计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(7):158−164 [doi: 10.15888/j.cnki.csa.008544]

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

融合多角度特征的文本匹配模型①

李　广, 刘　新, 马中昊, 黄浩钰, 张远明

(湘潭大学 计算机学院·网络空间安全学院, 湘潭 411105)
通信作者: 刘　新, E-mail: liuxin@xtu.edu.cn

摘　要: 文本匹配是自然语言处理的一个核心研究领域, 深度文本匹配模型大致可以分为表示型和交互型两种类型, 表示型模型容易失去语义焦点难以衡量词上下文重要性, 交互型模型缺少句型、句间等全局性信息. 针对以上问题提出一种融合多角度特征的文本匹配模型, 该模型以孪生网络为基本架构, 利用 BERT 模型生成词向量进行词相似度融合加强语义特征, 利用 Bi-LSTM 对文本的句型结构特征进行编码, 即融合文本词性序列的句型结构信息, 使用 Transformer 编码器对文本句型结构特征和文本特征进行多层次交互, 最后拼接向量推理计算出两个文本之间的相似度. 在 Quora 部分数据集上的实验表明, 本模型相比于经典深度匹配模型有更好的表现.

关键词: 文本匹配; 句型结构; Transformer 框架; 孪生网络; Bi-LSTM; 特征融合; 注意力机制; 自然语言处理

引用格式: 李广,刘新,马中昊,黄浩钰,张远明.融合多角度特征的文本匹配模型.计算机系统应用,2022,31(7):158–164. http://www.c-s-a.org.cn/1003-3254/8544.html

Text Matching Model Incorporating Multi-angle Features

LI Guang, LIU Xin, MA Zhong-Hao, HUANG Hao-Yu, ZHANG Yuan-Ming

(School of Computer Science & School of Cyberspace Security, Xiangtan University, Xiangtan 411105, China)

Abstract: Text matching is a core research area in natural language processing. Deep text matching models can be broadly classified into representational models and interactive models. The former tends to lose semantic focus and fails to measure the contextual importance of words. The latter lacks global information such as sentence type and inter-sentence information. To address these problems, we propose a text matching model incorporating multi-angle features based on Siamese neural network. The model generates word vectors using the BERT model and enhances semantic features by the similarity fusion of words. It then encodes the syntactic structured features using Bi-LSTM, namely the syntactic structured information containing the lexical sequence. A Transformer encoder is utilized to realize the multi-level interaction between the syntactic structured features and the text features. Finally, the similarity is deduced by spliced vectors. Experiments on part of Quora question pair show that this model performs better than the classical deep matching model.

Key words: text matching; sentence structure; Transformer framework; Siamese neural network; Bi-LSTM; feature fusion; attention mechanism; natural language processing (NLP)

在自然语言处理 (NLP) 中, 文本匹配[1] 是研究对给定的两个文本, 采用匹配模型预测两个文本在某种意义上是否相似. 自动评分系统[2]、推荐系统[3]、问答系统[4]、信息检索[5] 等都可以抽象成一个文本匹配问题. 在主观题评分过程中, 系统可以判断用户的答案与标准答案相似性来进行评分, 极大的减少了教师的工作量. 对于推荐系统, 可以根据用户浏览的信息来推荐同领域或者同事件的相关信息. 问答系统中的答案匹配可以减少对人工客服的需求. 在信息检索中, 查询文档匹配结果的准确性和相关性都很重要. 所以对文本

① 基金项目: 智能化公共法律服务关键技术湖南省重点研发项目 (2022SK2106)
收稿时间: 2021-09-22; 修改时间: 2021-10-19; 采用时间: 2021-10-29; csa 在线出版时间: 2022-05-17

相似度匹配任务的研究是必要且是具有重要意义的.

传统的文本匹配基于 TF-IDF[6]、BM25[7]、VSM[8] 等的算法, 主要解决了词汇层面的匹配问题, 但还是存在如“同义词”“一词多义”“双关”等的局限性. 虽然浅层语义分析 LSA[9]、LDA[10] 等技术可以弥补传统方法的不足, 但是还是不能完全替代关键词匹配技术. 随着深度学习不断地发展, 对深度文本匹配模型的研究也层出不穷. 大致可以分为两类: 表示型和交互型. 表示型模型注重对文本的唯一表示, 经典的模型有 DSSM[11]、CDSSM[12]、MV-LSTM[13] 等, 但是此类模型容易失去语义焦点, 难以把握词的上下文的重要性. 交互型模型将词匹配信号作为后续的建模, 经典的模型有 ARC-II[14]、Match-SRNN[15]、DRMM[16] 等, 但此类模型忽略了句型、句间关系等全局性信息.

针对以上问题, 本文提出了一种融合多角度特征的文本匹配模型. 以孪生网络为基本架构, 对输入文本使用 BERT 模型进行词向量化表示, BERT 转化的词向量本身具有一定的语义信息, 使用 BERT 词向量计算出两个文本之间词向量的相似度再融合到两个文本中, 加强输入文本的语义. 对文本进行词性的标注后, 使用 Bi-LSTM 对两个文本对应的词性序列进行编码, 使用 Transformer 编码器对两个文本信息和文本的词性进行特征提取, 并使两个文本之间对应的信息进行多层次的信息交互. 对输出后的语义表示进行池化之后, 将两个文本对应信息进行对齐拼接送入多层感知机 (MLP) 中进行两个文本之间的语义匹配. 在 Quora 部分数据集上的实验表明, 本模型相比于经典深度匹配模型有更好的表现.

1 NLP 技术的主流框架

孪生网络[17] 包含两个或者更多相同子网络的神经网络架构, 子网络共享参数和权重, 孪生网络在探索两个样本之间的关系任务中起到很大的作用, 子网络结构的参数和权重共享, 使训练的参数极大的减少, 孪生网络结构可以提取文本整体的语义再送入匹配层进行匹配, 利于更好的探索两个文本之间的相似性和联系.

双向长短期记忆模型 (Bi-LSTM)[18] 由长短期记忆神经网络 (LSTM) 发展而来, Bi-LSTM 是由前向的 LSTM 和后向的 LSTM 组成. 单向的 LSTM 能捕捉较长距离的文本信息之间的依赖关系. 双向的 LSTM 能捕捉双向的文本信息的依赖关系, 从两个方向对输入序列进行特征提取.

Transformer 由 Google 在 2017 年发表的论文中提出[19], 该模型在很多其他语言理解任务上都超越了以往的模型. 与循环神经网络类模型相比, Transformer 不需要循环的处理, 结合位置信息可以并行地处理所有的单词和符号, 同时利用自注意机制将上下文的信息结合起来并行处理, 并且在处理过程中可以注意到文本中重要的信息, 训练速度相比于循环神经网络有很大的提升, 训练的效果也超越了以往的模型, 逐渐替代了循环神经网络模型.

BERT[20] 是一个预训练语言模型, 以 Transformer 为主要框架, 捕捉文本中的双向关系, 通过 mask language model (MLM) 和 next sentence prediction (NSP) 两个任务来预训练模型, 进一步增加了词向量模型的泛化能力, 对字符级、词级、句子级甚至句间关系特征都可以充分描述, 利用 BERT 的特征表示代替 Word2Vec[21] 的特征表示作为任务的词嵌入特征, 相较于词袋模型, BERT 的特征表示包含了更多的语义信息.

2 融合多角度特征的文本匹配模型 IMAF

基于孪生网络结构的 IMAF (text matching model incorporating multi-angle features) 模型由输入层、交互层、表示层、预测层组成, 在输入层利用 BERT 模型训练出来的特征作为匹配任务的词嵌入特征, 解决一词多义问题; 利用 BERT 的词向量特征计算两个文本的词相似度, 并将相似度结果融合到文本特征矩阵中, 增强局部特征; 对输入文本进行词性标注后, 利用 Bi-LSTM 对文本的词性信息进行词性嵌入编码; 在表示层利用 Transformer 编码器作为特征提取; 在交互层对两个文本融合词相似度信息和词性信息分别进行的注意力[22,23] 交互, 让模型对重点信息关注并充分学习; 在预测层, 将交互后的结果进行池化之后送入多层感知器最终通过 LogSoftmax 分类器得到两个文本的匹配结果. IMAF 结构如图 1, N 为 Transformer 编码器数量.

2.1 输入层

本文使用 BERT 模型将文本转化为词级别嵌入矩阵. 相比于 Word2Vec, BERT 生成的特征矩阵由单词周围的单词动态生成, 包含了上下文信息, 可以更好地解决一词多义的问题. 该模型拥有 12 个 Transformer 编码器, 隐藏层维度为 768 维, 每个编码器拥有 12 个注意力头.

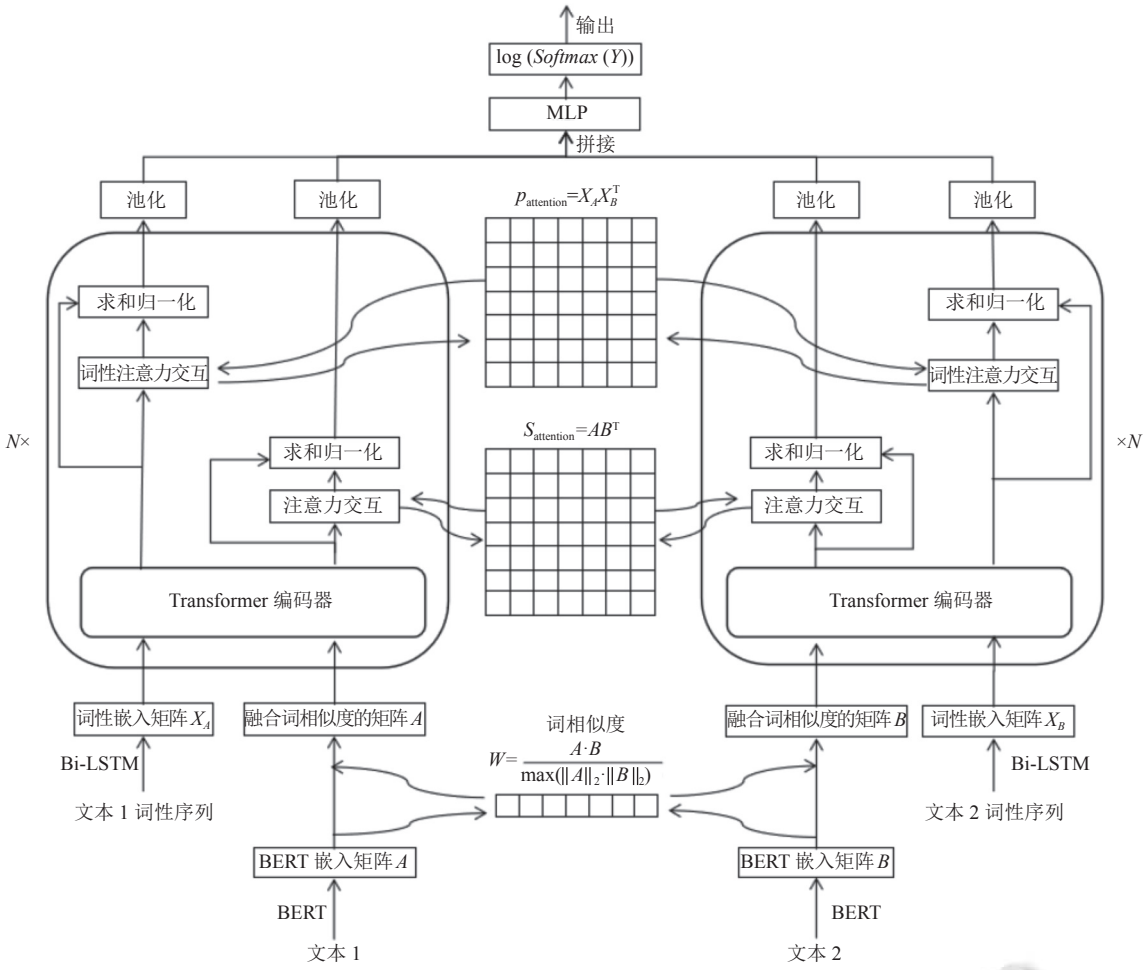

图 1 IMAF 结构图

BERT 生成的嵌入矩阵含有丰富的语义信息, 计算两个文本的词相似度作为匹配信号再分别融合到嵌入矩阵中, 增强文本的语义表示. BERT 对文本 1 的矩阵表示为 A, BERT 对文本 2 的矩阵表示为 B, 计算如下, 其中, $\|A\|_2$ 和 $\|B\|_2$ 代表矩阵 A 和 B 的二范数:

$$W=\frac{A\cdot B}{\max(\|A\|_2\cdot\|B\|_2)} \tag{1}$$

W 包含了 A 与 B 的词相似度信息, 再分别融入矩阵 A 和矩阵 B 中得到融合词相似度的矩阵, 融合计算过程如下:

$$A=AW \tag{2}$$

$$B=BW \tag{3}$$

对于词性, 将文本的词性序列进行向量表示, 送入 Bi-LSTM 模型学习文本语句结构的特征表示, 例如, 给定一个长度为 n 的文本序列 $[w_1, w_2,\cdots, w_n]$, 将单词在文本中的词性标注映射到向量空间, 对于单词 w_i 的词性, 都有一个唯一的索引表示, 通过将词性向量序列 $pos[w_1, w_2,\cdots,w_n]$ 输入到 Bi-LSTM 从两个方向, 即前向和后向, 学习语句结构特征表示. 公式如下:

$$X_A=BiLSTM(pos_A[w_1,w_2,\cdots,w_n]) \tag{4}$$

$$X_B=BiLSTM(pos_B[w_1,w_2,\cdots,w_n]) \tag{5}$$

2.2 表示层

表示层通过 Transformer 编码器对输入的信息进行特征提取, 编码器由 N 个相同的 layer 组成, 每个 layer 分别由多头注意力机制 (multi-head self-attention mechanism) 和全连接层 (fully connected feed-forward network) 两个子层组成, 每个子层都做了参差连接 (residual connection) 与归一化 (normalisation) 操作, Transformer 编码器的内部结构如图 2 所示.

使用 Transformer 进行特征提取, 增强输入信息的

矩阵表示, 步骤如下:

(1) 文本经过输入层的处理得到输入矩阵维度为 $S\times E$, 其中, S 为最大序列长度, E 为嵌入维度. 本文中 S 为 32, E 为 768. 假设一个文本经过输入层处理的输入矩阵为 $A_{S\times E}$. 和对应的语句结构特征表示 $(X_A)_{S\times E}$. 以矩阵 A 做计算为例, 对于另一文本的矩阵 B 和 $(X_B)_{S\times E}$ 做相同计算.

(2) 通过注意力机制计算矩阵 Q (query)、K (key)、V (value), 其中, W^Q、W^K、W^V 为权重矩阵.

$$Q = W^Q A \tag{6}$$

$$K = W^K A \tag{7}$$

$$V = W^V A \tag{8}$$

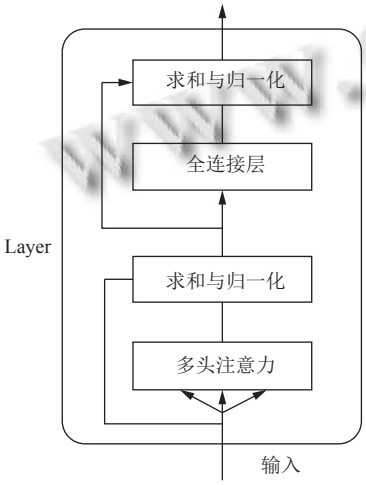

图 2 Transformer 编码器

(3) 得到矩阵 Q、K、V 之后进行 self-attention 计算. 其中 d_k 为 K 的维数.

$$Attention(Q, K, V) = Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{9}$$

(4) 通过多头注意力, 即 m 个不同线性变换对 Q、K、V 进行投影, 最后将所有的 attention 结果拼接得到 M, 传入一个线性层得到的多头注意力的输出 $M_{\text{attention}}$, 其中 m 为注意力的头数.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{10}$$

$$M = Concat(head_i, \cdots, head_m) \tag{11}$$

$$M_{\text{attention}} = Linear(M) \tag{12}$$

(5) 再对得到的结果进行残差连接和归一化之后作为全连接层的输入.

$$z_{\text{output}} = LayerNorm(A + M_{\text{attention}}) \tag{13}$$

(6) 最后送入全连接层之后再进行一次残差连接和层归一化, 得到最终结果, 输出矩阵的维度与 A 一致.

$$Y_{\text{output}} = LayerNorm(A + F(z_{\text{output}}) \tag{14}$$

2.3 交互层

经过 Transformer 特征提取后的文本 1 的矩阵表示为 $A_{S\times E}$、对应的词性嵌入矩阵为 $(X_A)_{S\times E}$, 文本 2 的矩阵表示为 $B_{S\times E}$, 对应的词性嵌入矩为 $(X_B)_{S\times E}$.

计算两个文本信息的交互注意力矩阵 $(S_{\text{attention}})_{S\times S}$:

$$S_{\text{attention}} = AB^{\mathrm{T}} \tag{15}$$

对 $(S_{\text{attention}})_{S\times S}$ 的每一行进行最大池化操作, 再融合到 A 得到交互后的 A, 此时 A 包含了 B 对 A 中所有的词注意力权重信息, 计算如下:

$$A = MaxPooling(S_{\text{attention}})_{\text{row}} \times A \tag{16}$$

对 $(S_{\text{attention}})_{S\times S}$ 的每一列进行最大池化操作, 再融合到 B 得到交互后的 B, 此时 B 包含了 A 对 B 中所有的词注意力权重信息, 计算如下:

$$B = (MaxPooling(S_{\text{attention}})_{\text{col}})^{\mathrm{T}} \times B \tag{17}$$

计算两个文本对应的词性嵌入矩阵交互注意力矩阵 $(P_{\text{attention}})_{S\times S}$:

$$P_{\text{attention}} = X_A X_B^{\mathrm{T}} \tag{18}$$

对 $(P_{\text{attention}})_{S\times S}$ 的每一行进行最大池化操作, 再融合到 X_A 得到交互后的 X_A, 此时 X_A 包含了 X_B 对 X_A 中所有的词性注意力权重信息, 计算如下:

$$X_A = MaxPooling(P_{\text{attention}})_{\text{row}} \times X_A \tag{19}$$

对 $(P_{\text{attention}})_{S\times S}$ 的每一行进行最大池化操作, 再融合到 X_B 得到交互后的 X_B, 此时 X_B 包含了 X_A 对 X_B 中所有的词性注意力权重信息, 计算如下:

$$X_B = (MaxPooling(P_{\text{attention}})_{\text{col}})^{\mathrm{T}} \times X_B \tag{20}$$

再将结果进行求和与归一化, 经过 N 次的交互后, 使得到的结果包含更多的交互信息和上下文信息, 其中 N 为 Transformer 编码器的数量.

2.4 预测层

假设经过交互后的两个文本矩阵表示为 $A_{32\times 768}$ 和 $B_{32\times 768}$, 预测方法来自文献 [24,25], 分别经过最大池化后得到对应向量为 a 和 b; 对应的交互后的词性矩阵表示为 $(X_A)_{32\times 768}$ 和 $(X_B)_{32\times 768}$, 分别经过最大池化后得到对应向量为 x_1 和 x_2; 进行向量拼接后送入多层感知机, 得到匹配结果, 计算如下:

$$Y = H(Concat(a,b,x_1,x_2,a\times b,|a-b|)) \quad (21)$$

其中, $a\times b$ 表示向量 a 与向量 b 按位相乘, 注重两个文本相同的地方; $|a-b|$代表向量 a 与向量 b 按位相减后的绝对值, 注重两个文本相异的地方, H 为多层的前馈神经网络, 将 6 个向量拼接后送入多层的前馈神经网络经过 LogSoftmax 分类器得到最终的预测结果, 计算如下:

$$y = \log(Softmax(Y)) \quad (22)$$

3 实验及分析

3.1 数据集

Quora Question Pair 是美国知识问答网站 Quora 发布的数据集, 包含了 40 万对的问句对, 旨在判断两句话是否同义. 为了验证模型在少数据量和短文本上情况下的有效性, 从中抽取了 2 万对短文本句子, 相同含义的句子标记为 1, 不同为 0, 并将其切分为训练集 (15 996 对)、验证集 (2 002 对) 和测试集 (2 002 对).

3.2 评估准则

实验采用的评估准则是 $F1$ 值和准确率 Acc, $F1$ 值由精确度和召回率得到, TP (true positive) 为真正例, FP (false positive) 为假正例, FN (false negative) 为假负例, TN (true negative) 为真负例, 计算如下:

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (23)$$

$$P = \frac{TP}{TP+FP} \quad (24)$$

$$R = \frac{TP}{TP+FN} \quad (25)$$

$$F1 = \frac{2\times P\times R}{P+R} \quad (26)$$

3.3 模型参数设置

模型的复杂程度通常与 Transformer 编码器的层数设置有着莫大的关系, 往往层数越多, 训练时间越长. 因此找到一个层数少, 训练速度快且准确率高的模型是迫切的. 本文将 Transformer 编码器层数分别设置为 1、2、3、4、5、6. $F1$ 值与 Transformer 编码器层数的实验结果如图 3 所示, Acc 值与 Transformer 编码器层数的实验结果如图 4 所示, 最终将编码器层数设置为 3.

表示层的性能与注意力头数有关. 但数量过多可能导致模型过拟合. 本文将注意力头的个数设置为 4、6、8、12. $F1$ 值与编码器注意力头数的实验结果如图 5 所示, Acc 值与编码器注意力头数的实验结果如图 6 所示, 最终将编码器注意力头数设置为 8.

训练模型时需要关注模型的收敛情况, 如果模型收敛了就应当停止训练, 否则模型将会过拟合, 达不到期望的效果. IMAF 模型收敛情况如图 7 所示. 训练次数在 20 左右模型就已经开始收敛, 因此将训练次数设置为 25.

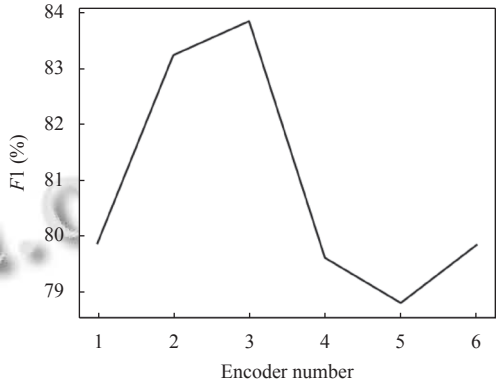

图 3 $F1$ 值随编码器层数变化图

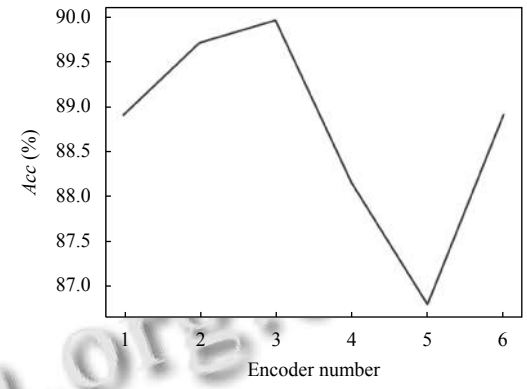

图 4 Acc 值随编码器层数变化图

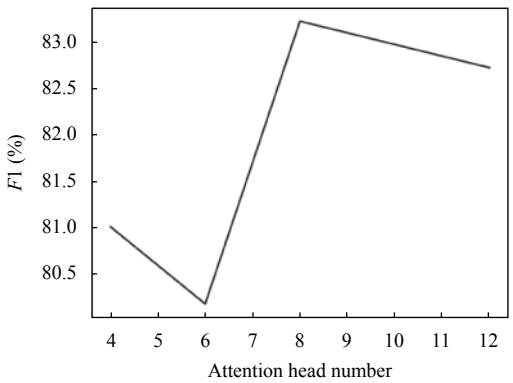

图 5 $F1$ 值随注意力头数变化图

3.4 实验对比

IMAF 模型实验部分主要参数如表 1 所示.

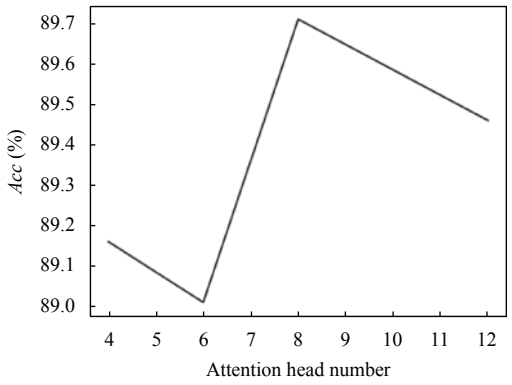

图 6 *Acc* 值随注意力头数变化图

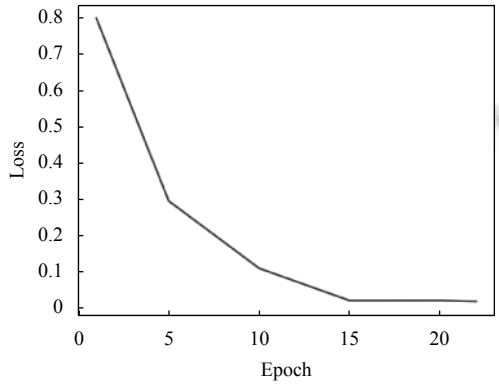

图 7 IMAF 模型收敛情况

表 1 模型参数设置表

参数	大小
Transformer编码器层数	3
Transformer注意力头数	8
Transformer隐藏层维度	768
最大序列长度	32
batch_size	512
优化器	Adam
训练次数	25
注意力的dropout	0.1

为了验证 IMAF 模型的效果, 本文选取多个经典的文本匹配模型进行对比实验.

针对 DSSM 和 CDSSM 无法捕捉较远距离的上下文距离的缺点, 文献 [26] 提出了 LSTM-DSSM 来解决该问题.

针对现有模型计算能力弱和特征提取能力弱的缺点, 文献 [27] 提出了 Transformer-DSSM 模型.

实验引入仅使用词相似度 $\text{IMAF}_{\text{word-similary}}$ 模型, 以及利用 LSTM 的变种代替 DSSM 的深度神经网络 BiLSTM-DSSM、BiGRU-DSSM 和 GRU-DSSM 做对比实验. 模型对比实验表如表 2 所示.

从实验结果可以看出, 本文提出的 IMAF 模型的 *F*1 值达到了 83.83%, 准确率和召回率都有着不俗的表现, 从前 5 组实验验证了 Transfomer 编码器提取特征的能力, 由第 5、6 组实验验证了引入词相似度的有效性; 由第 6、7 组实验可知, IMAF 模型的文本句型结构信息的引入确实提升了文本匹配的效果, 由第 1、5、7 组实验可知, IMAF 模型在文本匹配方面有着不错的效果, 主要体现在召回率、*F*1、准确率的提升. 其原因在于: 利用词相似度融合加强文本信息, 使之后的操作能更好的衡量词上下文重要性, 利用 Transformer 编码器作为优秀的特征提取器, 利用文本信息和句型结构信息的多次交互学习到更丰富的特征表现形式, 对文本匹配的效果有着不错的表现.

表 2 模型对比实验结果表

模型	*P*	*R*	*F*1	*Acc*
LSTM-DSSM	0.8594	0.7120	0.7788	0.8751
GRU-DSSM	0.8677	0.7217	0.7880	0.8801
BiLSTM-DSSM	0.9002	0.6570	0.7596	0.8716
BiGRU-DSSM	0.8835	0.7120	0.7885	0.8821
Transformer-DSSM	0.8591	0.7896	0.8229	0.8951
$\text{IMAF}_{\text{word-similary}}$	**0.9106**	0.7249	0.8072	0.8931
IMAF	0.8336	**0.8430**	**0.8383**	**0.8996**

4 结束语

针对现有文本匹配模型存在一些的问题, 提出了一种融合多角度特征的文本匹配模型 IMAF, 该模型以孪生网络为基础架构, 融合了词相似度, 对文本的信息和句型结构信息进行多层的交互, 使模型学习到更加丰富的特征表示, 从对比实验结果来看, 本文提出的 IMAF 模型在文本匹配上有着不错的效果.

参考文献

1 Guo JF, Fan YX, Ji X, *et al*. Matchzoo: A learning, practicing, and developing system for neural text matching. Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM, 2019. 1297–1300.

2 Walia TS, Josan GS, Singh A. An efficient automated answer scoring system for Punjabi language. Egyptian Informatics Journal, 2019, 20(2): 89–96. [doi: 10.1016/j.eij.2018.11.001]

3 Zhou K, Wang H, Zhao WX, *et al*. S3-Rec: Self-supervised learning for sequential recommendation with mutual information maximization. Proceedings of the 29th ACM International Conference on Information & Knowledge

Management. New York: ACM, 2020. 1893–1902.
4 Wu JM, Hao YB. Cross-sentence pre-trained model for interactive QA matching. Proceedings of the 12th Language Resources and Evaluation Conference. Marseille: European Language Resources Association, 2020. 5417–5424.
5 Wang X, Macdonald C, Tonellotto N, *et al*. Pseudo-relevance feedback for multiple representation dense retrieval. Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval. New York: ACM, 2021. 297–306.
6 黄承慧, 印鉴, 侯昉. 一种结合词项语义信息和 TF-IDF 方法的文本相似度量方法. 计算机学报, 2011, 34(5): 856–864.
7 Shan X, Liu CJ, Xia YQ, *et al*. GLOW: Global weighted self-attention network for Web search. arXiv: 2007.05186v3, 2020.
8 Qi JJ, Lei L, Zheng K, *et al*. Patent analytic citation-based VSM: Challenges and applications. IEEE Access, 2020, 8: 17464–17476. [doi: 10.1109/ACCESS.2020.2967817]
9 Suleman RM, Korkontzelos I. Extending latent semantic analysis to manage its syntactic blindness. Expert Systems with Applications, 2021, 165: 114130. [doi: 10.1016/j.eswa.2020.114130]
10 张小川, 余林峰, 张宜浩. 基于 LDA 的多特征融合的短文本相似度计算. 计算机科学, 2018, 45(9): 266–270.
11 Huang PS, He XD, Gao JF, *et al*. Learning deep structured semantic models for web search using clickthrough data. Proceedings of the 22nd ACM International Conference on Information & Knowledge Management. New York: ACM, 2013. 2333–2338.
12 Shen YL, He XD, Gao JF, *et al*. A latent semantic model with convolutional-pooling structure for information retrieval. Proceedings of the 23rd ACM International Conference on Information and Knowledge Management. New York: ACM, 2014. 101–110.
13 Wan SX, Lan YY, Guo JF, *et al*. A deep architecture for semantic matching with multiple positional sentence representations. Proceedings of the 13th AAAI Conference on Artificial Intelligence. Phoenix: AAAI, 2016. 2835–2841.
14 Hu BT, Lu DZ, Li H, *et al*. Convolutional neural network architectures for matching natural language sentences. Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014. Montreal: NIPS, 2014. 2042–2050.
15 Wan SX, Lan YY, Xu J, *et al*. Match-SRNN: Modeling the recursive matching structure with spatial RNN. Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York: AAAI Press, 2016. 2922–2928.
16 Guo JF, Fan YX, Ai QY, *et al*. A deep relevance matching model for ad-hoc retrieval. Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. New York: ACM, 2016. 55–64.
17 Vijjali R, Mishra A, Nagamalla S, *et al*. Semantic embeddings for food search using Siamese networks. Proceedings of the 4th International Conference on Natural Language Processing and Information Retrieval. New York: ACM, 2020. 138–143.
18 Zheng SH, Chen F, Wang XJ. Semantic matching for short texts: A cross attention mechanism. Journal of Physics: Conference Series, 2021, 1757(1): 012087. [doi: 10.1088/1742-6596/1757/1/012087]
19 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Red Hook: Curran Associates Inc., 2017. 6000–6010.
20 Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: Association for Computational Linguistics, 2019. 4171–4186.
21 Sharma AK, Chaurasia S, Srivastava DK. Sentimental short sentences classification by using CNN deep learning model with fine tuned Word2Vec. Procedia Computer Science, 2020, 167: 1139–1147. [doi: 10.1016/j.procs.2020.03.416]
22 Cheng Y, Yao LB, Xiang GX, *et al*. Text sentiment orientation analysis based on multi-channel CNN and bidirectional GRU with attention mechanism. IEEE Access, 2020, 8: 134964–134975. [doi: 10.1109/ACCESS.2020.3005823]
23 Rothe S, Narayan S, Severyn A. Leveraging pre-trained checkpoints for sequence generation tasks. Transactions of the Association for Computational Linguistics, 2020, 8: 264–280. [doi: 10.1162/tacl_a_00313]
24 Mou LL, Men R, Li G, *et al*. Natural language inference by tree-based convolution and heuristic matching. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: ACL, 2016. 130–136.
25 Yang RQ, Zhang JH, Gao X, *et al*. Simple and effective text matching with richer alignment features. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: Association for Computational Linguistics, 2020. 4699–4709.
26 Palangi H, Deng L, Shen YL, *et al*. Semantic modelling with long-short-term memory for information retrieval. arXiv: 1412.6629, 2014.
27 赵梦凡. 基于 Transformer 的文本语义相似度算法研究 [硕士学位论文]. 湘潭: 湘潭大学, 2020.

(校对责编: 孙君艳)