计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(10):295−302 [doi: 10.15888/j.cnki.csa.008756]

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

基于孪生网络和字词向量结合的文本相似度匹配①

李奕霖, 周艳平

(青岛科技大学 信息科学技术学院, 青岛 266061)
通信作者: 周艳平, E-mail: ypzhou@qust.edu.cn

摘　要: 文本相似度匹配是许多自然语言处理任务的基础, 本文提出一种基于孪生网络和字词向量结合的文本相似度匹配方法, 采用孪生网络的思想对文本整体建模, 实现两个文本的相似性判断. 首先, 在提取文本特征向量时, 使用 BERT 和 WoBERT 模型分别提取字和词级别的句向量, 将二者结合使句向量具有更丰富的文本语义信息; 其次, 针对特征信息融合过程中出现的维度过大问题, 加入 PCA 算法对高维向量进行降维, 去除冗余信息和噪声干扰; 最后, 通过 Softmax 分类器得到相似度匹配结果. 通过在 LCQMC 数据集上的实验表明, 本文模型的准确率和 $F1$ 值分别达到了 89.92% 和 88.52%, 可以更好地提取文本语义信息, 更适合文本相似度匹配任务.

关键词: 文本相似度匹配; 字词向量结合; 孪生网络; PCA 算法; BERT

引用格式: 李奕霖,周艳平.基于孪生网络和字词向量结合的文本相似度匹配.计算机系统应用,2022,31(10):295–302. http://www.c-s-a.org.cn/1003-3254/8756.html

Similar Text Matching Based on Siamese Network and Char-word Vector Combination

LI Yi-Lin, ZHOU Yan-Ping

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

Abstract: Text similarity matching is the basis of many natural language processing tasks. This study proposes a text similarity matching method based on a Siamese network and char-word vector combination. The method adopts the idea of the Siamese network to model the overall text so that the text similarity can be determined. First, when text feature vectors are extracted, BERT and WoBERT models are used to extract character-level and word-level sentence vectors which are then combined to have richer text semantic information. If the dimension is too large during feature information fusion, the principal component analysis (PCA) algorithm is employed for the dimension reduction of high-dimensional vectors to remove the interference of redundant information and noise. Finally, the similarity matching result is obtained through the Softmax classifier. The experimental results on the LCQMC dataset show that the accuracy and $F1$ score of the model in this study reach 89.92% and 88.52%, respectively, which can better extract text semantic information and is more suitable for text similarity matching tasks.

Key words: text similarity matching; char-word vector combination; Siamese network; principal component analysis (PCA) algorithm; BERT

计算文本语义相似度是在考虑自然语言表达的可变性和模糊性的同时, 确定句子在语义上是否等价, 它是自然语言处理领域的一个挑战性问题, 也是智能问答[1,2]、信息检索[3]、文档聚类[4]、机器翻译[5]、简答评分[6]等任务的重要组成部分.

传统的机器学习模型进行相似度计算时, 可以解决在词汇层面上文本之间的匹配, 但忽略了前后单词之间所具有的语义关联以及文本蕴含的语法信息. 例

① 收稿时间: 2022-01-21; 修改时间: 2022-02-22; 采用时间: 2022-03-11; csa 在线出版时间: 2022-06-24

如, 基于词袋模型[7] 的 TF-IDF, 把句子作为一个长向量, 以词为单位分开, 每一维代表一个词, 对应的权重代表这个词在文本中的重要程度. 但这种方法只能反应字面上的重要程度, 词之间各自独立, 无法反映序列信息和语义信息; Hofmann[8] 提出的 PLSA 模型引入了主题层, 采用期望最大化算法训练主题, 在训练到不同主题的情况下, 避免了同义词和多义词对相似度的影响, 在一定程度上考虑到了语义问题.

基于深度学习模型的文本相似度计算方法进一步关注到了文本语义层. CNN 和 RNN 通过对文本信息进行深层卷积, 使模型可以关注到文本的整体信息, 相比传统机器学习模型对文本的语义建模能力更强.

近年来, Transformer 模型[9] 因强大的语义建模能力被广泛应用于 NLP 领域, 其全局自注意力机制的运用使模型对文本特征的提取更加准确. Google 提出的 BERT (bidirectional encoder representations from Transformer) 模型[10] 在只保留 encoder 部分的前提下使用双向 Transformer, 这种模型对语境的理解比单向的语言模型更深刻. BERT 模型中每个隐藏层都对应着不同抽象层次的特征, 用来提取多维度特征, 独特的相对位置编码方法使得建模能力更强, 可以更准确地把握文本真实语义.

2020 年, 苏剑林[11] 在 BRET 模型的基础上开源了以词为单位的中文 WoBERT 模型, 基于词提取文本句向量, 相比字义能更好地对文本语义进行整体表达, 但是单纯的以词为单位存在一定的稀疏性, 会存在有未登录词出现的现象, 对于未登录词能否做到正确的语义理解具有不确定性. Reimers 等基于孪生网络 (Siamese network) 和 BERT 模型提出了 SBERT 模型[12], 沿用了孪生网络的结构, 将不同的英文文本输入到两个 BERT 模型中, 这两个 BERT 模型参数共享, 获取到每个句子的表征向量, 之后再做分类目标和回归目标. SBERT 在文本语义相似度匹配任务上明显优于 BERT 模型. 虽然 SBERT 提高了运算效率, 但在本质上还是基于表示的 BERT 方法, 即通过基于字的方法来提取句子表征向量, 而且句子的特征交互只在网络顶层进行, 将其运用到语义复杂度高的中文文本中仍会出现语义理解不充分的问题.

本文针对中文文本相似度匹配任务, 提出了一种基于孪生网络和字词向量结合的文本相似度匹配方法. 本文整体框架采用孪生网络模型, 对匹配的两段文本采用同样的编码器和预训练模型. 首先通过 BERT 和 WoBERT 模型分别获取字级和词级的句向量, 在字词向量表示层采用向量并联的方式得到融合特征向量, BERT+WoBERT 的句向量表征方法改变了仅基于 BERT 的表示方法, 通过联合 WoBERT 模型基于词的句向量表征方法, 让句子转换为具有充分语义信息的高维向量; 其次, 将得到的特征向量送入特征信息整合层, 得到复杂但富含充分语义信息的文本向量. 针对孪生网络整合过程出现的维度过高的问题, 使用 PCA 算法压缩数据空间, 将高维数据的特征映射到低维空间, 实现对特征向量的降维降噪. 通过这种计算方法使模型更有效的关注到文本的深层语义特征, 解决了中文数据集中出现的字词模糊性和差异性问题, 提高了文本相似度匹配的准确率.

1 基础模型

1.1 BERT 预训练模型

Google 提出的 BERT 是一个预训练的语言表征模型, 将文本中无法直接计算的字转变为可计算的向量形式, 这些向量能够更好地反映出字在句子中的含义.

BERT 模型使用两个无监督预训练任务.

(1) 遮蔽语言模型: 随机选择句子 15% 的词用于预测, 其中 80% 的词用 [MASK] 替换, 10% 的被随机换掉, 剩下的 10% 保持不变.

(2) 下一句预测: 判断两句话是否为前后句关系, 选择训练集里的句子 A 和 B 时, 句子 B 有 50% 几率是 A 的下一句, 50% 是随机选择的句子.

BERT 模型的编码层通过联合调节所有层中的双向 Transformer 来训练, 使模型能够充分提取输入文本的语义信息. 图 1 为 BERT 模型的结构图, Trm 为 Transformer 编码器, E_1, E_2, ⋯, E_n 为模型的输入向量, T_1, T_2, ⋯, T_n 为输出向量, 经过计算得到句子 *seq_A* 的特征向量表示 $f(seq\_A)$.

BERT 模型只使用了 Transformer 架构中的 encoder 模块, 弃用了 decoder 模块. 其中, encoder 模块的多头自注意力机制可以从多个维度准确提取文本语义特征, 其主要运算过程如下: 首先进行自注意力的计算, 将输入向量 E_1, E_2, ⋯, E_n 与给定的权重矩阵 W^Q、W^K、W^V 相乘得到向量 Q、K、V. Q 表示与这个单词相匹配单词的属性, K 表示这个单词本身的属性, V 表示这个单词所包含的信息本身.

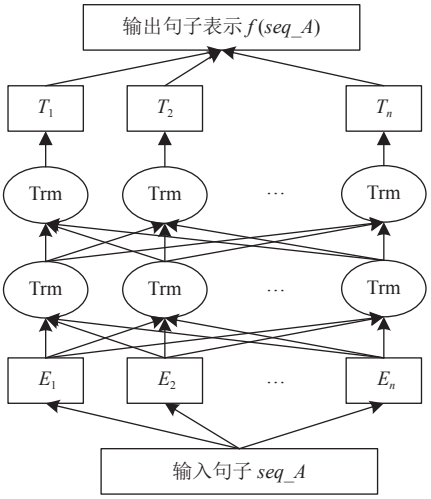

图 1 BERT 模型结构图

通过 attention 计算得到自注意力值:

$$Attention(Q,K,V)=Softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{1}$$

其中, d_k 为向量 K 的维度. 将 Q、K、V 通过线性映射的方式分为 n 份, 对每一份分别进行自注意力的计算, 最后通过并联的方式将 n 个自注意力模块结合起来, 然后通过左乘权重矩阵的线性映射方法得到最终输出, 完成整个多头注意力模块的计算, 计算如下:

$$MultiHead(Q,K,V)=Concat(head_0,head_1,\cdots,head_n)W^O \tag{2}$$

其中,

$$head_i=Attention(QW_i^Q,KW_i^K,VW_i^V) \tag{3}$$

1.2 WoBERT 预训练模型

WoBERT 是以词为单位的中文预训练模型, 让序列变短, 处理速度变快, 语义更明确.

WoBERT 模型相对 BERT 模型做出了如下改进:

(1) 加入前分词操作, 进行中文分词.

(2) 使用动态的 Mask 操作, 将训练数据重复 10 次, 使得每轮训练的 Mask 的位置不同.

(3) 学习任务只有遮蔽语言模型, 取消了下一句预测任务.

(4) batch size 从 256 扩大为 8k.

(5) 删除了 BERT 模型自带词汇表的中文冗余部分, 比如带##的中文字词, 将结巴分词自带的词汇表中词频最高的两万个加入词汇表, 减少了未登录词的出现概率, 最终词汇表规模为 33 586.

1.3 孪生网络

孪生网络定义两个网络结构分别表征对应的输入内容, 分为孪生网络和伪孪生网络. 孪生网络中的两个网络结构相同且共享参数, 当两个句子来自同一领域且在结构上有很大的相似度时可选择孪生网络; 伪孪生网络可以是不同结构的网络或不共享参数的同结构网络, 计算两个不同领域的句子相似度时可以选择伪孪生网络. 本文研究两个文本的相似度, 采用两个网络结构相同且共享参数的孪生网络模型. 其模型基础结构如图 2 所示, 孪生网络结构简单, 训练稳定, 以两个样本 input1 和 input2 为输入, 其两个子网络各自接收一个输入, 子网共享权重使得训练需要更少的参数, 这意味着需要更少的数据并且不容易过拟合.

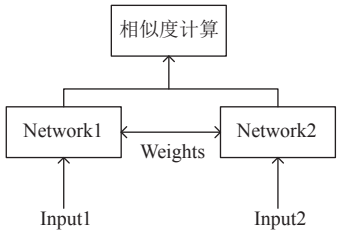

图 2 孪生网络基础结构

2 模型和方法

本文提出的基于孪生网络和字词向量结合的文本相似度匹配模型结构如图 3 所示, 主要分为 4 层: 输入层、字词向量表示层、特征信息整合层、输出层.

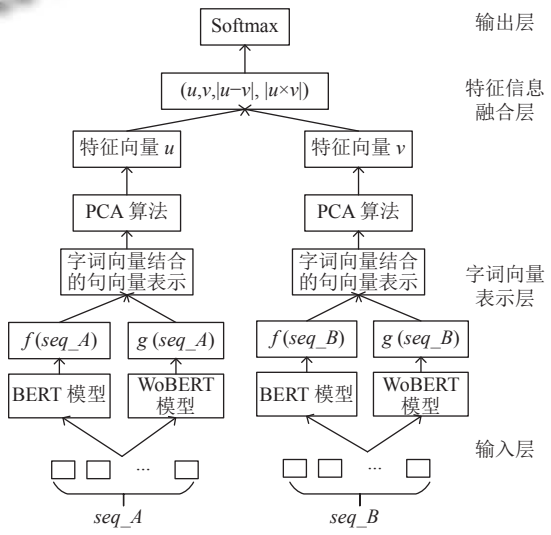

图 3 本文模型结构图

2.1 输入层

BERT 模型的输入是将字向量 (tokening embeddings)、文本向量 (segment embeddings) 和位置向量 (position embeddings) 拼接得到 E_{ci} 作为模型输入. 如图 4 所示.

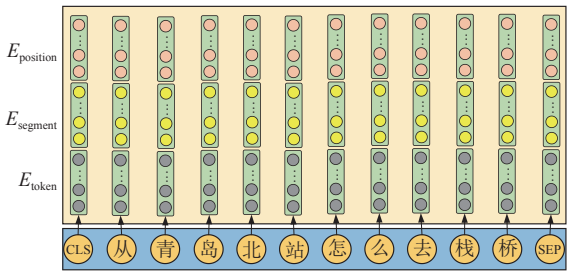

图 4 BERT 模型输入层

由于以字为单位的建模方法在处理中文数据集时存在语义确定性不高的问题, 模型往往难以对文本中重复出现的字准确提取真实语义特征. 本文引入 WoBERT 模型, 输入向量用 E_{wj} 表示, 此模型与 BERT 模型建模形式不同的地方在于 tokenize 为了分出中文单词在 BERT 模型的 tokenize 中加入了一个前分词操作. WoBERT 模型的 tokenize 方法流程如图 5 所示.

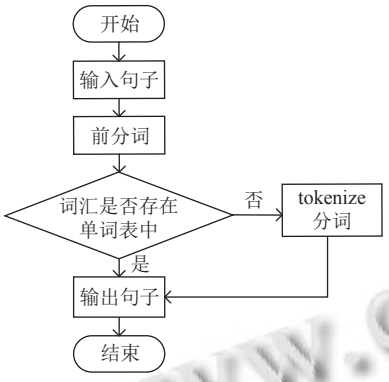

图 5 WoBERT 模型的 tokenize 流程图

对“从青岛北站怎么去栈桥”这句话, token embeddings 通过词汇表使用 WordPiece 嵌入, 用 E_{token} 表示; 字的位置向量用 E_{position} 表示; 由于模型中只有一个输入句子, 所以每一个字所处的某个句子信息是一样的 E_{segment}. 输入向量的计算公式如下:

$$E_{ci} = Concat(E_{\text{token}}, E_{\text{segment}}, E_{\text{position}}) \tag{4}$$

经过 WoBERT 的 tokenize 方法处理之后为 [‘[CLS]’, ‘从’, ‘青岛’, ‘北站’, ‘怎么’, ‘去’, ‘栈桥’, ‘[SEP]’].

使用孪生网络训练时的输入形式为:

[CLS]*seq_A*[SEP]\n[CLS]*seq_B*[SEP]

由于模型运行过程中, 内存占用率与输入模型句子长度 l 成平方增长, 但 batch 增加只略微影响到训练时间, 采用孪生网络的训练方式可以缩短模型的训练时间.

2.2 字词向量表示层

采用 BERT 和 WoBERT 模型分别获取句子的字向量和词向量表示, 最终得到一个句子的两种表达方式.

具体步骤如下:

(1) 通过 BERT 模型在 LCQMC 数据集上训练得到对应文本 x 的句向量表达. 每个句子得到一个二维矩阵 $char_i$, 行数为文本中字的个数 C_i, 列数为 768 维.

(2) 通过 WoEBRT 模型在 LCQMC 数据集上训练得到对应文本 x 的句向量表达. 每个句子得到一个二维矩阵 $word_j$, 行数为文本中词的个数 W_j, 列数为 768 维.

(3) 对得到的 C_i×768 和 W_j×768 维度的文本向量分别进行归一化, 对所有特征向量按行取平均作为最终向量 $f(x)$、$g(x)$, 维度均为 1×768.

(4) 对得到的基于字词级别的文本向量 $f(x)$、$g(x)$ 进行并联操作, 得到基于字词向量结合的文本向量 $s(x)$:

$$s(x) = Concat(f(x), g(x)) \tag{5}$$

2.3 特征信息整合层

降低向量维数会损失原始数据中具有可变性的一些特征向量, 但也会带来一些积极作用, 例如减少计算时间、避免过拟合、去除噪声等. PCA 算法 (principal component analysis) 是流行的线性降维算法之一, 它将一组相关变量 (P) 转换为较小的 K (K<P) 个特征子空间, 同时尽可能多地保留原始文本的主要特征.

Su 等人[13] 提出, 在处理相似度匹配任务时, 对 BERT 模型进行降维操作可以有效去除数据噪声, 提高模型准确率的同时降低计算复杂度. BERT 模型输出维度为 768, 在特征信息整合层, 并联组合会使输出的向量维度达到上千维, 冗余信息多且占用内存大. 本文用 PCA 算法对字词向量结合后的输出向量进行降维处理:

(1) 对输入的特征向量进行归一化.

(2) 计算输入样本特征向量的协方差矩阵.

(3) 计算协方差矩阵的特征值和特征向量.

(4) 选取协方差矩阵前 K 列作为降维矩阵.

(5) 降维矩阵映射到低维空间完成降维计算.

本文将融合后的向量 $S(seq\_A)$、$S(seq\_B)$ 维度降为 384 维, 得到两个输入句子的特征向量 u 和 v. 去除文本噪声的同时降低模型的整体建模难度, 提高了模型的灵活性和准确率.

本文探索了不同的特征整合方式对实验结果的影响, 采用通过字词向量结合并进行向量降维后得到向量 u、v、两个向量差的绝对值$|u-v|$和两个向量乘积的绝对值$|u\times v|$做并联的整合策略作为最终实验方案.

$$T = Concat(u, v, |u-v|, |u\times v|) \tag{6}$$

2.4 输出层

Softmax 函数在进行二分类任务时使用二项分布的计算方法, 相对于 Sigmoid 函数的单一建模方法, 它可以对两个类别进行建模, 得到两个相加为 1 的概率预测结果.

本文通过 Softmax 函数对输出的文本向量进行训练, 损失函数采用交叉熵损失. 最终输出结果为 0 和 1, 0 表示进行匹配的两段文本不相似, 1 表示相似.

3 实验及分析

3.1 数据集

本文使用的数据集 LCQMC 是一个大规模的中文问答数据集, 侧重于语义匹配而不是简单的复述, 要求模型能够深度挖掘文本的高层语义信息. 语料库由两个问题和一个标签组成, 标签是 0 和 1 两种形式, 0 表示不相似, 1 表示相似. 数据集共有 260 068 对句子对, 其中训练集 238 766, 验证集 8 802, 测试集 12 500. 部分数据集样例如表 1 所示.

表 1 部分数据集样例

Seq_A	Seq_B	标签
近期上映的电影	近期上映的电影有哪些	1
叔叔是什么人	我是叔叔的什么人	0
石榴是什么时候成熟的?	成熟的石榴像什么?	0
汇理财怎么样	怎么样去理财?	0
我国的基本国情是什么?	我国的基本国情有哪些?	1

3.2 评价指标

为验证本文方法的效果, 采用准确率、召回率、精确率、$F1$ 值的评价指标来验证算法的有效性.

(1) 准确率 (*accuracy*), 表示预测结果预测正确的比率.

$$accuracy = \frac{\text{所预测正确的样本数}}{\text{总样本数}} \tag{7}$$

(2) 召回率 (*recall*), 衡量检索文本相似度的查全率.

$$recall = \frac{\text{正确识别相似性的总个数}}{\text{实际存在的相似句子数}} \tag{8}$$

(3) 精确率 (*precision*), 衡量检索文本相似度的查准率.

$$precision = \frac{\text{正确识别相似性的总个数}}{\text{所有检测到的相似句子数}} \tag{9}$$

(4) $F1$ 值, 对精确率和召回率的整体评价. $F1$ 值越大, 说明精确率和召回率更均衡.

$$F1 = 2\times\frac{\text{精确率}\times\text{召回率}}{\text{精确率}+\text{召回率}} \tag{10}$$

3.3 实验环境配置和参数说明

本文实验环境如表 2.

表 2 实验环境配置信息

软硬件	配置
开发语言和工具	Python 3.6+sublime
开发框架	TensorFlow 1.14.0
开发环境	Annaconda3, h5py==2.10 bert4keras=0.10.8, Keras=2.3.1
操作系统	Linux 4.15.0
GPU	NVIDIA GTX 1080Ti × 4
内存	16 GB DDR4-2400 MHz×16

选用 BERT 预训练模型为 BERT-Base-Chinese, 最大序列长度为 128, 训练批次为 8, 学习率为 2E–5, 共训练 5 轮. 选用 WoBERT 预训练模型为苏剑林等[11]以 RoBERTa-wwm-ext 模型为基础训练得到的 WoBERT 模型, 最大序列长度为 128, 训练批次为 16, 学习率为 5E–6.

训练 Softmax 分类器时, 选用交叉熵损失函数, 训练批次为 100, 训练轮数为 1 000, 学习率为 0.01.

3.4 实验结果及分析

在 LCQMC 数据集上进行了如下 4 组对比实验:

(1) 将字词向量结合生成句向量的文本相似度计算方法, 与单一字向量和单一词向量生成句向量的方法进行了性能比较.

(2) 使用 PCA 算法对特征向量降至不同维度对模型性能的影响.

(3) 不同的特征向量融合方式对模型性能的影响.

(4) 将本文模型与已发表的方法进行性能比较.

在文本句向量表达模块中, 使用如下 3 种方法提取文本句向量:

(1) 使用 BERT 得到基于字级别的句向量表示.

(2) 使用 WoBERT 得到基于词级别的句向量表示.

(3) 使用 BERT+WoBERT 得到基于字词向量结合的句向量表示.

在分别得到两个句向量表示之后, 通过并联操作进行特征向量融合然后输入 Softmax 分类器进行实验验证, 实验结果如表 3 所示.

表 3 字词向量结合方法 (%)

模型	*Accuracy*	*Precision*	*Recall*	*F*1
BERT (Baseline)	88.01	84.12	92.56	88.14
WoBERT	88.47	84.18	**92.64**	88.21
BERT+WoBERT	**88.95**	**85.02**	92.43	**88.57**

由表 3 可见, 尽管字词向量结合的文本句向量提取方法在召回率上稍低, 但在准确率、精确率和 *F*1 值上较 BERT 和 WoBERT 模型都有所提升, 证明了字词向量结合方法的语义表征能力.

为了对 BERT 及 WoBERT 模型进行评估, 绘制了两种模型在数据集上训练过程中的 loss 变化以及验证集的准确率曲线, 如图 6–图 8. 从图中可见, BERT 模型相对 WoBERT收敛更快, 虽然两个模型单次训练时输入网络的数据量不同, 但 WoBERT 模型最终 loss 值更低, 且在验证集上的最高准确率值优于 BERT 模型.

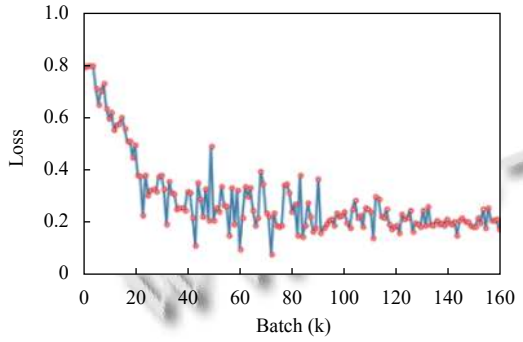

图 6 LCQMC 数据集上 BERT 模型训练的 loss 曲线

孪生网络在进行相似度计算时, 将两段文本分为两个 batch 分别提取句向量, 需要对两段特征向量进行融合. 本文对不同特征融合方法进行了对比实验, 经过 BERT+WoBERT 字词向量结合方法得到的两个输入文本的特征向量 u、v, 采用如下多种方式进行实验:

(1) 向量相加:

$$T = u + v \tag{11}$$

(2) 向量相乘:

$$T = |u \times v| \tag{12}$$

(3) 向量并联:

$$T = Concat(u, v) \tag{13}$$

(4) 向量并联组合 1:

$$T = Concat(u, v, |u - v|) \tag{14}$$

(5) 向量并联组合 2:

$$T = Concat(u, v, |u - v, |u \times v|) \tag{15}$$

本文针对不同向量融合方式进行了 5 组实验, 由表 4 可见, u+v 的融合方式较 (u, v) 方法准确率提高了 0.28%, 而 u×v 融合方式的实验效果不佳. 当采用 (u, v, |u–v|, |u×v|) 的融合方式时, 准确率和 *F*1 值分别达到了 88.86% 和 88.42%, 有着不错的匹配效果.

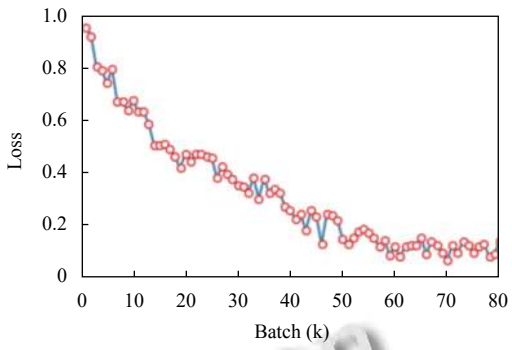

图 7 LCQMC 数据集上 WoBERT 模型训练的 loss 曲线

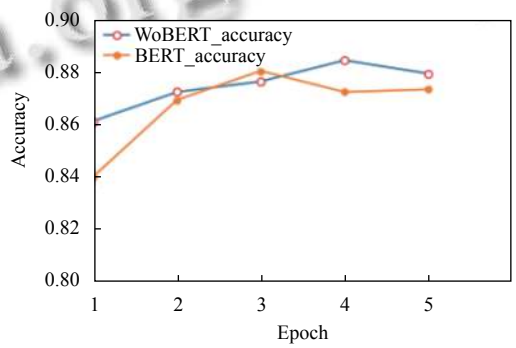

图 8 LCQMC 数据集上验证的 accuracy 值变化图

表 4 不同向量融合方式实验结果 (%)

融合方式	*Accuracy*	*Precision*	*Recall*	*F*1
(u, v) (baseline)	88.01	84.12	92.56	88.14
u×v	87.75	83.31	91.97	87.43
u+v	88.29	84.38	92.63	88.31
(u, v, |u–v|)	88.44	84.20	**92.77**	88.28
(u, v, |u–v|, |u×v|)	**88.86**	**84.53**	92.69	**88.42**

为证明采用 PCA 算法进行适当降维操作可以有效去除数据噪声, 提高模型的识别准确率且加快训练速度, 本文以 BERT 模型作为 baseline 在 LCQMC 数据集上进行了对比实验, 指定 PCA 的 n_components 参数也就是主成分分别为整数 384、256 和 100, 求得对应的贡献率, 实验如下:

由表 5 可见, 将 BERT 输出的 768 维向量降至 384 维使得输入 Softmax 的向量维度由 1 536 降至原来一半, 模型识别准确率达到最高. 再降至 256 维会损失一些特征值, 导致准确率降低. 说明在输入分类器的向量维度较高时, 使用 PCA 算法降维降噪的有效性.

表 5 不同向量维度的对比实验 (%)

实验方案	*Accuracy*	贡献率
维度768不降维 (baseline)	88.01	—
PCA降维至384	**88.24**	91.4
PCA降维至256	88.13	78.1
PCA降维至100	86.78	54.6

将本文方法在 LCQMC 数据集上与已发表的方法在准确率、精确率、召回率和 *F*1 值上做了对比实验, 实验结果如表 6 所示.

表 6 不同方法的测试结果 (%)

模型	*Accuracy*	*Precision*	*Recall*	*F*1
BiLSTM-char	76.14	70.59	89.45	78.91
BiLSTM-word	73.57	67.37	91.37	77.56
BiMPM-char	83.78	77.66	93.29	84.76
BiMPM-word	84.32	77.51	93.53	84.77
MSEM	85.84	78.98	93.71	85.72
Siamese- LSTM	84.73	84.18	83.42	83.80
BERT	88.01	84.12	92.56	88.14
WoBERT	88.47	84.18	92.64	88.21
SBERT	87.28	—	—	—
Ours	**89.92**	**84.69**	**92.72**	**88.52**

本文的对比模型如下.

(1) BiLSTM-char: 以字向量作为输入的双向 LSTM 文本相似度匹配模型.

(2) BiLSTM-word: 以词向量作为输入的双向 LSTM 文本相似度匹配模型.

(3) BiMPM-char: 以字向量作为输入, 基于 BiLSTM 的双边多角度文本相似度匹配模型.

(4) BiMPM-word: 以词向量作为输入, 基于 BiLSTM 的双边多角度文本相似度匹配模型.

(5) MSEM: 结合文本编码模型和近似最近邻搜索技术的通用语义检索框架.

(6) Siamese- LSTM[14]: 基于孪生网络和双层双向 LSTM 的文本相似度匹配模型.

(7) BERT: 以字为单位的预训练模型, 可以完成适用于文本匹配的下游任务.

(8) SBERT[15]: 基于孪生网络和 BERT 的文本相似度匹配模型.

(9) WoBERT: 以词为单位的中文预训练模型, 可以完成适用于文本匹配的下游任务.

从表 6 的结果可见, BiLSTM 和 BiMPM 只使用字或词单粒度下的特征提取方法, 不足以充分捕获中文文本的特征信息. MSEM 考虑词和字嵌入到一起作为文本表示, 准确率相对之前方法有所提高, 但没有捕捉不同粒度之间的相关特征, 表达能力仍然有限. BERT 模型凭借强大的建模能力相对之前方法取得了较大提升. SBERT 模型使用了最大池化和全连接层的 Siamese-BERT 模型, 在 LCQMC 数据集上的准确率与 BERT 模型相当, 验证了基于 BERT 的孪生网络模型的有效性. WoBERT 模型较 BERT 模型在准确度上有所提高, 说明以中文文本为基础的基于词粒度的预训练语言模型能更充分的理解中文语义. 本文在孪生网络的基础上, 基于字粒度和词粒度融合特征对文本进行建模, 解决了只使用 BERT 模型或 WoBERT 模型提取句子特征向量表达单一的问题, 验证了多角度获取文本特征信息方法的有效性, 进一步提高模型性能, 在 LCQMC 数据集上通过与其他模型的对比实验证明了本文模型在文本相似度匹配任务上的有效性.

4 结论与展望

本文提出了一种基于孪生网络和字词向量结合的文本相似度匹配方法, 采用字词向量结合的 BERT-WoBERT 模型解决了传统模型难以关注到中文文本语义语法信息的问题, 通过孪生网络和 PCA 算法探索多种融合方式以及降维降噪对相似度匹配结果的影响, 然后通过 Softmax 分类器进行二分类, 最终在 LCQMC 数据集上取得了不错的相似度匹配结果.

然而本文模型存在参数量过大, 计算时间复杂度过高的缺点, 下一步尝试将预训练模型进行知识蒸馏, 在不降低准确率的前提下加快模型速度, 解决资源占用率较大的问题.

参考文献

1 董自涛, 包佃清, 马小虎. 智能问答系统中问句相似度计算

方法. 武汉理工大学学报·信息与管理工程版, 2010, 32(1): 31–34.
2 Singh V, Dwivedi SK. Personalized approach for automated question answering in restricted domain. International Journal of Information Technology, 2020, 12(1): 223–229. [doi: 10.1007/s41870-018-0200-6]
3 王灿辉, 张敏, 马少平. 自然语言处理在信息检索中的应用综述. 中文信息学报, 2007, 21(2): 35–45. [doi: 10.3969/j.issn.1003-0077.2007.02.006]
4 贾晓婷, 王名扬, 曹宇. 结合 Doc2Vec 与改进聚类算法的中文单文档自动摘要方法研究. 数据分析与知识发现, 2018, 2(2): 86–95.
5 Wang Q, Li B, Xiao T, *et al*. Learning deep transformer models for machine translation. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019. 1810–1822.
6 程传鹏, 齐晖. 文本相似度计算在主观题评分中的应用. 计算机工程, 2012, 38(5): 288–290. [doi: 10.3969/j.issn.1000-3428.2012.05.089]
7 Harris ZS. Papers in structural and transformational linguistics. Dordrecht: Springer, 1970: 466–473.
8 Hofmann T. Probabilistic latent semantic analysis. Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence. Stockholm: Morgan Kaufmann Publishers Inc., 1999. 289–296.
9 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc. , 2017. 6000–6010.
10 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.
11 苏剑林. 提速不掉点: 基于词颗粒度的中文 WoBERT. https://kexue.fm/archives/7758. (2020-09-18).
12 Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: Association for Computational Linguistics, 2019. 3982–3992.
13 Su JL, Cao JR, Liu WJ, *et al*. Whitening sentence representations for better semantics and faster retrieval. arXiv: 2103.15316, 2021.
14 Palangi H, Deng L, Shen Y, *et al*. Semantic modelling with long-short-term memory for information retrieval. arXiv: 1412.6629, 2014.
15 彭浩然. 面向检索式问答的问句语义匹配方法研究 [硕士学位论文]. 哈尔滨: 哈尔滨工业大学, 2020.

(校对责编: 孙君艳)