计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(4):175−180 [doi: 10.15888/j.cnki.csa.007846]

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

基于正负样本和 Bi-LSTM 的文本相似度匹配模型①

周艳平, 朱小虎

(青岛科技大学 信息科学技术学院, 青岛 266061)
通讯作者: 朱小虎, E-mail: 206302630@qq.com

摘　要: 相似度匹配是自然语言处理领域一个重要分支, 也是问答系统抽取答案的重要途径之一. 本文提出了一种基于正负样本和 Bi-LSTM 的文本相似度匹配模型, 该模型首先为了提升问题和正确答案之间的相似度, 构建正负样本问答对用于模型训练; 其次为了解决分词错误引起的实验误差, 采用双层嵌入词向量方法进行预训练; 再次为了解决注意力机制导致的特征向量向后偏移的问题, 在特征提取之前, 采取内部注意力机制方法; 然后为了保留重要的时序特性, 采用 Bi-LSTM 神经网络进行数据训练; 最后为了能在语义层次上计算相似度, 提出一种包含语义信息的相似度计算函数. 将本文提出的文本相似度匹配模型在公共数据集 DuReader 上进行了仿真实验, 并和其他模型进行对比分析, 实验结果表明, 提出的模型不仅准确率高且鲁棒性好, top-1 准确率达到 78.34%.

关键词: 问答系统; 相似度匹配; 正负样本; Bi-LSTM

引用格式: 周艳平,朱小虎.基于正负样本和 Bi-LSTM 的文本相似度匹配模型.计算机系统应用,2021,30(4):175–180. http://www.c-s-a.org.cn/1003-3254/7846.html

Text Similarity Matching Model Based on Positive and Negative Samples and Bi-LSTM

ZHOU Yan-Ping, ZHU Xiao-Hu

(College of Information Science and Technology, Qingdao University of Science & Technology, Qingdao 266061, China)

Abstract: Similarity matching is crucial for natural language processing and also for extracting answers from the question answering system. This study proposes a model of text similarity matching based on positive and negative samples and Bi-LSTM. Firstly, this model constructs question answering pairs for positive and negative samples in model training, improving the similarity between a question and its correct answer. Secondly, it applies the dual-layer word vector embedding for pre-training to solve the experimental error caused by segmentation mistakes. Thirdly, it adopts the internal attention mechanism before feature extraction to solve the backward offset of the characteristic vectors caused by the attention mechanism. Then this model trains the data on the Bi-LSTM neural network to retain important temporal characteristics. Finally, it puts forward a similarity calculation function including semantic information to calculate similarity at the semantic level. The model proposed in this study is simulated on the public data set DuReader and compared with other models. The experimental results show that the proposed model has high accuracy and good robustness, and the accuracy of top-1 reaches 78.34%.

Key words: question answering system; similarity match; positive and negative samples; Bi-LSTM

① 基金项目: 国家自然科学基金 (61402246); 山东省高等学校科技计划 (J14LN31)
Foundation item: National Natural Science Foundation of China (61402246); Science and Technology Plan of Colleges and Universities in Shandong Province (J14LN31)
收稿时间: 2020-07-30; 修改时间: 2020-08-26; 采用时间: 2020-09-01; csa 在线出版时间: 2021-03-30

随着自然语言处理技术的快速发展, 问答系统已经成为人工智能的前沿领域[1], 例如小米公司的“小爱同学”、苹果公司的“Siri”, 它们能够为用户提供良好的人机交互体验. 相似度匹配[2] 是问答系统抽取答案的重要途径之一, 抽取答案的准确性决定了一个问答系统的质量[3].

Kumar 等[4] 通过 DMN (动态内存网络) 构造了一个改进的问答系统, 该系统主要用于处理输入序列并进行训练. Wang 等[5] 提出了一种基于注意力机制的 Bi-GRU-CapsNet 模型, 该模型采用了一种新的“向量输入、输出”传递方案, 其中神经元的输入和输出是向量. Santos 等[6] 提出了一个具有特征权重的问题和答案的注意力集中双向注意力机制 (Attentive pooling). Peters 等[7] 提出了一种新的深层语境化单词表示方法 ESIM + ELMo, 其中词向量是学习深度双向语言模型 (biLM) 内部状态的函数. Zhou 等[8] 提出了一种多视图响应选择模型 (Multiview), 该模型集成了来自两个不同视图 (单词序列视图和话语序列视图) 的信息. 尽管上述方法注意力机制等方法提高了问答匹配的准确性, 但包含语义层次信息的相似度匹配和中文分词错误仍然没有解决.

根据实验研究, 本文针对问答系统中的上述问题提出一种基于正负样本和 Bi-LSTM[9] 的文本相似度匹配模型 (PN-Bi-LSTM), 该模型不仅解决了包含语义层次信息的相似度匹配和中文分词错误造成的问题, 还提高了中文问答系统问答匹配的准确性.

1 模型框架

为了最大化问题与正确答案之间的相似度, 并且与错误答案之间的相似度最小, 构建的数据集中, 问答对存在的形式如表 1 所示.

表 1 数据集中问答对的形式

(Q)	青岛的夏天有哪些好玩的地方?
(A^+)	八大关的景色不错, 适合照相. 八大关有万国建筑博览会之说.
(A^-)	建议买书来学习比较好, 书上有习题, 通过做练习的方式来掌握NLP的基本技术.
(A)	一个英明果断的决策者; 团队具有向心力、凝聚力. 这两点就够了.

Q 是一个问题陈述, A^+是正确答案, A^-是错误答案. 通过神经网络计算每个句子的特征, 然后输出问题和答案之间的相似度差. 目标函数是保证相似区间最大. 当用户输入问题时, 系统将输出最合适的答案. 本文采用正负答案样本训练神经网络模型, 模型输入是问题和正负答案样本的代表向量. 我们需要截断或补充问答语句, 使句子长度一致并用于神经网络的训练.

问答系统中的句子分词错误会对实验结果产生很大的影响, 使用双层嵌入向量[10] 表示方法, 可以有效地减少分词引起的实验误差.

另外, 在提取句子特征之前, 我们采用了内部注意力机制 (IARNN)[11], 避免了特征的向后偏移力问题. 然后, 将注意机制处理的时间序列信息输入到 Bi-LSTM 模型中, 通过 LSTM[12] 选择序列特征.

在问答对匹配过程中, 对于给定的问题 (Q) 和答案池$\{a_1, a_2, \cdots, a_m\}$($m$ 是答案池中的答案数, 且至少包括一个正确答案), 则需要检索答案池中与问题 (Q) 相关的正确答案 (a_n)($1 \le n \le m$). 因此需要计算 Q 与每个候选答案之间的相似度, 并将最相似的候选答案记录为最佳答案. 如果最佳答案恰好是在基本事实中, 则该问题的答案将被成功地检索出来, 并算做 top-1 准确率[13].

我们提出模型的总体框架如图 1 所示.

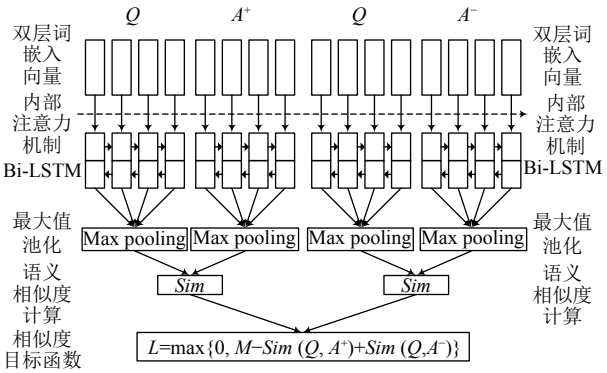

图 1 模型总体框架

2 主要方法

2.1 双层嵌入向量表示

问答系统中的句子的向量表示是文本特征生成的重要步骤. 利用 LSTM 神经网络处理匹配任务时需要获得句子的向量表示, 而句子分词错误会对实验结果产生很大的影响, 因此采用双层嵌入向量模型表示方法, 可以有效地减少分词引起的实验误差. 双层嵌入向量模型如图 2 所示.

如图 2 所示, 在对所有的问答句子进行分词后, 通过 Word2Vec[14] 模型进行单词和字符向量训练, 得到

所有单词和字符的训练模型. 利用单词嵌入和字符嵌入模型, 得到了字符向量和单词向量. 最后, 将句子中的字符向量和单词向量进行策略性组合, 得到每个句子的向量.

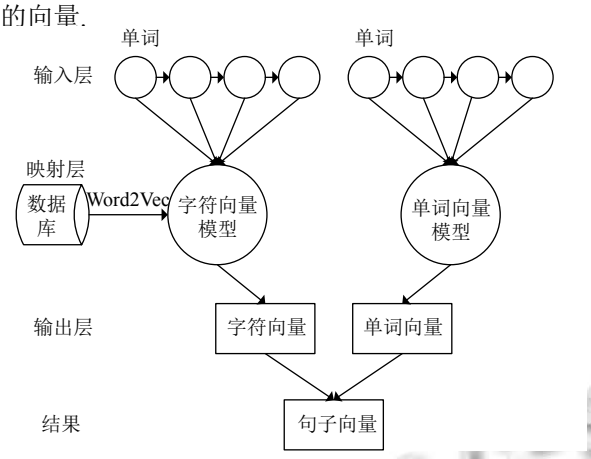

图 2 双层嵌入向量模型

由于单词嵌入和字符嵌入长度不一致, 我们首先采用零向量来补充单词嵌入, 再加权单词和字符的向量表示, 改进模型的最终句子向量由 *Sen* 表示:

$$Sen = \alpha * Sen_{\text{word}} + \beta * Sen_{\text{character}} \tag{1}$$

其中, Sen_{word}表示单词嵌入向量表示, $Sen_{\text{character}}$表示字符嵌入向量表示. α与β之和为常数 1, 本文将α设为 0.6. 通过双层嵌入将句子表示为 100 维向量, 然后通过内部注意力机制提取句子向量的特征.

2.2 内部注意力机制

句子中的单词之间可能存在协同效应, 这会降低测试集中模型的准确性. 由于 RNN[15] 注重时序性, 所以 t 时刻的神经网络模型包含了所有先前时刻的序列信息. 在 RNN 框架中加入注意力机制以获得更多的加权信息.

由于框架中包含了更多的前向信息, 因此会选择靠近句尾的文本特征, 从而导致特征向后偏移和权重偏差. 为了解决上述问题, 在特征提取之前, 采用内部注意力机制. 在计算句子时间信息方面过程时, 内部注意力机制结构如图 3 所示.

如图 3 所示, 在 LSTM 训练之前, 注意力机制提取了表示句子的x_t的时间信息. 该算法将每次的平均特征输出作为最后一次输出, 避免了特征信息的丢失. 此过程中进行了最大值池化操作, 这使每个时刻都会增加注意力机制的权重. 在注意力机制计算α_t后, 我们得到如下输出$\widetilde{x}_t$:

$$\widetilde{x}_t = \alpha_t * x_t \tag{2}$$

其中, x_t是时间t处的原始输入时序特征向量, α_t定义如下:

$$\alpha_t = \sigma\left(r_q^{\mathrm{T}} M_{qi} x_t\right) \tag{3}$$

其中, σ是一个 Sigmoid 函数, 因此α_t的值介于 0 和 1 之间; r_q是关于注意力机制的隐藏层的权重; M_{qi}是一个注意力矩阵, 它将问答句子转换为单词嵌入空间.

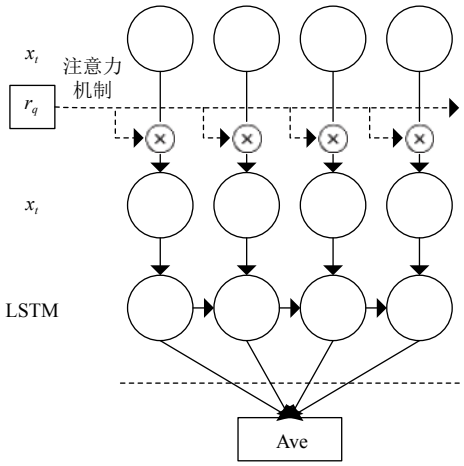

图 3 内部注意力机制结构

2.3 Bi-LSTM 神经网络模型

RNN 是一种能够存储历史状态的时间序列网络结构. 然而, 由于梯度爆炸和梯度消失, 多层 RNN 在计算上下文信息时往往会受到限制. LSTM 是 RNN 的一种变体, 主要解决 RNN 长距离梯度计算的问题. 在 LSTM 结构中, 隐藏层向量为h_t时, 时刻t的状态更新如下:

$$\begin{cases} i_t = \sigma(W_i x(t) + U_i(t-1) + b_i) \\ f_t = \sigma\left(W_f x(t) + U_f(t-1) + b_f\right) \\ o_t = \sigma(W_o x(t) + U_o(t-1) + b_o) \\ \widetilde{C}_t = \tanh(W_c x(t) + U_c(t-1) + b_c) \\ C_t = i_t * \widetilde{C}_t + f_t * C_{t-1} \\ h_t = o_t * \tanh(C_t) \end{cases} \tag{4}$$

其中, i_t, f_t, o_t, C_t分别是输入门的输出值、遗忘门的输出值、输出门的输出值和存储单元的输出值, σ是 Sigmoid 函数; W, U, R是 LSTM 神经网络的参数.

Bi-LSTM 可以解决单向 LSTM 无法计算逆序上下文信息的问题. 将正向序列和反向序列组合以获得输出:

$$r_t = \overrightarrow{h}_t \| \overleftarrow{h}_t \tag{5}$$

其中, $\overrightarrow{h}$和$\overleftarrow{h}$分别是 Bi-LSTM 正向和反向隐藏层的计算结果. Bi-LSTM 在两个方向上进行了计算和更新. 这

种双向 LSTM 结构的两个训练序列直接连接到输出层, 为每个单词提供完整的上下文状态信息.

2.4 目标函数与相似度计算

训练后的神经网络模型能最大化问题与正确答案之间的相似度, 最小化问题与错误答案之间的相似度. 目标函数是使正样本和负样本之间的差异最大化. 其他问答系统一般只计算向量间的余弦相似度, 而不涉及语义层面的深度相似度计算, 这有相当大的局限性. 因此, 我们提出一种包含语义的相似度计算[16] 来定义一个目标函数:

$$L=\max\{0,M-Sim(Q,A^{+})+Sim(Q,A^{-})\} \quad (6)$$

其中, M 为最大区间值, 取值为 0.1, Sim 为问答语句的语义和文本联合相似度计算方法, 定义如下:

$$Sim(Q,A)=\theta_1*Sim_{\text{semantic}}(Q,A)+\theta_2*Sim_{\text{text}}(Q,A) \quad (7)$$

其中, Sim_{text}是向量的余弦相似度计算方法, Sim_{semantic}是向量的语义相似度计算方法. θ_1与θ_2之和是常数 1, 本文设置θ_1为 0.6. 语义相似度计算方法的简化过程如图 4 所示.

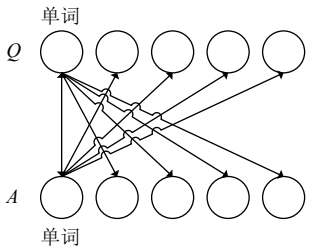

图 4 语义相似度计算过程

两行圆形分别表示问答语句, 每个圆形代表一个单词. 语义相似度计算方法解释为: 问句 Q 中有 m 个词向量, 分别是$\{q_1, q_2,\cdots, q_m\}$; 答案语句 A 中有 n 个词向量, 分别是$\{a_1, a_2,\cdots, a_n\}$. 首先计算量 q_1 和 a_1–a_n 之间的余弦相似度, 记录 q_1 和 a_1–a_n 之间相似度最大的相似度值, 同理计算 q_2–q_m 和 a_1–a_n 之间最大相似度值, 然后再计算问句 Q 中所有词向量的最大相似度值之和.

同理对于答案语句 A, 计算出 A 中所有词向量最大相似度值之和, 最后将两个最大相似度相加除以两个句子的长度之和, 得到 Q 和 A 之间的语义相似度, 解释如下:

$$Sim_{\text{semantic}}=\frac{\sum_{i=1}^{m}Q_{\max}+\sum_{i=1}^{n}A_{\max}}{m+n} \quad (8)$$

式中, $Q_{\max}$为问题句中每个词的最大相似度之和, $A_{\max}$为回答句中每个词的最大相似度之和:

$$\begin{aligned}Q_{\max}&=Max(\cos(q_i,a_1),\cdots,\cos(q_i,a_n))\\A_{\max}&=Max(\cos(q_1,a_i),\cdots,\cos(q_m,a_i))\\Sim_{\text{text}}(q,a)&=\cos(q,a)=\frac{\|q\cdot a\|}{\|q\|\cdot\|a\|}\end{aligned} \quad (9)$$

为了避免局部最优解的问题, 我们选择 Adam 作为优化器. 在 Bi-LSTM 层中, 我们添加 Dropout[17] 机制来避免过拟合问题.

3 实验及结果

3.1 实验数据集

本文使用的数据集是公共数据集 DuReader[18], 并提取了其中 50 000 个问题样本和 90 563 个答案样本. 问题陈述的平均长度为 60 个字符, 回答语句的平均长度为 80 个字符. 在整个答案库中, 每个问题平均有 2 个正确答案. 在训练集中, 我们选择 4 万个问题组成 24 万个训练样本, 其中 4 万个是正样本, 20 万个是负样本, 每个问题有 1 个正样本和 5 个负样本. 正样本是一个问题和它的正确答案的配对. 负样本是一个问题和从 90 563 个答案中随机抽取一个错误答案的配对. 在测试集中, 剩余的 10 000 个问题被用来构建 100 万个样本, 其中 1 万个是正样本, 99 万个是负样本, 每个问题有一个正样本和 99 个负样本. 我们将每个问题的答案池大小设置为 100, 并根据每个答案池记录 top-1 的准确率. 我们采用 top-k 准确率和训练、测试集的损失作为模型的评价标准.

3.2 实验设置

本文提出的模型是用 Python 语言和 TensorFlow[19] 神经网络框架实现的. 使用 Jieba 和 Gensim 工具进行分词和词向量预训练. 单词向量预训练窗口设置为 5, 向量维数设置为 100. 此外, 我们将问题语句的长度设置为 60, 将答案语句的长度设置为 80. 目标函数的最大区间值 M 设为 0.1.

在神经网络超参数设置方面, 我们选择 Adam 作为 Bi-LSTM 网络的优化器, 并将 LSTM 层数设为两层. 我们的 Dropout 参数的值设置为 0.5, 隐层节点数设置为 200, 学习率设置为 0.1, LSTM 输出特征通过最大池化层进行选择.

3.3 实验结果

在词向量的预训练阶段, 我们对字符嵌入、单词嵌入和双层嵌入进行了一系列比较实验. 双层嵌入在

训练集上的准确度比其他方法高 1~4 个百分点, 且损失函数性能更好, 在测试集上的准确度比其他方法高 1~2 个百分点. 因此, 改进的双层嵌入可以解决分词导致的误差和序列信息丢失的问题. 实验结果如表 2 所示.

表 2 预训练方法实验对比

预训练方法	Train acc (%)	Train loss	Test acc (%)
双层嵌入	84.19	1.06	78.06
字符嵌入	80.56	1.46	77.24
单词嵌入	83.35	1.39	76.84

在相似度计算阶段, 语义相似度计算方法在训练集上比余弦相似度计算方法高 2 个百分点, 在测试集上高出 7 个百分点, 所以语义相似度计算方法优于余弦相似度计算方法, 实验结果如表 3 所示.

表 3 不同相似度计算方法比较

相似度	Train acc (%)	Train loss	Test acc (%)
语义相似度	83.95	0.864	78.86
余弦相似度	81.66	0.639	71.76

本文提出的模型 PN-Bi-LSTM 与其他现有的方法相比具有很大的优势良好的性能. 实验结果如表 4 所示.

表 4 各模型 Top-1 准确率和损失对比

模型	Train acc (%)	Train loss	Test acc (%)
DMN	75.24	0.92	74.38
CapsNet	82.43	1.41	77.19
Attentive pooling	85.25	1.68	73.42
ESIM + ELMo	79.33	1.24	76.65
Multiview	79.14	1.31	74.85
CapsNet	81.83	1.42	76.21
PN-Bi-LSTM	83.94	0.98	78.34

为了验证 PN-Bi-LSTM 在不同应用需求下的有效性, 我们在测试集上采用 *F*1 值[20]、召回率、top-2 准确率、top-3 准确度作为我们的性能指标. PN-Bi-LSTM 和其他 6 种比较模型的实验结果见表 5.

表 5 模型在测试集上性能指标对比

模型	*F*1 (%)	Recall (%)	Top-2 (%)	Top-3 (%)
DMN	74.64	75.23	76.37	80.16
CapsNet	77.83	79.15	80.24	83.04
Attentive pooling	73.25	73.53	76.16	77.32
ESIM + ELMo	77.41	77.93	79.04	80.18
Multiview	75.84	76.34	78.42	80.16
PN-Bi-LSTM	78.94	80.16	82.43	83.96

如表 5 所示, 无论 top-*k* 准确率的 *k* 值如何, PN-Bi-LSTM 的性能都优于其他几种模型. PN-Bi-LSTM 在 *F*1 和召回率下也表现良好, 表明我们提出的方法在不同的性能指标下都是有效.

为了验证 PN-Bi-LSTM 的有效性, 我们提出了一种新的评价标准. 当一个模型在训练集上 top-1 准确率第一次达到 50% 时, 记录训练步数. 使用较少步数的模型可以更快地从问答语句中提取有用的信息. 表 6 显示了模型首次达到 50% 准确率时所采取的步骤数.

表 6 准确率达到 50% 所用的步数

模型	DMN	CapsNet	ESIM + ELMo	PN-Bi-LSTM
步数	468	167	189	103

如表 6 所示, PN-Bi-LSTM 可以用最少的步数 top-1 达到 50% 的准确率. 从这个角度看, 该方法是有效的, 对问答语句有较高的敏感性.

4 结论与展望

本文基于正负样本, 提出了一个包含语义信息的双层嵌入 Bi-LSTM 模型,该模型大大提高了中文问答匹配的准确性.

实验结果表明, 本文提出的方法模型优于其他几种问答方法. 在测试集上 top-1 的准确度可达 78.34%, 在训练集上损失可降至 0.98. 此外, 我们采用 F1 值、召回率和 top-*k* 准确率来验证 PN-Bi-LSTM 的有效性, 实验结果表明, PN-Bi-LSTM 在不同的性能指标下具有鲁棒性并且是有效的. 最后, 我们提出了一个新的性能指标来验证 PN-Bi-LSTM 在语句信息提取方面比其他几种方法更有效. 因此, 本文的研究具有应用和实用价值.

在未来, 我们将进一步使用不同问答系统评估提出的模型, 例如基于文章内容的答案预测. 此外, 我们将增加数据量, 来进一步验证 PN-Bi-LSTM 在不同数据集上的性能.

参考文献

1 桑瑞婷. 面向高校迎新的机器人问答系统研究 [硕士学位论文]. 重庆: 重庆理工大学, 2019.
2 卢超. 基于深度学习的句子相似度计算方法研究 [硕士学位论文]. 太原: 中北大学, 2019.
3 徐雄. 基于深度学习的问答系统研究. 湖北师范大学学报(自然科学版), 2019, 39(1): 10–18.
4 Kumar A, Irsoy O, Ondruska P, *et al*. Ask me anything: Dynamic memory networks for natural language processing. Proceedings of the 33rd International Conference on

International Conference on Machine Learning. New York City, NY, USA. 2016. 1378–1387.
5 Wang Q, Xu CM, Zhou YM, *et al*. An attention-based Bi-GRU-CapsNet model for hypernymy detection between compound entities. Proceedings of 2018 IEEE International Conference on Bioinformatics and Biomedicine. Madrid, Spain. 2018. 1031–1035.
6 Dos Santos C, Tan M, Xiang B, *et al*. Attentive pooling networks. arXiv: 1602.03609, 2016.
7 Peters M E, Neumann M, Iyyer M, *et al*. Deep contextualized word representations. Proceedings of 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. New Orleans, LA, USA. 2018. 2227–2237.
8 Zhou XY, Dong DX, Wu H, *et al*. Multi-view response selection for human-computer conversation. Proceedings of 2016 Conference on Empirical Methods in Natural Language Processing. Austin, TX, USA. 2016. 372–381.
9 Huang ZH, Xu W, Yu K. Bidirectional LSTM-CRF models for sequence tagging. arXiv: 1508.01991, 2015.
10 李琳, 李辉. 一种基于概念向量空间的文本相似度计算方法. 数据分析与知识发现, 2018, 2(5): 48–58.
11 Wang BN, Liu K, Zhao J. Inner attention based recurrent neural networks for answer selection. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany. 2016. 1288–1297.
12 Graves A. Supervised Sequence Labelling. Berlin, Heidelberg: Springer. 2012. 5–13.
13 Adomavicius G, Zhang JJ. Classification, ranking, and top-k stability of recommendation algorithms. INFORMS Journal on Computing, 2016, 28(1): 129–147. [doi: 10.1287/ijoc.2015.0662]
14 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv: 1301.3781, 2013.
15 İrsoy O, Cardie C. Deep recursive neural networks for compositionality in language. Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, QC, Canada. 2014. 2096–2104.
16 周艳平, 李金鹏, 蔡素. 基于同义词词林的句子语义相似度方法及其在问答系统中的应用. 计算机应用与软件, 2019, 36(8): 65–68.
17 Srivastava N, Hinton G, Krizhevsky A, *et al*. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 2014, 15(1): 1929–1958.
18 He W, Liu K, Liu J, *et al*. Dureader: A chinese machine reading comprehension dataset from real-world applications. Proceedings of the Workshop on Machine Reading for Question Answering. Melbourne, Australia. 2017. 37–46.
19 Abadi M, Agarwal A, Barham P, *et al*. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv: 1603.04467, 2016.
20 赵明, 董翠翠, 董乔雪, 等. 基于 BIGRU 的番茄病虫害问答系统问句分类研究. 农业机械学报, 2018, 49(5): 271–276.