计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(10):95−101 [doi: 10.15888/j.cnki.csa.008092]

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

基于文档库的信息服务问答系统①

王明乾, 杨文静, 倪　林

(国防科技大学 信息通信学院, 西安 710100)
通讯作者: 王明乾, E-mail: 245717004@qq.com

摘　要: 当前信息环境下, 非结构化文本是各类信息的重要组成部分, 如何针对用户信息需求, 从文本数据中快速提取所需信息, 为用户提供快速高效的信息获取方式成了当前信息服务领域亟待解决的问题. 该文基于语义检索以及抽取式文档阅读理解模型, 研究了如何快速有效地从大型文档库中根据用户问题提取出所需答案信息的技术, 构建了基于文档库的信息服务问答系统. 对于解决当前海量信息环境下快速有效的帮助用户获取所需信息, 提升信息服务效率具有重要意义. 实验表明, 该系统可以快速精确的定位用户所提问题的答案, 帮助用户快速有效的获取所需信息.

关键词: 问答系统; 语义检索; 阅读理解

引用格式: 王明乾,杨文静,倪林.基于文档库的信息服务问答系统.计算机系统应用,2021,30(10):95–101. http://www.c-s-a.org.cn/1003-3254/8092.html

Question Answering System for Information Service Based on Document Library

WANG Ming-Qian, YANG Wen-Jing, NI Lin

(School of Information and Communication, National University of Defense Technology, Xi'an 710100, China)

Abstract: In the current information environment, unstructured text is an important part of information. How to quickly extract the required information from text data and provide users with an efficient way to acquire information has become an urgent problem to be solved in the current information service field. Based on semantic retrieval and an extraction-type document reading comprehension model, this work studies how to effectively extract required answers from large document libraries according to users' questions and constructs an information service question answering system based on document libraries. In the current environment with massive information, it is of great significance to improve the efficiency of users' information acquisition. Experiments show that the system can quickly and accurately locate the answers to the users' questions and help them get the required information rapidly.

Key words: question answering system; semantic retrieval; reading comprehension

当前信息环境下, 日常工作领域所需的各类专业领域信息也变得多种多样. 如何针对信息服务用户需求, 为用户提供快速高效的领域内信息获取方式也成了当前信息服务亟待解决的问题. 领域知识问答[1]方面的相关研究是解决上述问题的有效方法, 其能够通过对用户问题进行语义层面的解析并在领域文档库中匹配符合用户需求的信息并提供给用户. 问答系统(Question Answering system, QA) 是一种基于深度学习的文本处理模型, 它基于用户提出的问题在语义层面对用户需求进行分析, 并智能、简洁的回答用户提出的问题. 可以满足领域信息服务保障任务从海量信息中快速、准确、有针对性地获取信息的需求.

① 收稿时间: 2021-01-02; 修改时间: 2021-01-29; 采用时间: 2021-02-02

早在 19 世纪 60 年代, 就出现了基于问答模板、人工规则生成答案的问答系统. 70 年代还有一些基于文档库的问答系统研究. 90 年代左右, 随着搜索引擎技术的出现与发展, 基于检索的问答系统取得了一定发展. 2010 年之后, 随着自然语言处理技术的不断发展, 问答系统出现了 3 大主流方法: 语义解析、信息抽取、向量建模. 2015 年开始, 由于深度学习在自然语言建模方面取得的重大进展[2], 出现了大量使用深度学习的问答系统. 2016 年, 斯坦福大学推出了高质量机器阅读理解数据集 Stanford Question Answering Dataset (SQuAD)[3], 它是基于自然问题的抽取式阅读理解数据集. 2017 年, Wang 等[4] 提出了基于 Match-LSTM 的端到端神经网络模型. 2018 年, Yu 等[5] 提出了 QANet 模型仅使用 CNN 和 self-attention 使得模型的训练和预测的速度大大加快, 并且可以并行处理输入的单词. 2017 年, Facebook 的 Chen 等[6] 提出了机器阅读理解模型问答系统 DrQA, 利用机器阅读理解技术在非结构化文本库上构建 QA 系统.

与现有常见的开放领域问答系统相比, 限定领域问答具有显著的特点[1]:

(1) 问题的多样性较少, 即同一个问题存在多种问法的情况较少;

(2) 知识来源比较少, 数据收集存在不少困难, 也缺乏开放的知识源;

(3) 问题的理解和回答需要深入利用专业知识, 通过深度的推理准确理解问题和生成答案.

这些特点给领域信息服务问答系统的研发带来了难题. 新体制下信息服务保障需要新技术、新手段挖掘信息服务用户信息需求. 本文使用了领域内文档库作为知识来源, 构建了能够为用户提供领域内信息服务保障的信息服务问答系统.

1 信息服务问答系统

1.1 基本架构

本文中采用了端到端的闭合域问答模型来构建信息服务问答系统, 其架构基于两个主要部分: 检索器 (retriever) 和解答器 (reader). 模型的总体架构如图 1 所示.

首先利用海量的专业领域文档构建一个文档库, 当信息服务用户提出问题并输入信息服务问答系统时, 信息检索器从领域文档库中检索出 N 个语义与问题最相关, 即最可能包含答案的文档. 选择了可能性最大 N 个文档之后, 系统把 N 个文档和问题发送至解答器. 当解答器输入一个文档-问题对时, 会输出在该文档中找到的最可能的答案, 同时给出该答案是正确答案可能性评分. 最后, 将答案按照评分排序, 选择评分最高的作为最终答案.

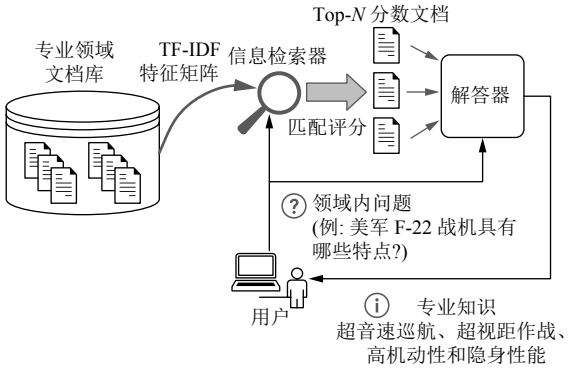

图 1 信息服务问答系统总体框架

1.2 信息检索器

本系统的信息检索器基于 TF-IDF (Term Frequency-Inverse Document Frequency) 特征计算了问题语句和文档库中的每个文档的余弦相似性, 相似性越大的文档越有可能包含问题的答案.

1.2.1 TF-IDF 算法

TF-IDF[7] 是一种统计方法, 可以用来评估一个词语对于整个文档库中每一篇文档的重要程度. 其核心思想是, 在某一个文档中出现频率高且在整个文档库中出现在其他文档中的频率少的词对该文档更重要. 因此取词频 TF (词语在文档中出现的次数) 和逆向文件频率 IDF (总文件数目除以包含该词语的文件的数目取对数) 的乘积 TF-IDF 构成了文档库的特征矩阵. 其计算公式如式 (1)–式 (3) 所示.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

$$idf_i = \ln \frac{|D|}{|\{j : w_i \in d_j\}|} \tag{2}$$

$$tfidf_{i,j} = tf_{i,j} \times idf_i \tag{3}$$

式 (1) 中, $n_{i,j}$是词语w_i在文档d_j中出现的次数, 分母则是文档d_j中所有词语出现的次数总和; $tf_{i,j}$是词频, 表示词语w_i在文档d_j中出现的频率. 式 (2) 中, $|D|$是文档库中文档的数量. $|j : t_i \in d_j|$表示包含词语w_i的文档数

量. 最后, 如式 (3) 所示, $tf_{i,j}$与idf_i相乘得到词语对于文档的权重$tfidf_{i,j}$. 在本文中使用了机器学习 scikit-learn[8] 库中的 TfidfVectorizer 模块实现了文档及问题向 TF-IDF 特征矩阵的转化.

1.2.2 匹配算法

当把文档库中的文档及用户提出的问题完全转换为包含 TF-IDF 特征矩阵后, 就可以对通过计算问题 TF-IDF 向量与每个文档 TF-IDF 的相似度, 来计算文档与用户问题之间的相似度了. 对于问题来说, 与其相似度越大的文档越有可能包含问题的答案. 本文中相似度的计算采用了余弦相似度算法[9], 其计算公式如下:

$$\cos\theta = \frac{a\cdot b}{\|a\|\times\|b\|} = \frac{\sum_{i=1}^{n}(x_i\times y_i)}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\times\sqrt{\sum_{i=1}^{n}(y_i)^2}}$$

当接收到用户提交的问题之后, 模型通过 TF-IDF 模型将问题序列转化为 TF-IDF 表征向量, 然后计算该向量与文档库的 TF-IDF 矩阵中所有向量的余弦相似度, 与问题的相似度越大的文档与问题越相关. 本文取与问题向量余弦相似性最大的 N 个文档, 即与问题相似度最大的 N 个文档作为检索的结果, 并传递给解答器.

1.3 解答器

解答器可以对信息检索器搜索的结果进行进一步处理, 对检索出的 N 个文档分别计算问题答案, 进行比较后选出最佳的答案. 解答器的核心算法就是对文档和问题对进行阅读理解, 从而分析、推理、定位问题的答案. 经典的神经网络阅读理解模型基本框架[10] 主要包括词嵌入层、语义编码层、文档-交互层、问答作答层.

本系统采用了 BiDAF 阅读理解模型[11] 作为解答器, 其具体结构如图 2 所示.

1.3.1 词嵌入层

模型使用了词嵌入层将词语映射为维度固定的词向量, 获取词嵌入的方法是在训练的过程中自动从数据中学习词嵌入向量, 学习方式与神经网络中的权重相同. 词嵌入向量蕴含了词语的语义信息, 词嵌入向量之间的几何关系可以表示词之间的语义关系.

1.3.2 语义嵌入层

语义嵌入层对词嵌入层得到的文档向量和问题向量分别进一步编码. 采用了可以让每个词语的特征向量与上下文进行了交互的双向长短时记忆网络 (Bi-LSTM), 从而捕捉到了文档中词语的语境信息, 使特征向量更好的编码了词语在当前语境下的含义. Bi-LSTM 模型是由正序的 LSTM 模型与倒序的 LSTM 模型组合而成, 分别可以获取词语对上文和下文的依赖. 从而将文档和问题原始文本整合成了包含篇章、句子级语义信息的特征表示.

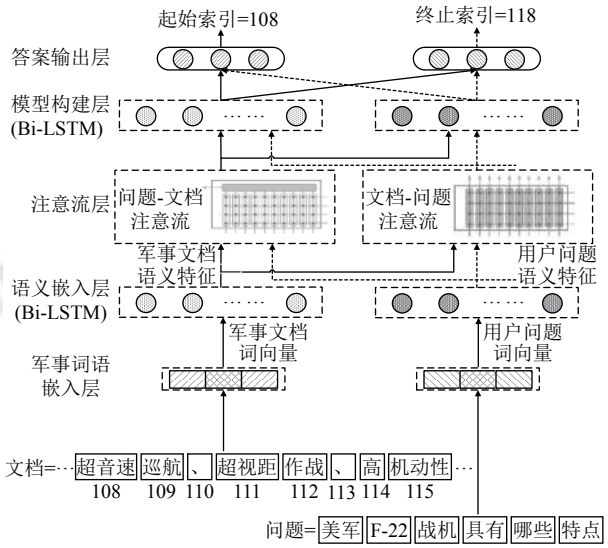

图 2 解答器模型结构

1.3.3 注意流层

获取了文档和问题各自的语义特征向量之后, 为了完成问题的解答, 需要进一步探索文档与问题之间的相关关系. 对于用户问题来说文档中的每个词语对其的重要性是不同的, 即答案部分对于问题来说比其他部分应该更重要. 反之, 对于文档来说问题中与文档相关性更大的词语对于问题解答更加重要, 因此本文采用了双向注意流模型. 包括了文档-问题和问题-文档两个方向的注意流, 前者用于获取文档更关注哪些词语, 后者用于获取对于问题来说哪个单词更重要. 最后, 双向注意流拼接起来得到输出矩阵 G, G 中每个列向量都可视为对应词语的查询感知表征.

1.3.4 模型构建层

模型构建层的输入为注意流层的输出 G, 经过一个 Bi-LSTM 层得到$|M\in R^{2d\times T}|$, M 的每一个列向量都包含了对应单词查询感知的上下文信息, 捕获输入矩阵在时序上依赖关系, 而且还具有编码降维的功能.

1.3.5 答案输出层

问答答案输出层的功能是生成答案起始位置 p1 和结束位置 p2, 最终根据 p1 和 p2 从文档中截取预测的答案.

2 评价指标

2.1 检索器评价指标

检索器的评估指标使用正确率来衡量, 已知每个问题对应一篇文章, 检索器的作用就是根据问题从文档库里找到其答案所在的文章. 当给定一个问题时, 检索器从文档库中选出与问题最相似的 N 篇文档作为返回结果. 如果其答案所在的文档包含在检索结果中可以认为找到了所需的文章, 即检索成功, 结果正确. 如果问题所对应的文档没有包含在检索结果中, 说明检索失败, 结果错误. 本文使用检索的正确率来检验检索器的效果. 公式如下:

$$R_{\text{right}} = \frac{N_{\text{right}}}{N_{\text{all}}}$$

其中, N_{right}为测试样本中检索结果正确的检索数量, N_{all}为检索所有测试检索的总数.

2.2 解答器评价指标

阅读理解的评价一般以预测的答案与实际答案的匹配度进行衡量, 本文中同时计算了 ROUGE-L 和 BLEU 两个指标, 其中, ROUGE-L 作为第一参考指标, BLEU 作为第二参考指标.

(1) ROUGE-L

ROUGE 评价方法[12] 一般是用于摘要的评价, 其基于摘要中 n 元词 (n-gram) 的共现信息来评价摘要, 是一种面向 n 元词召回率的评价方法. 其通过统计人工摘要与自动摘要共现的基本单元 (n 元语法、词序列和词对) 的数目, 来评价摘要的质量. ROUGE 准则由一系列的评价方法组成, 其中, ROUGE-L 使用了最长公共子序列作为基本单元, 字母 L 即是最长公共子序列 (Longest Common Subsequence, LCS) 的首字母. ROUGE-L 计算方式如下:

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

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

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

其中, $LCS(X,Y)$是 X 和 Y 的最长公共子序列的长度, m, n 分别表示人工标准摘要和机器自动摘要的长度 (包含词语的数量), R_{LCS}, P_{LCS}分别表示召回率和准确率. 最后的F_{LCS}即是 ROUGE−L. 使用 LCS 的优点是不需要连续匹配, 而且反映了句子级词序的顺序匹配, 由于它自动包含最长的顺序通用 n-gram, 因此不需要预定义的 n-gram 长度. 缺点是只计算一个最长子序列, 最终的值忽略了其他备选的最长子序列及较短子序列的影响.

(2) BLEU

BLEU 方法[13] 计算两段文本之间的相似度, 不考虑词语的顺序, 将待评价文本和参考文本的 n-gram 单元进行匹配, 并计算匹配单元的个数. 匹配单元数越多, 则待评价文本与参考文本越相似, 即质量越好. 算法中 N 的值可以变化, BLEU-4 即 N 值为 4. 公式为:

$$BLEU\text{-}N = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \ln p_n\right)$$

其中, BP 为惩罚因子, p_n为多元精度, w_n为多元精度对应的权重. 结果值介于 (0, 1), 越大越好.

3 实验分析

3.1 数据准备

本次实验使用莱斯杯阅读理解初赛数据集对检索器和解答器进行了训练和评估, 该数据集包括新闻类、防务快讯类 2 万余篇的文档, 每个文档对应 5 个左右的问题以及人工标注的问题答案, 约 10 个万问题答案对.

数据集为 JSON 文件格式, 包含多个文章以及每篇文章的问题-答案对, 其中每个文章的答案为单行的 JSON 数据.

3.2 实验环境

本文选用了百度的飞桨 (PaddlePaddle) 作为深度学习模型的基础框架, 其目前国内自主研发、开源开放、功能完备的产业级深度学习平台, 集深度学习核心框架、基础模型库、端到端开发套件、工具组件和服务平台于一体. 采用基于编程逻辑的组网范式, 支持声明式和命令式编程, 兼具开发的灵活性和高性能. 采用了在计算科学领域的领先地位、具有生态完整性和接口易用性的 Python 作为编程语言. 具体环境配置如表 1 所示.

3.3 评估

3.3.1 检索器评估

实验中, 首先构建检索器评估数据集. 将数据集转换为问题-文档对. 使用原始数据集构建了 90 000 多条问题-文档对, 使用 sklearn 的 TfidfVectorizer 将其转化为 TF-IDF 矩阵作为检索数据库. 并从问题-文档对中选取了 1000 条数据作为测试集. 依次设定 N 为 3–19, 测试检索器的正确率. 具体结果如图 3 所示.

表 1 软硬件环境配置

软硬件	参数
CPU	4110
内存	32 GB
GPU	RTX 2080Ti
系统平台	Ubuntu18.04
CUDA	10.0
Python	3.7.4
PaddlePaddle	1.8.4

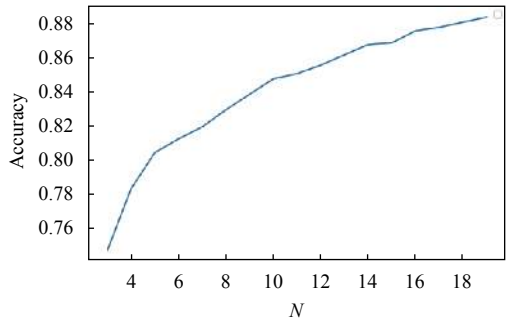

图 3 检索器评估结果

由图 3 可见, N 值越大, 检索器的正确率越高, 但是当 N 大于 12 之后, 其正确率增长变缓. 由于 N 值越大, 其结果包含的文档数越多, 不利于解答器进一步从检索结果中查找最终答案, 同时会大大增加解答器的计算量.

3.3.2 解答器评估

(1) 数据预处理

首先对问题和文档进行分词, 然后在对于文档中定位答案范围, 这里使用了一个简单的策略, 将真实答案与每段文章进行匹配, 搜索与真实答案 $F1$ 分数最高的字符串, 并使用这个字符串的范围作为候选答案范围. 本文为每个问题寻找了一个范围作为候选项, 使用其在文档中的位置索引表示 [start_idx, end_idx], 其中, start_idx 是答案的起始端, end_idx 为答案的终止端.

(2) 模型训练

训练时将文档与问题作为输入, 模型的输出与正确的答案范围进行比较计算误差后对模型权重进行更新.解答器重要超参数设置如表 2 所示.

表 2 中, batch_size 为训练时每批次数据数量; cuda 为布尔型变量, 为 True 时使用 GPU 进行训练, 为 False 时使用 CPU 进行训练; embed_size 为字嵌入层词向量的维度; hidden_size 为模型隐藏层单元维度; init_lr 表示初始学习率, 表示模型学习的快慢; max_answer_len 表示答案保留的最长长度; max_article_len 表示文档保留的最长长度; max_question_len 表示问题保留的最长长度.

表 2 解答器重要超参设置

参数	值
batch_size	32
cuda	True
embed_size	300
hidden_size	150
init_lr	0.001
max_answer_len	200
max_article_len	500
max_question_le	60

本文在训练时初始学习率设置为 0.001, 训练过程中每个 epoch 完成后, 模型在测试集上的损失 Loss 及评估指标 ROUGE-L、BLEU-4 变化如图 4 所示.

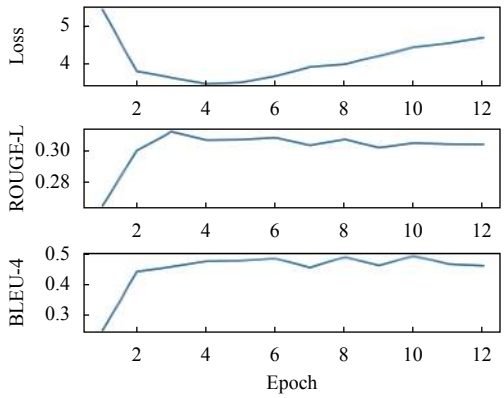

图 4 训练过程中模型在测试集上的 Loss、ROUGE-L、BLEU-4

由图 4 可知, 随着训练的进行, 模型在测试集上的损失逐渐降低, 说明模型正在不断优化, 在第 5 个 epoch 之后, 损失又会变大说明从此时开始模型开始过拟合. 与之对应 ROUGE-L、BLEU-4 指标, 一开始随着 Loss 的减少而增加. 当发生过拟合时, 随着 Loss 的增加有所下降, 因此, 本文选用第 4 个 epoch 的训练结果作为解答器.

3.3.3 问答系统评估

最终将检索器和解答器整合起来构建的完整的问答系统, 首先对问题进行分词. 然后, 将分词后的问题输入检索器, 检索器会检索出与问题最相关的 N 个文档. 将问题与以上 10 个文档分别输入解答器, 得到 N 个答案以及其对应的概率分值, 概率越大则答案越有可能是正确答案, 因此选择概率分值最大的答案作为最终答案. 本文选取了 9000 多个问题对设置了不同

N 的系统效果进行了评估, 结果如图 5所示.

可见 N 为 3 时, 综合 ROUGE-L、BLEU-4 来看系统效果最佳, 说明尽管随着 N 的增加检索器的结果包含正确答案的概率变高了, 但是其结果也包含了更多错误答案, 引入了噪声. 因此最终设置 N 为 3, 此时系统, ROUGE-L 指标为, BLEU-4 指标为. 对比解答器的测试结果有所下降, 这由两个原因导致, 一是检索器不一定找到正确答案所在的文档并提供给解答器; 二是检索器找到正确的文档且解答器找到正确答案的情况下, 存在正确答案的概率值小于错误答案概率值的情况.

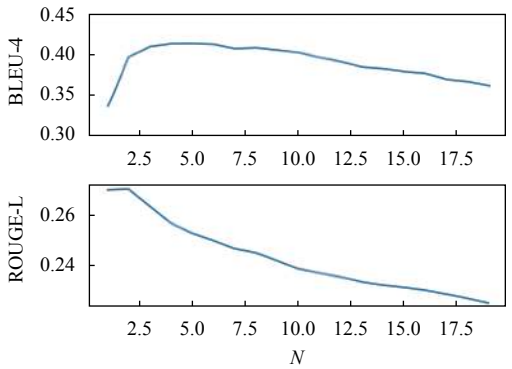

图 5　不同 N 值的系统效果

3.3.4　问答预测

问答系统建立完成之后, 就可以使用其获取信息了. 使用时, 系统输入为用户问题, 输出为系统从文档库中检索并截取的问题答案. 解答时, 系统根据用户提的问题先从文档中检索出可能包含答案的候选文章, 然后使用解答器从候选文章中截取答案, 并选取概率值最大的答案作为系统输出. 随机选取了几个问题及其参考、预测答案作为样例, 如图 6 所示.

{'ques': 'F-35 战机的能力是',
'ref': '为地面部队提供支援',
'pred': 'F-35 战机具现代空战必备的隐身与垂直起降能力'},
{'ques': 'ZTQ 轻型坦克可能比 62 式更重，估计至少多重',
'ref': '30 吨',
'pred': '30 吨'},
{'ques': '但印度意识到，仅仅有一个可靠导弹拦截系统是不够的，还需要什么',
'ref': '还需要一体化的通信和雷达预警系统',
'pred': '一体化的通信和雷达预警系统。'},
{'ques': '此次演习可能以小规模进行的原因？',
'ref': '美国并不准备派遣航空母舰和核潜艇',
'pred': '有效监测并阻止朝鲜潜艇的入侵'},
{'ques': '索马里什么组织有关联的极端组织，近年来在索马里及其邻国多次发动恐怖袭击',
'ref': '“青年党”是与“基地”',
'pred': '“青年党”是与“基地”'}

图 6　样例问题-参考答案-预测答案对比

由图 6 可知, 对于大部分问题模型预测的答案与实际答案基本一致, 也存在有些问题答案不合理的情况. 总之, 本模型可以从文档中截取比较合理的问题答案, 可以有效减少用户获取信息时所需要的时间, 不过模型的精度还有待提高.

3.3.5　系统效果展示

由于本系统对于模型答案的预测还存在一定的误差, 在最终问答界面选取了概率最大的前 5 个答案作为系统最终的输出, 用户可以从中选取所需答案. 系统效果如图 7 所示.

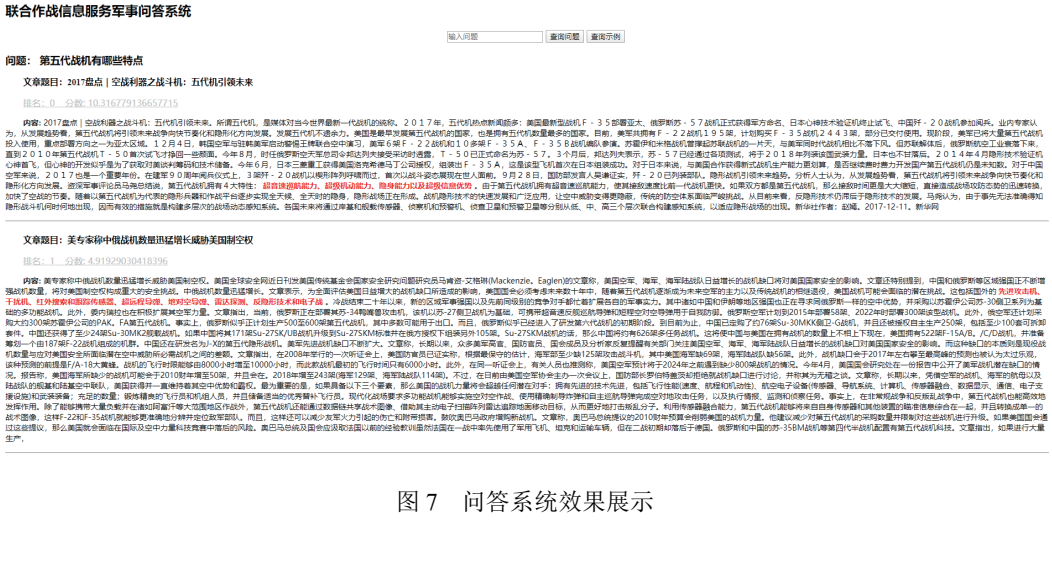

图 7　问答系统效果展示

本系统最终输出了排名靠前的答案、分值以及答案所在的文章, 对答案做了标注, 用户可以直观的获取答案, 也可以对答案的上下文进行进一步的了解.

4 结论

本文对信息服务问答系统进行了研究, 模型基于领域阅读理解数据集构建了基于文档库的面向问题的检索、解读系统. 当用户提出问题时, 可以依次通过信息检索器找到答案所在文档, 并通过解答器找到答案在文档中的位置, 从而直接得到问题对应的答案. 本文的研究对于解决当前海量信息环境下如何快速有效的获取用户所需的信息, 提升综合信息服务用户获取信息的效率具有重要意义.

参考文献

1 王东升, 王卫民, 王石, 等. 面向限定领域问答系统的自然语言理解方法综述. 计算机科学, 2017, 44(8): 1–8, 41. [doi: 10.11896/j.issn.1002-137X.2017.08.001]

2 Dong L, Wei FR, Zhou M, *et al*. Question answering over freebase with multi-column convolutional neural networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing. Beijing: Association for Computational Linguistics, 2015. 260–269.

3 Rajpurkar P, Zhang J, Lopyrev K, *et al*. SQuAD: 100, 000+ questions for machine comprehension of text. Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing. Austin: Association for Computational Linguistics, 2016. 2383–2392.

4 Wang SH, Jiang J. Machine comprehension using Match-LSTM and answer pointer. Proceedings of the 5th International Conference on Learning Representations. arXiv: 1608.07905v2, 2016.

5 Yu AW, Dohan D, Luong MT, *et al*. Qanet: Combining local convolution with global self-attention for reading comprehension. Proceedings of the 6th International Conference on Learning Representations. arXiv: 1804.09541v1, 2018.

6 Chen DQ, Fisch A, Weston J, *et al*. Reading wikipedia to answer open-domain questions. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 1870–1879.

7 Salton G, Yu CT. On the construction of effective vocabularies for information retrieval. ACM Sigplan Notices, 1975, 10(1): 48–60. [doi: 10.1145/951787.951766]

8 Pedregosa F, Varoquaux G, Gramfort A, *et al*. Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, 2011, 12: 2825–2830.

9 张振亚, 王进, 程红梅, 等. 基于余弦相似度的文本空间索引方法研究. 计算机科学, 2005, 32(9): 160–163. [doi: 10.3969/j.issn.1002-137X.2005.09.041]

10 Weissenborn D, Wiese G, Seiffe L. FasTQA: A simple and efficient neural architecture for question answering. arXiv: 1703.04816, 2017.

11 Seo M, Kembhavi A, Farhadi A, *et al*. Bidirectional attention flow for machine comprehension. Proceedings of the 5th International Conference on Learning Representations. arXiv: 1611.01603v6, 2018.

12 Lin CY. Rouge: A package for automatic evaluation of summaries. Barcelona: Association for Computational Linguistics, 2004. 74–81.

13 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: Association for Computational Linguistics, 2002. 311–318.