务, 问题生成的方式主要有两种: 基于模板和基于模型的方法. 基于模板的方法[3,4] 依赖于人类的努力来设计模板规则, 因此无法跨数据集进行扩展. 相反, 基于模型的方法[14,15,19] 采用端到端神经网络以及注意力机制, 在文章中选择合适的候选答案, 生成符合该答案的问题. 然而, 这种方法无法直接从文章中生成问题, 需要有标注的文本语料库来训练候选答案抽取模型或者序列标注模型[7,20] 来确定文章的哪一部分是值得提问的.

现有的大部分问答对生成任务[21–25] 通过各种方法寻找文章中哪些内容应该被提问, Liu 等人[21] 使用事件抽取和模板设计生成问题, 并通过 BERT 模型将事件中的参数提取作为问题的答案. 该方法可以生成带有上下文相关信息的问题. Liu 等人[24] 通过抽取文本中的候选答案和线索信息生成问题, 该方法一旦选定候选答案与线索信息, 问题生成将成为接近于一对一的映射任务, 以解决问题与答案存在一对多的关系. Pan 等人[25] 抽取文章中的命名实体作为答案生成问题, 避免使用复杂模型从文章中获取候选答案. 这些方法在一定程度上提升了问答对的质量, 但是依然可能生成相关性低[9] 的问答对, 即模型抽取到的候选答案无法生成符合该答案的问题或生成的问题无法回答等. Saxena 等人[22] 提出学习知识图谱在嵌入空间中的表示与问题的嵌入, 而后结合这些嵌入来预测答案. 该方法实现了从多跳的知识图谱中寻找答案. 然而使用知识图谱生成问答对需要提供复杂的实体间关系, 且通过得分的高低判定实体是否是最符合问题的答案依旧可能出现错误. Alberti 等人[5] 提出通过往返一致性来过滤相关性低的问答对. 该方法将候选答案与真实答案不一致的问答对过滤, 提高了问答对的相关性, 但实际上可能存在候选答案与真实答案不完全一致但意思相同的情况. Cui 等人[11] 提出使用一站式方式从文章中抽取问答对来确保问答对的相关性. 但该模型训练需要一个文本中只能对应一对问答对, 而实际上的一个文本可能对应多个问答对.

不同于上述工作, 我们提出了一种关键短语抽取与过滤的方法, 旨在能够从未标记的文章中抽取生成高相关性问题的关键短语作为候选答案, 我们还提出一种过滤方法, 过滤掉关键词与生成答案相关性差的问答对, 旨在保证最终生成的问答对的质量. 不同于往返一致性[5] 的过滤方式, 我们的过滤方法可以保留关键词与答案不一致但是意思相近的问答对.

2 方法介绍

本文提出的基于关键短语抽取与过滤的问答对生成方法 (question-answer pair generation based on key phrase extraction and filtering, KPEF-QA), 主要包括关键短语抽取模块, 问答对生成模块和相似度过滤模块, 总体框架如图 1 所示.

我们定义P为文本输入, 可以是一篇文章, 一段话或一个句子; $K=\{k_1,k_2,\cdots,k_n\}$为从P中抽取的关键短语集合; $Q=\{q_1,q_2,\cdots,q_n\}$为由k_i ($k_i\in K, i=1,2,\cdots,n$)与P生成对应问题的集合; $A=\{a_1,a_2,\cdots,a_n\}$为q_i ($q_i\in Q$, $i=1,2,\cdots,n$)在P中对应答案的集合; $Q'=\{q'_1,q'_2,\cdots,q'_m\}$为过滤后问题集合, $A'=\{a'_1,a'_2,\cdots,a'_m\}$为过滤后答案集合.

KPEF-QA 主要工作流程为: 输入P, 关键短语抽取模块通过命名实体识别 (NER) 与依赖分析 (DP) 自动从P中抽取K, 并将$<P,K>$输入问答对生成模块中. 问答对生成模块中有问题生成器和答案生成器, 根据$<P,K>$生成Q与A组合成$<P,Q,A>$, 再与$<P,K>$一起输入相似度过滤模块.相似度过滤模块通过对每一组问答对的答案与产生对应问题的关键短语进行重合度过滤与相似度过滤, 以保证问答对的相关性与质量, 最终输出过滤后的问答对$<P,Q',A'>$.

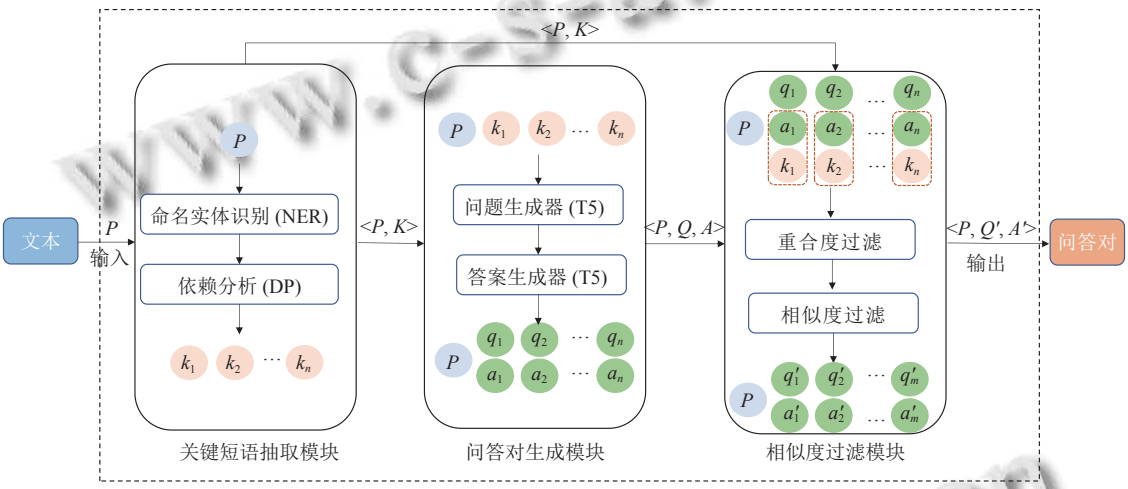

图 1 KPEF-QA 框架

2.1 关键词短语抽取模块

当文本中的候选答案能够生成相关性高的问题时, 将此类候选答案称为文本的关键短语. 本文提出一种关键短语抽取方法, 能够从任意文本中快速抽取关键短语, 该方法采用命名实体识别 (NER) 以及依赖解析 (DP) 共同完成. NER 负责标记文章中的所有命名实体, DP 负责分析该命名实体的依赖关系, 以便于发现适合生成问题的短语. 我们在 SQUAD1.1[12] 和 NewsQA[13] 的文章上进行关键短语的抽取和分析, 结合人类的提问方式, 将与依赖词的关系标签[26] 为: nsubj (名词主语), nsubjpass (被动名词主语), nummod (数值修饰), advmod (状语), amod (形容词修饰语), npadvmod (名词作状语), appos (同位修饰语), pobj (介词宾语) 的命名实体直接抽取作为关键短语. 将与依赖词的关系标签为: poss (占有修饰词), compound (复合词) 的命名实体, 根据其依赖词的位置进行组合, 生成新的关键短语. 我们去除其他抽取到的冗余命名实体, 根据上述规则定义集合$Label1$, $Label2$如下:

$$Label1=\{\text{nsubj},\text{nsubjpass},\text{nummod},\text{advmod},\text{amod},\text{npadvmod},\text{appos},\text{pobj}\} \quad (1)$$

$$Label2=\{\text{poss},\text{compound}\} \quad (2)$$

我们使用 spaCy 库[27] 来实现抽取命名实体与构建依赖树. 关键短语抽取具体过程如算法 1 所示.

算法 1. 关键短语抽取

输入: 需要抽取关键短语的文本 P
输出: 文本 P 的关键短语集合 K

1) $ners$= NER(P) #抽取P的所有命名实体$ners$
2) dps= DP($ners$) #对每个命名实体做依赖解析
3) $keyphrase$= []
4) for ner in $ners$ do:

```
5)      if ner.dps in Label1: #命名实体的依赖标签在集合Label1中
6)          keyphrase.append(ner)
7)      end if
8)      if ner.dps in Label2: #命名实体的依赖标签在集合 Label2 中
9)          if ner.end<ner.head.pos: #命名实体的结束位置在其依赖词位置之前
10)             ner′= join(ner.start,ner.head.pos) #连接命名实体的开始位置到其依赖词位置间所有单词
11)         end if
12)     if ner.start>ner.head.pos: #命名实体的开始位置在其依赖词位置之后
13)         ner′ = join(ner.head.pos,ner.end) #连接其依赖词位置到命名实体的结束位置间所有单词
14)     end if
15)         keyphrase.append(ner′)
16)     end if
17) end for
```

从文本中抽取关键短语如图 2 所示, 其中阴影部分为抽取的命名实体, 箭头指向该实体词的依赖词, 箭头上的标签为实体词的依赖标签, 圆角矩形内为抽取到该文本的关键短语. 在该文本中 NER 标记出了 4 个命名实体, 经过 DP 分析, 其中命名实体“2015-2016”“Notre Dame”“18th”的依赖标签分别为pobj, nsubj, advmod ∈ *Label*1, 因此直接成为关键短语, 而命名实体“U.S. News & World Report’s”的依赖标签为poss ∈ *Label*2, 因此与其依赖词“Colleges”组合成为“U.S. News & World Report’s Best Colleges”作为关键短语.

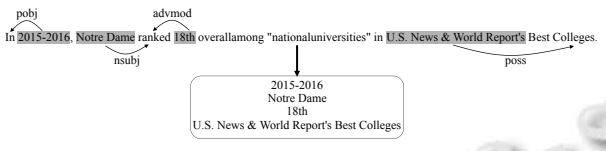

图 2 文本中抽取关键短语

2.2 问答对生成模块

问答对生成模块中有问题生成器和答案生成器, 其工作流程如图 3 所示. 模块首先将文本P和从中抽取的关键短语集合$K=\{k_1,k_2,\cdots,k_n\}$组合成$<P,K>$输入问题生成器中, 问题生成器将生成每一个关键短语相对应的问题$Q=\{q_1,q_2,\cdots,q_n\}$, 之后将Q与P组合成$<P,Q>$输入到答案生成器中, 答案生成器将生成每一个问题对应的答案$A=\{a_1,a_2,\cdots,a_n\}$, 最后将文本与对应的问答对$<P,Q,A>$输出.

实验中, 我们使用经过格式处理的 SQUAD1.1 数据集[12], 微调 text-to-text transfer Transformer (T5) 模型[28]作为问题生成器和答案生成器, 实现问题生成和答案生成的下游任务.

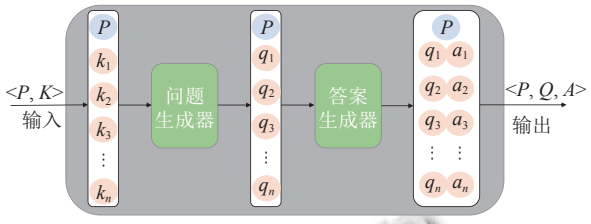

图 3 问答对生成模块工作流程

2.3 相似度过滤模块

为了解决问答对可能出现的相关性差[11] 的情况, 我们提出一种相似度过滤方法, 通过对比生成问答对$<q_i,a_i>$ $(q_i\in Q,a_i\in A)$ 的关键短语k_i $(k_i\in K)$ 与答案a_i的相似度, 判断该问答对是否相关. 若k_i与a_i一致或者相似度较高, 则认为此对问答对相关性高, 反之, 则过滤掉该问答对.

由于实验中单纯使用余弦相似度[29] 方法对比k_i与a_i的相似度, 存在k_i与a_i两个短语表达的意思完全不相同但余弦相似度仍较高的情况, 为避免这种情况的发生, 我们先计算k_i与a_i的精准率 *precision* 和召回率 *recall*, 并设置重合度阈值σ (实验中设置$\sigma=0.2$), 若*precision* 或*recall* 小于σ, 表示关键短语与答案中的单词重合度过低, 直接过滤该问答对, 否则进行余弦相似度[30] 对比.

*precision*和*recall*的计算公式如下:

$$precision(k_i,a_i)=\frac{1-gram_{k_i,a_i}}{len(k_i)} \tag{3}$$

$$recall(k_i,a_i)=\frac{1-gram_{k_i,a_i}}{len(a_i)} \tag{4}$$

其中, $1-gram_{k_i,a_i}$为k_i与a_i中重合的单词数, *len*为句子的长度.

设$\vec{\alpha}=(\alpha_1,\alpha_2,\cdots,\alpha_m)$与$\vec{\beta}=(\beta_1,\beta_2,\cdots,\beta_m)$分别为关键短语$k_i$与答案$a_i$长度为$m$的词频向量, k_i与a_i的余弦相似度$similarity(\vec{\alpha},\vec{\beta})$定义如下:

$$similarity(\vec{\alpha},\vec{\beta})=\frac{\sum_{i=1}^{m}(\alpha_i\times\beta_i)}{\sqrt{\sum_{i=1}^{m}(\alpha_i)^2}\times\sqrt{\sum_{i=1}^{m}(\beta_i)^2}} \tag{5}$$

δ为设定的相似度阈值, 当$similarity(\vec{\alpha},\vec{\beta})>\delta$时, 保留该问答对. δ的设置要同时兼顾问答对的数量与相

关性. 问答对相似度过滤算法如算法 2 所示.

算法 2. 问答对相似度过滤

输入: 文本, 关键短语与问答对P, K, Q, A
输出: 文本与过滤之后的问答对P, Q', A'

```
l= len(K)      #获取集合K的长度
for i in range(1, l)
    precision= get_precision(k_i, a_i)
    recall= get_recall(k_i, a_i)
    if precision or recall<σ:
        filter(k_i, a_i)
    else
        similarity= get_similarity(k_i, a_i)
        if similarity<δ:
            filter(k_i, a_i)
        end if
    end if    #过滤重合度小与σ或相似度小与δ的问答对
end for
```

3 实验评估

本节重点介绍实验中使用的数据集, 采用的模型评估方式与评估指标.

3.1 数据集

实验使用 SQUAD1.1[12] 与 NewsQA[13] 数据集进行评估测试. SQUAD1.1 是一个阅读理解数据集, 其中包含来自维基百科的文章与关于该文章的问题, 每个问题的答案都是来自相应段落的文本片段. NewsQA 中的文章来自 CNN 的新闻, 每篇文章较长, 与 SQUAD 类似, 其中包含关于文章的问题, 答案在相应的文章中.

3.2 评价指标

实验使用 BLEU[30], ROUGE-L[31], METEOR[32] 方法测试模型的性能. BLEU 通过模型生成句子中的单词出现在参考句子中的数量来计算精度, BLEU-1、BLEU-2、BLEU-3 和 BLEU-4 分别使用 1-gram 到 4-gram 进行精度计算. ROUGE-L 使用基于最长公共子序列 (LCS) 的统计数据通过参考句子中的单词出现在模型生成句子中的次数来计算召回率. METEOR 通过单元词组 (unigram) 匹配, 计算基于准确率和召回率的调和平均值.

我们使用 EM[12] 和 $F1$ 值[12] 测试生成答案的准确性, EM (exact match) 计算模型预测的答案和正确标注答案完全匹配的数量, $F1$ 则根据模型预测的答案和正确标注答案之间的重合程度计算出一个 0 到 1 之间的得分, 即词级别的正确率和召回率的调和平均值.

3.3 问题生成器质量评估

我们使用 SQUAD1.1 与 NewsQA 数据集中的文章测试 KPEF-QA 中问题生成的性能. 输入 SQUAD1.1 和 NewsQA 中的文章和答案, 通过问题生成器生成问题, 对比原问题和生成问题的 BLEU-1, BLEU-2, ROUGE-L 值, 从而评估生成问题的质量. 由于使用了相同的数据集与测试方法, 我们直接沿用了文献 [11] 中问题生成评估表, 比较结果如表 1 与表 2 所示. 结果表明我们问题生成器生成的问题质量优于大部分主流的问题生成模型.

表 1 SQUAD1.1 数据集问题生成质量评估

模型	BLEU-1	BLEU-2	Rouge-L
DeepNQG	17.49	8.81	17.77
CRF-DeepNQG	19.61	9.68	18.92
BART-QG	31.36	21.25	29.06
BART-A2QG	20.85	10.51	18.81
OneStop	31.32	21.28	29.10
KPEF-QG	**45.29**	**35.14**	**45.26**

表 2 NewsQA 数据集问题生成质量评估

模型	BLEU-1	BLEU-2	Rouge-L
DeepNQG	14.30	6.22	14.79
CRF-DeepNQG	17.06	7.93	17.11
BART-QG	22.30	13.48	21.81
BART-A2QG	21.53	11.81	21.75
OneStop	22.28	13.46	21.90
KPEF-QG	**22.44**	**13.05**	**25.64**

3.4 相似度阈值对问答对数量与准确率的影响

实验通过设置 KPEF-QA 方法中不同的相似度阈值δ, 对比从文本P中抽取的关键短语k_i ($k_i \in K$) 与使用k_i生成问答对$< q_i, a_i > (q_i \in Q, a_i \in A)$ 中a_i的 BLEU[30], ROUGE-L[31], METEOR[32] 值和生成的问答对的数量变化, 反映不同的相似度阈值对实验结果的影响. 我们使用 SQUAD1.1 中 19 047 篇文章和 NewsQA 中 5 127 篇文章作为数据集测试在不同的相似度阈值δ下问答对数量与质量的指标, 结果如表 3 与表 4 所示, 表中 Bi 表示 BLEU-i, sum为总共生成问答对数量, avg为平均每篇文章问答对数量.

实验发现δ从 0.5 提升至 0.95 的过程中, 在 19 047 篇 SQUAD 文章中, B1 从 49.27 提升至 69.83, B2 从 28.13 提升至 43.30, 而 METEOR 则从 51.99 提升至 71.62, ROUGE_L 从 61.92 提升至 73.48. 在 5 127 篇 NewsQA 文章中, B1 从 36.58 提升至 63.37, B2 从

19.17 提升至 42.02, METEOR 从 45.97 提升至 69.70, ROUGE_L 从 54.57 提升至 69.54. 随着δ的提升生成的问答对的数量在减少, 当$\delta=0.95$时, 平均每篇 SQUAD 文章只生成 3.5 对问答对, NewsQA 文章只生成 7.16 对问答对. 在实际应用中可根据需求通过调整相似度阈值来平衡问答对的质量和数量.

表 3 SQUAD1.1 数据集在不同相似度阈值下各项指标

δ	B1	B2	B3	B4	METEOR	ROUGE-L	sum	avg
0.5	49.27	28.13	17.36	10.82	51.99	61.92	96 143	5.04
0.65	52.45	30.89	19.39	12.41	55.33	64.79	87 161	4.58
0.75	55.25	33.23	21.47	14.17	57.06	66.28	84 148	4.42
0.85	58.46	36.36	24.74	17.36	59.23	67.82	80 221	4.21
0.95	65.83	43.30	32.49	25.33	71.62	73.48	66 624	3.50

表 4 NewsQA 数据集在不同相似度阈值下各项指标

δ	B1	B2	B3	B4	METEOR	ROUGE-L	sum	avg
0.5	36.58	19.17	10.01	5.22	45.97	54.57	58 016	11.32
0.65	42.25	23.26	12.61	6.84	50.05	58.80	50 632	9.87
0.75	45.30	25.45	14.08	7.74	52.53	61.35	47 627	9.30
0.85	54.50	33.11	20.48	12.75	57.82	64.21	43 666	5.50
0.95	63.37	42.02	29.60	21.54	69.70	69.54	36 702	7.16

3.5 生成答案准确性测试

用 NER, key-phrase, key-phrase+filter ($\delta=0.9$) 这 3 种方法从 SQUAD1.1 文章中抽取候选答案, 对比候选答案与生成答案 EM 与 $F1$ 值, 验证生成答案的准确性, 结果见表 5. 实验表明 key-phrase+filter 方法能够有效提升生成答案的准确性.

表 5 不同方法抽取候选答案与生成答案准确率对比 (%)

方法	EM	$F1$
NER	50.06	75.62
Key-phrase	51.12	77.88
Key-phrase+filter	88.61	98.48

3.6 问答对质量评估

由于目前没有广为认可的问答对自动评估指标, 因此采取人工评估验证问答对的质量. 使用 KPEF-QA 方法, 设置$\delta=0.9$, 从 SQUAD1.1 数据集中随机抽取 50 篇文章生成共 186 对问答对, 邀请福建师范大学硕士生与本科生对每一对问答对从问题是否符合语法规则, 问题是否与文章相关, 答案是否正确 3 个方面进行质量评估. 评估结果如表 6 所示.

表 6 人工评估问答对质量 (%)

评估项目	问题是否符合语法规则			问题是否与文章相关		答案是否正确		
	符合	能读懂	不符合	相关	不相关	正确	部分正确	不正确
结果	85.5	11.8	2.7	96.8	3.2	92.5	1.6	5.9

结果显示使用 KPEF-QA 方法生成的 186 对问答中 97.3% 的问题是符合语法规则或是可以理解, 96.8% 的问题与文章相关, 94.1% 的答案正确或比分正确, 这证明了我们的方法可以生成高质量的问答对.

4 结语

本文提出 KPEF-QA, 一种快速从未标记的文本语料库中抽取关键短语, 生成问答对并过滤输出的方法, 该方法通过抽取关键短语与对问答对进行相似度过滤提高问答对的相关性. 实验通过自动评估与人工评估验证了生成问答对的质量, 其结果表明 KPEF-QA 能够有效从文本中生成高质量问答对. 鉴于目前还无法产生较为复杂的问答对, 如何解决这个问题是我们今后努力的方向.

参考文献

1 Hermann K M, Kočiský T, Grefenstette E, *et al*. Teaching machines to read and comprehend. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 1693–1701.

2 Joshi M, Choi E, Weld DS, *et al*. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017. 1601–1611.

3 Heilman M, Smith NA. Good question! Statistical ranking for question generation. Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Los Angeles: ACL, 2010. 609–617.
4 Labutov I, Basu S, Vanderwende L. Deep questions without deep understanding. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing: ACL, 2015. 889–898.
5 Alberti C, Andor D, Pitler E, *et al*. Synthetic QA corpora generation with roundtrip consistency. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 6168–6173
6 Du XY, Cardie C. Harvesting paragraph-level question-answer pairs from Wikipedia. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: ACL, 2018. 1907–1917.
7 Wang SY, Wei ZY, Fan ZH, *et al*. A multi-agent communication framework for question-worthy phrase extraction and question generation. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 7168–7175.
8 Du XY, Shao JR, Cardie C. Learning to ask: Neural question generation for reading comprehension. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017. 1342–1352.
9 Liu B, Zhao MJ, Niu D, *et al*. Learning to generate questions by LearningWhat not to generate. Proceedings of the 2019 World Wide Web Conference. San Francisco: ACM, 2019. 1106–1118.
10 Shinoda K, Sugawara S, Aizawa A. Improving the robustness of QA models to challenge sets with variational question-answer pair generation. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop. AAAI, 2021. 197–214.
11 Cui SB, Bao XT, Zu XX, *et al*. OneStop QAMaker: Extract question-answer pairs from text in a one-stop approach. arXiv:2102.12128, 2021.
12 Rajpurkar P, Zhang J, Lopyrev K, *et al*. SQuAD: 100 000+ questions for machine comprehension of text. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin: ACL, 2016. 2383–2392.
13 Trischler A, Wang T, Yuan XD, *et al*. NewsQA: A machine comprehension dataset. Proceedings of the 2nd Workshop on Representation Learning for NLP. Vancouver: ACL, 2017. 191–200.
14 Chan YH, Fan YC. A recurrent BERT-based model for question generation. Proceedings of the 2nd Workshop on Machine Reading for Question Answering. Hong Kong: ACL, 2019. 154–162.
15 Kim Y, Lee H, Shin J, *et al*. Improving neural question generation using answer separation. Proceedings of the 33rd AAAI Conference on Artificial Intelligence. Honolulu: AAAI, 2019. 6602–6609.
16 Pan LM, Lei WQ, Chua T, *et al*. Recent advances in neural question generation. arXiv:1905.08949, 2019.
17 Sun XW, Liu J, Lyu YJ, *et al*. Answer-focused and position-aware neural question generation. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Brussels: ACL, 2018. 3930–3939.
18 Perez E, Lewis P, Yih WT, *et al*. Unsupervised question decomposition for question answering. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing. ACL, 2020. 8864–8880.
19 Liu DH, Gong YY, Fu J, *et al*. RikiNet: Reading wikipedia pages for natural question answering. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 6762–6771.
20 Subramanian S, Wang T, Yuan XD, *et al*. Neural models for key phrase detection and question generation. arXiv:1706.04560, 2017.
21 Liu J, Chen YB, Liu K, *et al*. Event extraction as machine reading comprehension. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, 2020. 1641–1651.
22 Saxena A, Tripathi A, Talukdar P. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 4498–4507.
23 Lee DB, Lee S, Jeong WT, *et al*. Generating diverse and consistent QA pairs from contexts with information-maximizing hierarchical conditional VAEs. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020. 208–224.
24 Liu B, Wei HJ, Niu D, *et al*. Asking questions the human way: Scalable question-answer generation from text corpus. Proceedings of the 2020 Web Conference. Taipei: ACM, 2020. 2032–2043.

25 Pan LM, Chen WH, Xiong WH, *et al*. Zero-shot fact verification by claim generation. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL, 2021. 476–483.

26 Nivre J. Dependency parsing. Language and Linguistics Compass, 2010, 4(3): 138–152. [doi: 10.1111/j.1749-818X.2010.00187.x]

27 Vasiliev Y. Natural Language Processing with Python and spaCy: A Practical Introduction. San Francisco: No Starch Press, 2020.

28 Raffel C, Shazeer N, Roberts A, *et al*. Exploring the limits of transfer learning with a unified text-to-text Transformer. The Journal of Machine Learning Research, 2020, 21(1): 140.

29 Faloutsos C, Lin KI. FastMap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data. San Jose: ACM, 1995. 163–174.

30 Papineni K, Roukos S, Ward T, *et al*. BLEU: A method for automatic evaluation of machine translation. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics. Philadelphia: ACL, 2002. 311–318.

31 Lin CY. ROUGE: A package for automatic evaluation of summaries. Proceedings of the 2004 Text Summarization Branches Out. Barcelona: ACL, 2004. 74–81.

32 Lavie A, Agarwal A. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. Proceedings of the 2nd Workshop on Statistical Machine Translation. Prague: ACL, 2007. 228–231.

(校对责编: 孙君艳)