文章编号：1003-0077(2019)01-0118-07

融合深度匹配特征的答案选择模型

冯文政，唐 杰

（清华大学 计算机科学与技术系，北京 100084）

摘 要：答案选择是自动问答系统中的关键任务之一，其主要目的是根据问题与候选答案的相似性对候选答案进行排序，并选择出相关性较高的答案返回给用户。可将其看作成一个文本对的匹配问题。该文利用词向量、双向LSTM、2D 神经网络等深度学习模型对问题—答案对的语义匹配特征进行了提取，并将其与传统 NLP 特征相结合，提出一种融合深度匹配特征的答案选择模型。在 Qatar Living 社区问答数据集上的实验显示，融合深度匹配特征的答案选择模型比基于传统特征的模型 MAP 值高 5%左右。

关键词：问答系统；答案选择；深度匹配模型

中图分类号：TP391 **文献标识码**：A

A Ranking Model for Answer Selection with Deep Matching Features

FENG Wenzheng, TANG Jie

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: Answer Selection is one of the key tasks in question answering system. Its main purpose is to rank the candidate answers according to the similarity between the questions and the candidate answers and select the more relevant answers to users. It can be seen as a text pair matching problem. In this paper, we use the deeplearning model such as word embedding, bidirectional LSTM, 2D neural network and so on to extract the semantic matching features for question-answer pairs, and incorperate these into a ranking model together with traditonal NLP features. The experiments on the Qatar Living community question answering data show that the answer selection model with deep matching features is about 5% higher than only using traditional features on the MAP values.

Keywords: question answering; answer selection; deep matching model

0 引言

答案选择是自动问答系统中的关键问题。其主要任务是给定一个问题 q 和一组候选答案集合 $C=\{a_i \mid i=1,2,\cdots,n\}$，计算 q 与每个候选答案 a_i 的相似度，并根据相似度对所有的候选答案进行排序。可以将其看做为一种文本匹配问题，即如何计算两个文本之间的相关度。

由于自然语言的复杂性，计算文本之间的匹配程度存在着多项挑战[1]，主要有：1）词语语义的多元性。不同的词语可能表示相同的语义如“荷花”、“莲花”都是表示一种植物，同理一个相同的词在不同的语境下会有不同的语义，例如，“苹果”既可以是一种水果也可以是一家公司。2）短语匹配的结构性，多个词语可以按照一定的结构组合成短语，匹配两个短语需要考虑短语的结构信息。例如，“机器学习”和“机器学习”是两个词之间的顺序匹配，而“机器学习”和“学习机器”只有词语是匹配的，而顺序是打乱的。这两种情况的匹配程度是不一样的。3）文本匹配的层次性。文本是以层次化的方式组织起来的，词语组成短语，短语再组成句子，这导致两个相似问题往往会具有不同的语法或结构特点。这样的特性使得我们在考虑文本匹配时要考虑不同层次的匹配信息。

为了解决上述三种挑战，我们提出了一种融合深度匹配特征的排序模型。模型除了利用传统自然语言特征之外，还使用双向 LSTM 模型、2D 神经网

收稿日期：2017-03-16 **定稿日期**：2017-04-26

络模型等深度匹配模型对文本对之间的语义匹配特征进行抽取。在 Qatar Living 社区问答数据集中的实验结果显示，融合深度匹配特征的答案选择模型比基于传统特征的模型 MAP 值高 5%左右。

1　模型描述

如图 1 所示，整个模型分为三个部分：文本预处理，特征抽取和特征融合。首先，对原始文本进行预处理；随后对处理后的文本进行特征抽取，模型从传统自然语言处理模型、深度匹配模型两个方面进行特征抽取；最后将这些特征进行融合，将抽取出来的特征利用 XGBoost 排序框架[2] 进行排序。下面本文对这三部分做具体描述。

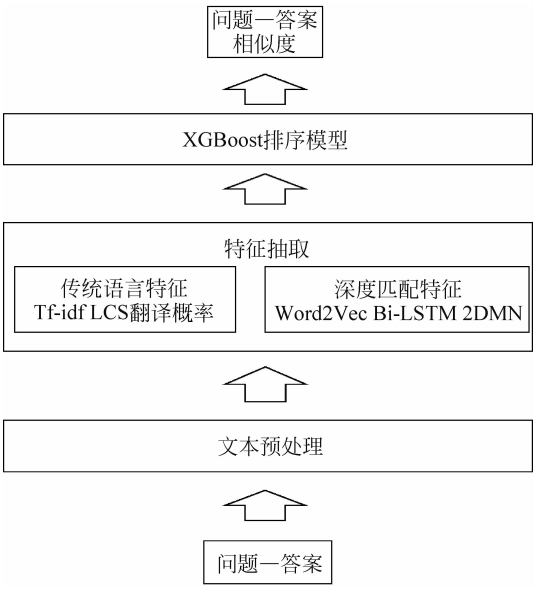


图 1　模型结构图

1.1　文本预处理

在文本预处理过程中，首先去除停用字，并将所有字母换成小写，然后对每个句子进行句法分析，获取所有句子的句法树。我们使用 NLTK[3] 工具包进行文本分词，使用斯坦福大学 PCFG 文法解析器[4] 进行句法解析。

1.2　传统自然语言特征

针对答案选择任务，我们首先使用了一些传统的自然语言处理模型进行了词法分析和句法分析，并抽取了关于词语和句子结构的特征。

1.2.1　Tf-idf 余弦相似度

Tf-idf(term frequency-inverse document frequency)[5] 是一种常用的词语加权技术，可以评估词语在一个语料库中的重要程度，广泛应用于信息检索系统中。Tf-idf 加权技术主要从两方面考虑：词频(term frequency，tf)和逆文档频率(inverse document frequency，idf)。

词频被定义为一个单词 w 在文档 D 中出现的次数，如式(1)所示。

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

其中，$\mathrm{tf}_{i,j}$ 表示文档 d_j 中单词 w_i 的词频值：分子 $n_{i,j}$ 是单词 w_i 在文档 d_j 中出现的次数；分母表示文档 d_j 出现的所有单词的总次数，其主要用来进行归一化以防止词频值偏向于较长的文档。

相对于词频用来衡量单个文档中词语的重要性，逆文档频率则用来衡量多个文档中词语的普遍性。假如一个单词在过多的文档中都有出现，则说明这个词比较常见，重要程度就会偏低，如式(2)所示。

$$\mathrm{idf}_i = \log \frac{|D|}{|\{d_j : w_i \in d_j\}| + 1} \tag{2}$$

其中，分子 $|D|$ 为文档库中的所有文档数量，分母 $|\{d_j : t_i \in d_j\}|$ 为所有文档中包含单词 w_i 的数量，分母加 1 是为了防止除数为 0。

从中可以看出，文档中一个单词的重要程度与词频和逆文档频率分别成正比。因此，单词 w_i 在文档 d_j 中的 Tf-idf 权值可以定义为：

$$\mathrm{tfidf}_{i,j} = \mathrm{tf}_{i,j} \times \mathrm{idf}_i \tag{3}$$

我们将每个文本当作一个文档 d_j，所有文本集合当作文档集合 D。利用 Tf-idf 模型，即为文本中的每一个单词计算 Tf-idf 权值。因此每个文本对 (q,a) 就可以用一个 One-Hot 权值向量 (v_q, v_a) 来表示。通过计算两个向量间的余弦 $cos(v_q, v_a)$ 即可得到文本对之间关于重要单词覆盖的相似度，一定程度上反应了文本之间的相似程度。

1.2.2　最长公共子序列

若一个序列 S 分别是两个序列 (q,a) 的子序列，且是所有符合条件的序列中最长的，则称 S 为 q 和 a 的最长公共子序列(longest common subsequence，LCS)[6]，最长公共子序列经常被用于两个字符串之间的相似度衡量。本文将最长公共子序列应用于衡量两个文本序列之间的词法相似度，即将一个句子看作一个序列，序列的元素为每个句子中的所有单词。最长公共子序列可以使用动态规划(dynamic programming，DP)的方法求解。对于文

本对(q,a),其 LCS 算法描述如下:

(1) 输入文本对 (q,a),其中 $q=\{q_1,q_2,\cdots,q_N\}$,$a=\{a_1,a_2,\cdots,a_M\}$

(2) 定义函数 same(x,y): if x == y,return 1,else return 0

(3) 初始化二维数组 LCS[i,j],表示 q 中第 i 位和 a 中第 j 位之间的最长公共子序列,其中,LCS[1,1] = same (q_1,a_1)

(4) For i = 1 to N,do:

lcs[i,j]=max(lcs[$i-1$,$j-1$]+same(q,i,$a-j$),lcs[$i-1$,j],lcs[i,$j-1$])

endFor

(5) 输出 $lcs[N,M]$

为了避免 LCS 的值偏向比较长的文本,本文中的模型在计算出两个文本的 LCS 之后还要用两个文本的最大长度进行归一化,最终的 LCS 相似性如式(4)所示。

$$\mathrm{LCS}_{score}(q,a)=\frac{\mathrm{LCS}(q,a)}{max\ (|q|,|a|)} \tag{4}$$

1.3 深度匹配特征

传统的自然语言处理模型主要从词语和句子结构的角度对文本之间的相似关系进行建模,不能充分考虑两个句子之间的语义联系。因此,除了上述传统模型之外,我们还利用词向量技术和深度匹配模型对文本对之间的语义联系进行了建模,抽取了句子之间的语义特征。

1.3.1 词向量余弦相似度

词向量(word emdedding)技术可以把每个单词映射为连续空间中的向量,词语之间的语义相似度可以用词向量之间的余弦相似度表示。本文采用 Google 开发的 Word2Vec[7-8] 在 Qatar Living 数据集上训练得到每个单词的词向量,设置向量维度为 200。

得到每个词向量之后,下一步就是利用词向量计算出句子之间的相似度,在这里本研究先采用了一个简单的词袋模型(Bag of Word)得到每个句子的向量表示,即对句子中的每个词向量求平均值,计算如式(5)所示。

$$s=\frac{1}{|s|}\sum_{i=1}^{|s|}s_i \tag{5}$$

其中,s 为句子的向量表示,s_i 为句子中每个词的向量表示。计算出每个句子的相似度之后,便可以采用余弦相似度衡量两个句子之间的语义相似性,对于每个文本对 (q,a),词向量余弦相似度计算方法如式(6)所示。

$$emb_{score}(q,a)=\frac{q_{emb}\cdot a_{emb}}{|q_{emb}|\times|a_{emb}|} \tag{6}$$

这种方法计算简单有效,计算效率高,但不能处理好短语匹配的结构性,文本匹配的层次性等挑战,因此我们还使用了双向 LSTM 模型、2D 神经网络模型对文本匹配进行建模。

1.3.2 双向 LSTM 匹配模型

LSTM(long short term memory)[9] 单元是由 Hochreiter 等提出的一种网络结构,其将传统 RNN 网络中的隐含节点替换成了 LSTM 单元,可以有效避免传统 RNN 模型中的梯度消失等问题。

LSTM 用状态变量 cell 当前时刻的信息,通过"门控"单元控制 cell 中的信息更新,其总共分为三个门:输入门,输出门和遗忘门,"门控"单元一般用一组 sigmoid 单元表示,若函数输出为 1 代表信息完全通过,函数为 0 代表信息没有任何信息通过。Cell 单元更新的公式如式(7)~式(11)所示:

$$i_t=\sigma(W^i X_t+V^i h_{t-1}+b^i) \tag{7}$$

$$f_t=\sigma(W^f X_t+V^f h_{t-1}+b^f) \tag{8}$$

$$o_t=\sigma(W^o X_t+V^o h_{t-1}+b^o) \tag{9}$$

$$c_t=f_t\otimes c_{t-1}+i_t\otimes\tanh(W^c X_t+V^c h_{t-1}+b^c) \tag{10}$$

$$h_t=o_t\otimes\tanh(c_t) \tag{11}$$

其中的 i_t,f_t,o_t,c_t 和 h_t 分别代表在 t 时刻输入门、遗忘门、输出门和 cell 的输出。b^i,b^f,b^o,b^c 为偏置向量,W^i,W^f,W^o,W^c,V^i,V^f,V^o,V^c 为权重矩阵。LSTM 神经网络模型通过使用门限控制单元,使得反向传播时梯度由传统 RNN 的累乘变成了累加,成功避免了梯度消失问题。除此之外,LSTM 表达力强且容易训练,在实际中被广泛应用。

由于单向 LSTM 是将文本中的单词顺序输入的,在训练过程中只能利用之前单词的信息。所以在文本建模任务中,为了充分利用前面单词和后面单词的信息,充分掌握整个句子的语义信息,生成更全面的句子向量表示,我们使用双向 LSTM 型进行特征抽取。双向 LSTM 模型使用两个 LSTM 模型分别对文本进行顺序建模和倒序建模,对两个 LSTM 的隐状态输出连接起来作为整个模型的隐状态输出,整个结构如图 2 所示。

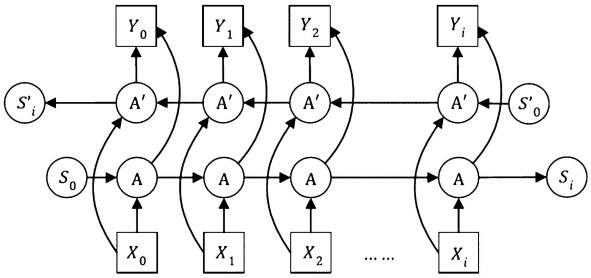


图 2　双向 LSTM 模型架构

如果令 $\overrightarrow{h_t}$ 和 $\overleftarrow{h_t}$ 分别作为 t 时刻正向 LSTM 和反向 LSTM 的隐状态的输出，则 t 时刻双向 LSTM 的隐状态输出 h_t 如式(12)所示。

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{12}$$

这样计算出的 h_t 可以更全面地表示出词语的上下文信息。

我们采用的双向 LSTM 模型具体结构如图 3 所示。对于文本对 (q,a)，首先将句子中每个词的词向量输入到双向 LSTM 中，将两个 LSTM 的隐状态输出连接成为一个向量表示，作为双向 LSTM 的隐含输出，之后将所有单词的隐含输出取平均分别得到两个句子的向量表示。最后将两个句子的向量输入到一个 MLP 模型中进行分类，计算出整个两个文本的相似度。模型采用上节中用的词向量作为输入，使用交叉熵作为代价函数，交叉熵函数定义如式(13)所示。

$$-\sum_{i=1}^{N}[l_i \log(f(s_{x,i}, s_{y,i})) + (1-l_i)\log(1-f(s_{x,i}, s_{y,i}))] \tag{13}$$

其中，$l_i \in \{0,1\}$ 为类别标记，$f(s_{x,i}, s_{y,i})$ 表示需要训练的神经网络模型。

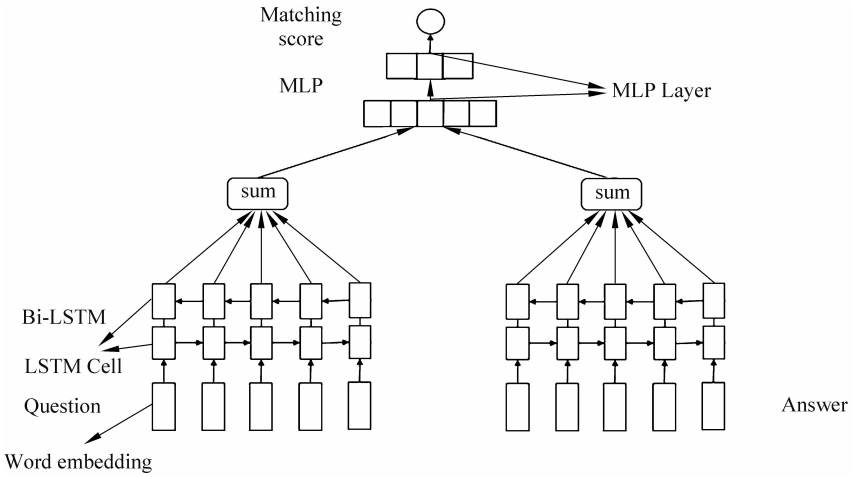


图 3　双向 LSTM 匹配模型

1.3.3　2D 神经网络模型

为了解决文本匹配的层次性，我们还采用 2D 神经网络模型对文本匹配关系进行了建模，利用分层卷积操作层次化地抽取文本对之间的匹配特征。不同于上节的 Bi-LSTM 模型，2D 神经网络借鉴了 MatchPyramid 模型[10]的思想将句子间的交互提前，利用二维卷积神经网络直接在句子的交互空间上进行建模，再对文本间的匹配模式进行分层次抽象。

2DMN 模型四个部分组成：双向 LSTM、匹配矩阵、二维卷积神经网络和多层感知机(MLP)组成，其整个架构如图 4 所示。

给定一个文本对 (S_x, S_y) 及相应的词向量序列 $(\{v_{x,i}\}_{i=1}^{I}, \{v_{y,i}\}_{i=1}^{J})$，首先将词向量序列输入到一个双向 LSTM 模型中，获得两个文本的 LSTM 隐状态表示 $(\{h_{x,i}\}_{i=1}^{I}, \{h_{y,i}\}_{i=1}^{J})$，然后分别根据词向量和 LSTM 隐状态表示计算两个匹配矩阵 M_1 和 M_2。对于 $\forall i,j$，矩阵 M_1 的第 (i,j) 个元素采用如下方式计算如式(14)所示。

$$M_{1,i,j} = v_{x,i}^{T} ? v_{y,j}^{T} \tag{14}$$

M_2 的第 (i,j) 个元素采用如下方式计算：

$$M_{2,i,j} = h_{x,i}^{T} A\, h_{y,j}^{T} \tag{15}$$

其中 A 为参数矩阵。由于 LSTM 隐状态表示中包括词语的上下文信息，故可将其看作为词语在句子中特定语境下的向量表示，因此使用 LSTM 隐状态表示计算出的匹配矩阵可以更精确地捕捉到多义词在特定文本中的匹配关系。计算出匹配矩阵之后，将 M_1 与 M_2 作为两个输入特征面输入到二维卷积神经网络中进行分层卷积和最大池化。设 $z^{(l,f)} = [z_{i,j}^{(l,f)}]_{I^{(l,f)} \times J^{(l,f)}}$ 为第 l 层第 f 个输出特征面，则

$z^{(0,f)}=M_f,\forall f=1,2$。卷积层中的卷积核大小为 $r_w^{(l,f)}\times r_h^{(l,f)}$，则 $z_{i,j}^{(l,f)}$ 可以定义为式(16)。

$$z_{i,j}^{(l,f)}=\sigma\Big(\sum_{f'=0}^{F_{l-1}}\sum_{s=0}^{r_w^{(l,f)}}\sum_{t=0}^{r_h^{(l,f)}}w_{s,t}^{(l,f)}\cdot z_{i+s,j+t}^{(l-1,f')}+b^{l,k}\Big) \quad (16)$$

其中，激活函数 $\sigma(\cdot)$ 采用 ReLU[11] 函数，$w_{s,t}^{(l,f)}\in\mathbb{R}^{r_w^{(l,f)}\times r_h^{(l,f)}}$ 和 $b^{l,k}$ 为 l 层第 f 个特征面的参数，F_{l-1} 为 $l-1$ 层特征面的数目。卷积层之后对输出特征面进行最大池化操作如式(17)所示。

$$z_{i,j}^{(l,f)}=\max_{p_w^{(l,f)}>s\geqslant 0}\ \max_{p_h^{(l,f)}>t\geqslant 0} z_{i+s,j+t} \quad (17)$$

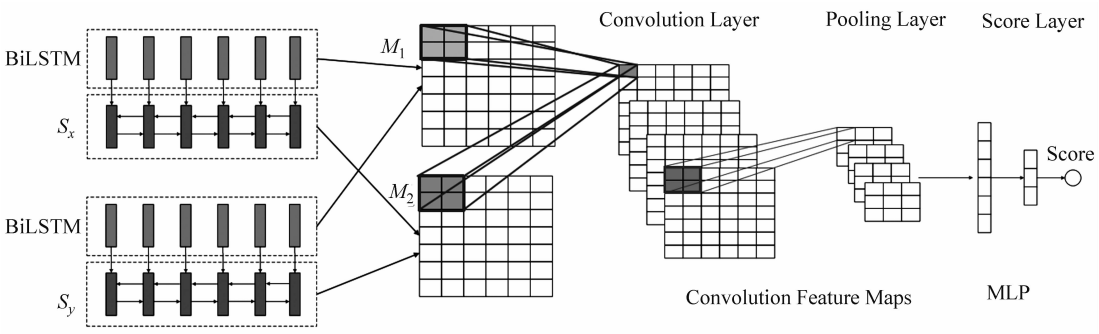


图 4 2D 神经网络模型

经过多层卷积、池化操作之后，最后将最高层的特征向量输入到 MLP 分类器中即可得到最终的匹配得分。

与双向 LSTM 模型类似，模型同样采用 1.1 节中用的词向量作为输入，使用交叉熵作为代价函数进行训练。

1.4 特征融合

对数据集进行特征抽取之后就是将抽取到的不同的特征融合在一起，本文使用 XGBoost 训练了一个基于梯度提升回归树(gradient boosted regression)的回归模型，将所有的特征连接成一个特征向量作为 XGBoost 的输入。训练中选用 pairwise loss 作为目标函数。

2 模型描述与实验分析

2.1 数据集描述与评价标准

本文使用 Qatar Living 社区问答数据集[12] 数据集，其详细统计数据见表 1。

表 1 答案选择数据集

	训练集	测试集
相关问题	6 725	293
答案	43 558	2 930

训练集中总共的“问题—答案”对共有 43 558 个，测试集中共有 2 930 个。对于“问题—答案”对的匹配关系，数据集中共有三种类型。“Good”，“PotentiallyUseful”，“Bad”三种类型，在实际应用中，一般把“Good”当作正类(即相关)，把“PotentiallyUseful”，和“Bad”当做负类(即不相关)。表 1 中的“原始问题”即为用户输入的问题，“相关问题”为可能与“原始问题”相关的候选问题。

本文中答案选择和问题检索任务采用平均精度均值(Mean Average Precision，MAP)，平均召回率(Average Recall，AvgRec)，平均排序倒数(Mean Reciprocal Rank ，MRR)作为评价指标。MAP 可以衡量模型整体的排序效果，具体定义见式(18)和式(19)。其中，$AP(q_i)$ 表示第 i 个检索的精确率；$|RD|$ 为检索对应的相关文本总数，$rank_{rd_i}$ 为第 i 个相关文本在排序结果中的次序。MRR 注重于排序结果中的第一个相关项，具体定义见式(20)，其中 $rank_i$ 为第 i 个检索结果中第一个相关文本的排名。AvgRec 用来衡量模型整体的“查全率”，即查出来的相关项占所有相关项的比例，如式(21)，其中，$Rec(q)$ 为检索 q 的召回率。

$$\mathrm{MAP}=\frac{1}{|Q|}\sum_{i1}^{|Q|}\mathrm{AP}(q_i) \quad (18)$$

$$\mathrm{AP}(q)=\sum_{i1}^{|RD|}rank_{rd_i} \quad (19)$$

$$\mathrm{MRR}=\frac{1}{|Q|}\sum_{i1}^{|Q|}\frac{1}{rank_i} \quad (20)$$

$$\mathrm{AvgRec}=\frac{1}{Q}\sum_{q1}^{Q}Rec(q) \quad (21)$$

2.2 调参与特征选择

为了达到比较精确的排序效果，XGBoost 排序

模型需要设置很多参数，因此调参和模型选择非常关键，XGBoost 中需要调整的参数有 eta、max_depth、min_child_weight、gamma、subsample、colsample_bytree 6 个参数，为了缩小参数范围，我们首先找出了 3 个敏感参数：gamma、subsample 和 colsample_tree，之后固定其它参数的值，主要对这 3 个参数进行调节。

表 2　XGBoost 主模型参数

Gamma	19
Subsample	0.5
Colsample_bytree	0.5
Max_depth	10
Eta	0.01
Scale_pos_weight	0.7

为了避免过拟合，本工作将每个参数的值设定为一定范围内的等差数列，采用 5 折交叉验证的方式分别对答案选择和问题检索两个任务的模型参数进行选择，找出交叉验证 MAP 平均分值最高的参数作为主模型的参数，XGBoost 主模型参数如表 2 所示。除了主模型之外，我们还选出交叉验证 MAP 分值方差最小的两组参数，训练出两个对照模型，用以同主模型进行对比分析。

对于特征提取时用到的神经网络模型，本文采用 Adagrad[13] 的优化方法进行训练。Adagrad 是一种基于随机梯度下降（stochastic gradient descent）的优化方法，其可以根据每个参数的数值大小对随机梯度下降中的学习率进行动态调整，加快模型的收敛速率。此外，为了防止模型会过拟合，本文在训练模型双向 LSTM 模型和 2D 神经网络匹配模型时采用 dropout[14] 机制和 early-stopping[15] 策略。两个模型的详细参数如表 3 所示。

表 3　深度匹配模型参数

	双向 LSTM	2D 神经网络
词向量维度	200	200
LSTM 状态变量维度	200	200
CNN 卷积核数量	—	8
CNN 卷积核大小	—	(3,3)
MLP 神经元数量	(200,50,2)	(400,50,2)
Dropout 速率	0.5	0.5

2.3　实验结果

本文对模型中用到的所有模型做了特征重要性分析，即将每类特征一一去掉，利用 5 折交叉验证得出相应的平均评价分值进行对比分析。详细结果如表 4 所示。

表 4　答案选择模型特征重要性

	5 折交叉验证			测试集		
特征	MAP	AvgRec	MRR	MAP	AvgRec	MRR
全部特征	70.65	88.54	76.17	88.24	93.87	92.34
-传统自然语言特征	69.06	87.94	75.16	87.83	93.60	92.73
-Tf-idf 余弦相似度	70.28	88.21	76.23	87.88	93.75	92.21
-最长公共子序列	69.95	88.01	76.15	88.04	93.90	92.21
-深度匹配特征	64.81	82.85	71.91	85.06	91.40	91.52
-词向量余弦相似度	69.90	88.28	76.24	88.31	93.81	92.40
-双向 LSTM	67.57	86.67	74.54	88.02	93.90	92.54
-2D 神经网络	69.72	88.01	75.86	88.17	94.04	92.50

从表中可以看出，在使用全部特征时，模型效果达到最高，其中 MAP 分值为 70.65。在去掉传统自然语言特征后，模型 MAP 分值降到 69.06；而在去掉深度匹配特征后，模型的 MAP 分值降到 64.81，大约降了 5 个百分点，显示出深度匹配特征在答案选择任务中的重要性。

3 结论

针对自动问答系统中的关键任务——答案选择，我们提出一种融合深度匹配特征的排序模型。其中采用词向量模型、双向 LSTM、2D 神经网络等深度学习技术对文本对的语义匹配特征进行提取，可以有效应对文本匹配问题中的三个挑战：词语语义的多元性、短语匹配的结构性和文本匹配的层次性。实验结果显示，融合深度匹配特征的答案选择模型比基于传统特征的模型 MAP 值高 5%左右。

参考文献

[1] 庞亮，等. 深度文本匹配综述[J]. 计算机学报，2017，(04)：985-1003.

[2] Chen T，Guestrin C. XGBoost：A Scalable Tree Boosting System[C]//Proceedings of the ACM SIGKDD International Conference. ACM，2016：785-794.

[3] Loper E，Bird S. NLTK：the Natural Language Toolkit[C]//Proceedings of the Acl-02 Workshop on Effective TOOLS and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Association for Computational Linguistics，2002：63-70.

[4] Klein D，Manning C D. Accurateunlexicalized parsing[C]//Proceedings of the Meeting on Association for Computational Linguistics. Association for Computational Linguistics，2003：423-430.

[5] Salton G，Buckley C. Term-weightingapproaches in automatic text retrieval[J]. Information Processing and Management，1988，24(5)：513-523.

[6] Allison L，Dix T I. A bit-string longest-common-subsequencealgorithm[J]. Information Processing Letters，1986，23(5)：305-310.

[7] Mikolov T，Sutskever I，Chen K，et al. Distributed representations of words and phrases and their compositionality[C]//Proceedings of the Advances in neural information processing systems，F，2013.

[8] Ikolov T，Yih W-T，Zweig G. Linguistic Regularities in Continuous Space Word Representations[C]//Proceedings of the HLT-NAACL，F，2013.

[9] Graves A. Long Short-Term Memory[M]. Supervised Sequence Labelling with Recurrent Neural Networks. Springer Berlin Heidelberg，2012：1735-1780.

[10] Pang L，Lan Y，Guo J，et al. Text Matching as Image Recognition[C]//AAAI. 2016：2793-2799.

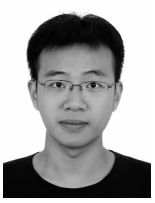
[11] Nair V，Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of International Conference on Machine Learning. Omnipress，2010：807-814.

[12] Nakov P，et al. SemEval-2017 Task 3：Community Question Answering[C]//Proceedings of International Workshop on Semantic Evaluation. 2017.

[13] Duchi J，Hazan E，Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research，2011，12(7)：2121-2159.

[14] Srivastava N，et al. Dropout：a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research，2014，15(1)：1929-1958.

[15] Caruana R，Lawrence S，Giles C L. Overfitting in neural nets：Backpropagation，conjugate gradient，and early stopping[C]//Proceedings of Advances in neural information processing systems. 2001：402-408.



冯文政(1996—)，本科生，主要研究领域为自然语言处理、问答。

E-mail：wenzhengfeng96@163.com

唐杰(1977—)，博士，副教授，主要研究领域为数据挖掘和机器学习。

E-mail：jery.tang@gmail.com