计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(2):291−297 [doi: 10.15888/j.cnki.csa.008313]

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

长文本匹配 LTM-B 模型①

刘　龙, 刘　新, 蔡林杰, 唐　朝

(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: 长文本匹配是自然语言处理的一项基础工作, 在文本聚类、新闻推荐等方面有着关键作用. 受语料、篇幅结构、文本表示技术的限制, 长文本匹配工作进展缓慢. 近年提出的 BERT 模型在文本表示方面具有非常卓越的表现, 而对于 BERT 来说, 长文本的处理有截断法、分段法和压缩法 3 种常用方式, 截断法丢失大量文本信息, 分段法保留文本信息却丢失部分语义信息, 压缩法可能丢失部分关键信息. 针对以上问题, 本文对分段法加以改进, 提出一种基于 BERT 的长文本匹配模型 (long text matching model based on BERT, LTM-B), 它以孪生网络为基础, 采用分层的思想将文档切分成多个分段, 使用 BERT 模型处理文本向量化, 从而得到文档的矩阵表示, 并采用 BiLSTM 产生位置矩阵, 然后将文档矩阵及其位置矩阵求和输入至 Transformer 编码器进行特征提取, 最后将两个文档矩阵进行交互、池化、拼接后经由全连接层分类输出匹配结果. 实验表明, 相比于其他方法, LTM-B 模型在长文本匹配问题上拥有更好的表现.

关键词: 长文本匹配; BERT; 孪生网络; BiLSTM; Transformer

引用格式: 刘龙,刘新,蔡林杰,唐朝.长文本匹配 LTM-B 模型.计算机系统应用,2022,31(2):291–297. http://www.c-s-a.org.cn/1003-3254/8313.html

LTM-B Model of Long Text Matching

LIU Long, LIU Xin, CAI Lin-Jie, TANG Chao

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

Abstract: Long text matching is a basic work of natural language processing, and it plays a key role in text clustering, news recommendation, etc. Due to the limitations of the corpus, space structure, and text representation technology, long text matching has been progressing slowly. The bidirectional encoder representations from Transformer (BERT) model proposed in recent years has an excellent performance in the text representation. For BERT, there are three common methods for processing long texts: truncation, segmentation, and compression. The truncation method causes the loss of massive text information; the segmentation method retains text information but loses part of the semantic information; the compression method may lose part of the key information. In response to the above problems, this study improves the segmentation method and proposes a long text matching model based on BERT (LTM-B), which is based on the Siamese neural network and adopts a layered idea to divide the document into multiple segments. The BERT model is used for text vectorization. As a result, the matrix representation of the document is obtained. The bidirectional long short-term memory (BiLSTM) is employed to generate the position matrix, and then the sum of the document matrix and the position matrix is input to the Transformer encoder for feature extraction. Finally, the two matrices are interacted, pooled, and spliced, and then the matching results are output through the fully connected layer classification. Experiments show that the LTM-B model outperforms other methods in long text matching.

Key words: long text matching; BERT; siamese neural network; BiLSTM; Transformer

① 基金项目: 湖南省重点研发项目 (2022SK2106)
收稿时间: 2021-04-25; 修改时间: 2021-05-19; 采用时间: 2021-05-28; csa 在线出版时间: 2022-01-17

文本匹配[1]是自然语言处理 (NLP)[2]中一项基础任务, 旨在研究两个文本之间的语义匹配关系. 长文本匹配是文本匹配的一个重要子方向, 主要应用于文本聚类[3]、新闻推荐[4]、搜索引擎[5]、文本去重[6]、机器翻译[7]等领域. 在文本聚类方面, 两篇文档的相似度判断是必不可少的工作. 在新闻个性化推荐中, 系统可以根据用户近期阅读的新闻类型来向用户推送相关系列的新闻. 对于搜索引擎来说, 精准地查找到与用户搜索内容相关的文档是极其重要的. 文本去重可以抽象为文本与文本的相似度匹配问题, 而机器翻译可以理解为两种语言之间的匹配. 可想而知, 对长文本匹配任务的研究是一项具有重要意义的工作.

过去关于长文本匹配的工作比较少, 其原因在于: 第一, 长文本匹配相关语料相对来说较为匮乏, 缺少权威的数据集; 第二, 文档的篇幅很大, 篇章结构较为复杂, 语义信息的提取存在一定难度; 第三, 采用的文本表示方法处于较浅层次, 难以满足文档语义复杂性的要求.

BERT 预训练模型[8]是 NLP 领域近年来最具突破性的一项技术, 它可以很好地融合文本的多层次特征, 能够获得文本的深层双向表示, 是一种动态的文本表示方法, 解决了一词多义的问题. 因此, 本文将基于 BERT 模型对长文本匹配展开深入研究. BERT 模型最多支持输入 510 个字符, 这对于文档级别的文本来说远远不够, 那么就需要将文档灵活地转变成可被 BERT 模型处理的形式, 主要有 3 类方法: 一是截断法, 主要包括截取文档头部分段、截取文档尾部分段、截取文档头尾部分分段 3 种方式, 截断法总共截取 510 个字符, 但对于超长文档, 必然丢失大量文本信息; 二是分段法, 采用“字符-分段-文档”的分层思想, 将整个文档拆分成多个固定长度的分段, 每个分段的字符数不超过 510, 再通过 BERT 模型计算每个分段的向量表示, 最后池化得到文档向量, 该方法没有考虑分段的先后顺序和相互联系, 容易丢失部分语义信息; 三是压缩法, 一般而言, 文档中每一段的第一句为关键句, 将这些关键句抽取出来重新组成文档, 若超过 510 字符, 则采用截断法, 此方法可能丢失一些关键的文本信息. 针对以上问题, 本文改进分段法, 提出一种基于 BERT 的长文本匹配模型 LTM-B, 该模型建立在孪生网络的基本框架上, 考虑到数据集中每个文档的字符数, 首先将文档拆分成 4 个分段, 经过 BERT 模型处理产生 4 个文本向量, 由此组成文档矩阵, 再利用双向长短时记忆网络 (BiLSTM)[9]模型得到位置矩阵, 然后将文档矩阵和位置矩阵求和送入 Transformer 编码器[10]进行特征提取, 最后在匹配层使两个文档矩阵交互, 并让两个文档矩阵进行池化、拼接操作, 经由全连接层分类输出两篇文档之间的匹配关系. 实验结果表明, 相较于其他方法, 本文提出的 LTM-B 模型有着更好的效果.

1 相关技术

孪生网络[11]包含两个结构相同、权重共享的子网络, 子网络各自接收一个输入, 将其映射至高维特征空间, 并输出对应的表示, 然后通过计算两个表示的距离得到两个输入之间的语义关系.

Transformer 模型是谷歌在 2017 年推出的一种 NLP 模型[10], 它由编码器-解码器结构组成. 模型使用了自注意力机制, 没有采用循环神经网络 (RNN)[12]的顺序结构, 能够并行化训练, 拥有非常优秀的特征提取能力.

BERT 模型是谷歌在 2018 年推出基于 Transformer 的预训练模型[8], 在 NLP 领域的多个方向大幅刷新了纪录, BERT 模型作为 Word2Vec[13]的替代者, 它的网络架构使用了多层带有 Attention 机制[14]的 Transformer 结构, 相较于 Word2Vec 的浅层神经网络, BERT 模型是深层且动态的, 可以解决一次多义的问题, 对于词和句的向量化处理有突出的贡献.

2 长文本匹配模型 LTM-B

目前, 大多数长文本匹配方法要么采用像 Word2Vec 这样的静态浅层网络作为文本表示方式, 存在一词多义问题, 要么采用了动态的文本表示方式却丢失了许多语义信息.

因此, 本文提出了一种基于 BERT 的长文本匹配模型 LTM-B, 其结构如图 1 所示, 它以孪生网络为基础, 拥有输入层、表示层和匹配层. 在输入层, 文档分段后通过 BERT 模型得到文档矩阵, 并利用 BiLSTM 模型产生位置矩阵, 将两个矩阵之和送入表示层; 表示层为 Transformer 编码器, 能够对文档矩阵进行深层次特征提取; 匹配层对两个文档进行交互, 并将池化后的两个文档向量拼接输入至全连接层, 最终分类输出两个文档之间的匹配关系.

2.1 输入层

模型的输入层是将文档转化成矩阵表示, 先对文

档进行分段处理. 而数据集中单篇文档大多集中在 700–2 000 个字符之间, 并考虑到 BERT 模型的输入序列最多不能超过 510 个字符, 本文将文档前 2 040 个字符截取下来, 分成 4 段, 每段最多为 510 个字符. 若文档未能达到 4 段, 则按照 4 段处理, 例如单个文档的字符数为 723, 那么第 1 段为 510 个字符, 第 2 段为 213 个字符, 第 3 段和第 4 段为空, 即设为全零向量.

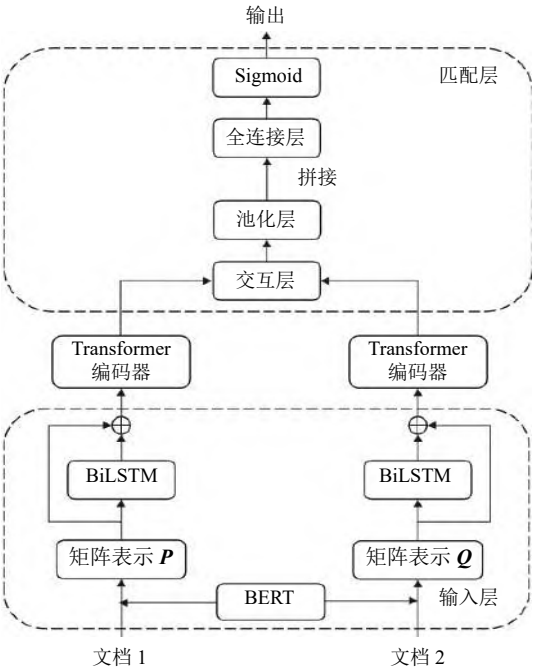

图 1 LTM-B 模型结构

本文使用的是 BERT-Base 中文模型, 该模型拥有 12 层 Transformer 编码器, 隐藏层的维度是 768, 自注意头的个数为 12. 因此, 通过 BERT 模型可以分别得到两个文档的矩阵表示 $\boldsymbol{P}_{4\times768}$ 和 $\boldsymbol{Q}_{4\times768}$. BERT 模型实现文本向量化的过程如图 2 所示.

在得到两个矩阵表示后, 考虑到 Transformer 编码器不能获取位置信息, 本文使用 BiLSTM 模型来得到位置矩阵, 将文档矩阵和位置矩阵相加得到文档的输入矩阵表示, 其公式如下:

$$\boldsymbol{P} = \boldsymbol{P} + Softmax(\text{BiLSTM}(\boldsymbol{P})) \tag{1}$$

$$\boldsymbol{Q} = \boldsymbol{Q} + Softmax(\text{BiLSTM}(\boldsymbol{Q})) \tag{2}$$

2.2 表示层

两篇文档经过输入层后产生各自的文档矩阵表示, 再通过表示层对文档进行特征提取, 表示层的权重矩阵是共享的.

模型的表示层采用 Transformer 编码器, 每一个 Transformer 编码器都有两层子结构: 自注意力层和前馈神经网络 (FNN) 层[15]. 每层结构后都会进行残差连接和层归一化处理, 从而保证每层的输出数据更加平滑, Transformer 编码器结构图如图 3 所示.

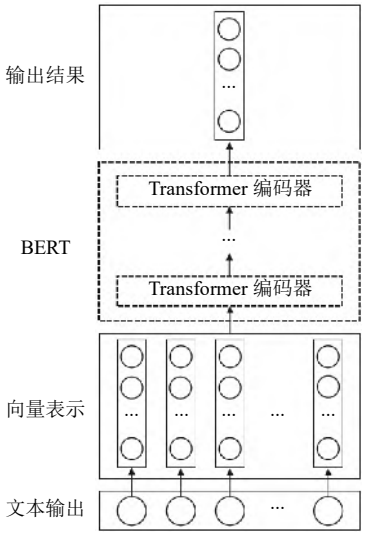

图 2 文本向量化过程

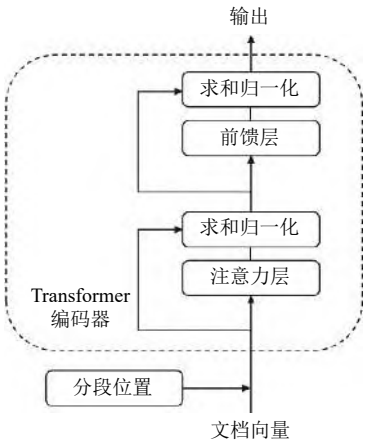

图 3 Transformer 编码器结构

本文使用多个 Transformer 编码器作为表示层来对矩阵做特征提取, 从而增强文档的矩阵表示, 单个 Transformer 编码器计算过程如下步骤.

(1) 文档矩阵: 通过输入层产生的文档矩阵表示 $\boldsymbol{X}$, 矩阵 $\boldsymbol{X}$ 的维度是 4×768.

(2) 计算矩阵 $\boldsymbol{Q}$、$\boldsymbol{K}$、$\boldsymbol{V}$: 通过模型的参数 $\boldsymbol{W}^{\boldsymbol{Q}}$、$\boldsymbol{W}^{\boldsymbol{K}}$、$\boldsymbol{W}^{\boldsymbol{V}}$ 结合矩阵输入 $\boldsymbol{X}$ 来进行计算:

$$\boldsymbol{Q} = \boldsymbol{W}^{\boldsymbol{Q}}\boldsymbol{X} \tag{3}$$

$$\boldsymbol{K}=\boldsymbol{W}^{\boldsymbol{K}}\boldsymbol{X} \tag{4}$$

$$\boldsymbol{V}=\boldsymbol{W}^{\boldsymbol{V}}\boldsymbol{X} \tag{5}$$

(3) 计算单头自注意力层的输出矩阵 $\boldsymbol{Z}$: 首先计算字符在上下文中的意义以及字符之间的相互影响 $\boldsymbol{QK}$, 之后进行缩放和归一化处理, $d_{\boldsymbol{K}}$ 默认值为 64, 最后加权求和得到单头注意力层输出矩阵 $\boldsymbol{Z}$:

$$\boldsymbol{Z}=Softmax\left(\frac{\boldsymbol{QK}^{\mathrm{T}}}{\sqrt{d_{\boldsymbol{K}}}}\right)\boldsymbol{V} \tag{6}$$

(4) 计算融合所有注意力头信息的矩阵 $\boldsymbol{Z}_{\mathrm{sum}}$: Transformer 编码器模型使用了 m 个注意力头, 通过第 3 步可以得到 m 个不同的 $\boldsymbol{Z}$ 矩阵, 将它们拼接并乘以附加的权重矩阵 $\boldsymbol{W}^{\boldsymbol{O}}$ 可得到 $\boldsymbol{Z}_{\mathrm{sum}}$, $\boldsymbol{Z}_{\mathrm{sum}}$ 的维度为 4×768. 公式中 Concat($\boldsymbol{Z}_i$) 表示为 m 个注意力头输出矩阵的拼接:

$$\boldsymbol{Z}_{\mathrm{sum}}=Concat(\boldsymbol{Z}_i)\boldsymbol{W}^{\boldsymbol{O}} \tag{7}$$

(5) 通过残差连接和层归一化得到 $\boldsymbol{Z}_a$: 将注意力层输出结果的 $\boldsymbol{Z}_{\mathrm{sum}}$ 和输入矩阵 $\boldsymbol{X}$ 相加后做层归一化 (LayerNorm) 得到 $\boldsymbol{Z}_a$ 直接作为前馈层的输入, $\boldsymbol{Z}_a$ 的维度是 4×768:

$$\boldsymbol{Z}_a=LayerNorm(\boldsymbol{Z}_{\mathrm{sum}}+\boldsymbol{X}) \tag{8}$$

(6) 前馈层即前馈神经网络层: 首先对 $\boldsymbol{Z}_a$ 进行两次线性转换, 然后使用激活函数处理得到输出 $\boldsymbol{Z}_h$, $\boldsymbol{Z}_h$ 的维度是 4×768:

$$\boldsymbol{Z}_h=FNN(\boldsymbol{Z}_a) \tag{9}$$

(7) 通过残差连接和层归一化得到 $\boldsymbol{Z}_{\mathrm{out}}$: 将前馈层得到的结果 $\boldsymbol{Z}_h$ 和输入 $\boldsymbol{Z}_a$ 相加后做层归一化处理得到 Transformer 编码器的输出 $\boldsymbol{Z}_{\mathrm{out}}$, $\boldsymbol{Z}_{\mathrm{out}}$ 的维度是 4×768:

$$\boldsymbol{Z}_{\mathrm{out}}=LayerNorm(\boldsymbol{Z}_h+\boldsymbol{Z}_a) \tag{10}$$

表示层通过 Transformer 编码器来对文档进行特征提取, 不同的分段具有不同的注意力权重, 能够体现分段之间的相互联系, 其输出矩阵可以更好地表示文档特征.

2.3 匹配层

使用表示层将文档特征提取后, 需要对两篇文档进行语义匹配计算, 本文模型的匹配层主要包括交互层和全连接层.

交互层主要是让两篇文档分别获取相互之间的注意力, 这对于文档之间语义匹配度的判断是非常有必要的. 参考缩放点积注意力机制, 交互层对表示层输出的文档矩阵 $\boldsymbol{P}_{4\times768}$ 和文档矩阵 $\boldsymbol{Q}_{4\times768}$ 进行如下处理:

$$\boldsymbol{P}=Softmax\left(\frac{\boldsymbol{PQ}^{\mathrm{T}}}{8}\right)\boldsymbol{P} \tag{11}$$

$$\boldsymbol{Q}=Softmax\left(\frac{\boldsymbol{QP}^{\mathrm{T}}}{8}\right)\boldsymbol{Q} \tag{12}$$

表示层考虑的是文档内部分段之间的相互影响, 而交互层考虑的是两篇文档之间的影响.

通过交互层获得交互信息后, 下一步是对文档全局特征进行整合, 本文采用的是最大池化 (max pooling), 得到两个文档的特征向量 $\boldsymbol{p}$ 和 $\boldsymbol{q}$, 二者都为 768 维;

$$\boldsymbol{p}=MaxPooling(\boldsymbol{P}) \tag{13}$$

$$\boldsymbol{q}=MaxPooling(\boldsymbol{Q}) \tag{14}$$

将最大池化后的文档向量 $\boldsymbol{p}$、$\boldsymbol{q}$ 和$|\boldsymbol{p}-\boldsymbol{q}|$进行拼接, 其中$|\boldsymbol{p}-\boldsymbol{q}|$可以有效反映两个文档之间的差异, 从而得到一个 2 304 维的组合向量 $\boldsymbol{c}$;

$$\boldsymbol{c}=Concat(\boldsymbol{p},\boldsymbol{q},|\boldsymbol{p}-\boldsymbol{q}|) \tag{15}$$

最后通过全连接层和激活函数的处理, 将组合向量 $\boldsymbol{c}$ 映射输出为 (0, 1) 的数值, 该数值 D 越大说明两个文档之间的距离越大, 相似度越低, 公式如下:

$$D=Sigmoid(F(\boldsymbol{c})) \tag{16}$$

2.4 损失函数

损失函数 (loss function) 是用于评价模型的输出值与实际值不一样的程度, 也可用于修正模型的权重矩阵, 最后通过最小化损失函数来达到模型最好效果. 当 Sigmoid 函数作激活函数使用时, 通常使用的损失函数是交叉熵损失函数 (cross-entropy loss function), 公式如下:

$$loss=-\frac{1}{n}\sum_{x}[y\ln a+(1-y)\ln(1-a)] \tag{17}$$

其中, x 表示样本, y 表示实际值, a 表示模型的输出值, n 表示样本的数量.

3 实验

3.1 数据集

本文实验使用的数据集来自清华大学的 THUCNews 新闻文本分类数据集, THUCNews 数据集是根据新浪新闻 2005–2011 年间的历史数据筛选过滤生成, 包含 74 万篇新闻文档, 均为 UTF-8 纯文本格式. 此数据集在原始新浪新闻分类体系的基础上, 重新整合划分出 14 个候选分类类别: 财经、彩票、房产、股票、家

居、教育、科技、社会、时尚、时政、体育、星座、游戏、娱乐.

实验选取 THUCNews 数据集的部分数据来构造长文本匹配数据集, 数据集共有 10 000 个文档对, 以 6:2:2 切分为训练集、测试集和验证集. 在 THUCNews 数据集的 14 个类别文档中抽取某一类中的两篇文档作为一个同类文档对, 标注距离为 0, 抽取 5 000 对; 再抽取非同类中的两篇文档作为一个异类文档对, 标注距离为 1, 同样抽取 5 000 对. 数据集中数据的分布如表 1 所示.

表 1 数据集分布

数据集	同类	非同类
训练集	3 000	3 000
测试集	1 000	1 000
验证集	1 000	1 000

3.2 评价标准

实验使用的评价标准是准确率 A (accuracy) 和 $F1$ 值, $F1$ 值由精确率 P (precision) 和召回率 R (recall) 计算得到. 准确率 A 定义为正确分类的样本数与总样本数之比, 精确率 P 表示为预测为正的样本数与真正的正样本数的比例, 召回率 R 表示为样本正例中预测正确的比例. A、P、R、$F1$ 值的计算如以下公式所示, 其中, TP 表示“实际是正类, 预测是正类”, FP 表示“实际是负类, 预测是正类”, FN 表示为“实际是正类, 预测是负类”, TN 表示为“实际是负类, 预测是负类”.

$$A=\frac{TP+TN}{TP+FP+FN+TN} \tag{18}$$

$$P=\frac{TP}{TP+FP} \tag{19}$$

$$R=\frac{TP}{TP+FN} \tag{20}$$

$$F1=\frac{2\times P\times R}{P+R} \tag{21}$$

3.3 模型参数设置

本文提出的 LTM-B 模型涉及到的超参数有许多, 主要的超参数包括 Transformer 编码器的层数、注意力头的个数、批大小, 下面通过控制变量法来设置这 3 个超参数.

Transformer 编码器的层数设置关系到表示层的复杂程度, 通常层数越多, 训练时间越长, 模型耗时越多. 因此, 找到层数少且效果好的模型是非常有必要的. 本次实验为二分类实验, 较为简单, 可以设置 Transformer 编码器的层数为 1、2、3、4, 实验结果如图 4 所示.

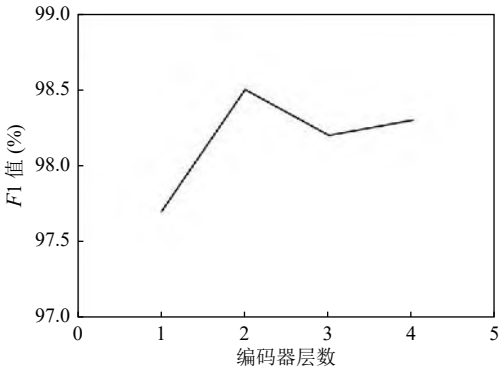

图 4 $F1$ 值随编码器层数变化图

注意力头个数的增加可以提高表示层的性能, 但个数过多可能导致模型过拟合, 本次实验设置注意力头的个数为 4、8、12、16, 实验结果如图 5 所示.

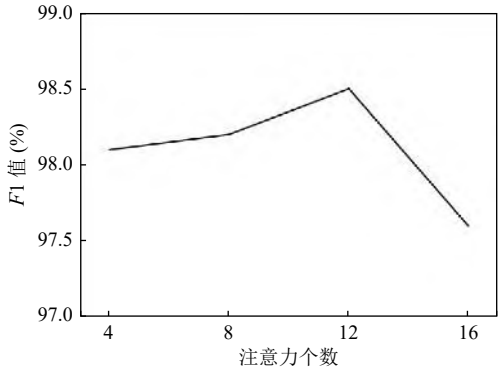

图 5 $F1$ 值随注意力头个数变化图

批大小关系着模型训练的效果, 批次太小不利于收敛, 批次太大容易陷入局部最小值, 分别设为 64、128、256、512 进行实验, 其结果如图 6 所示.

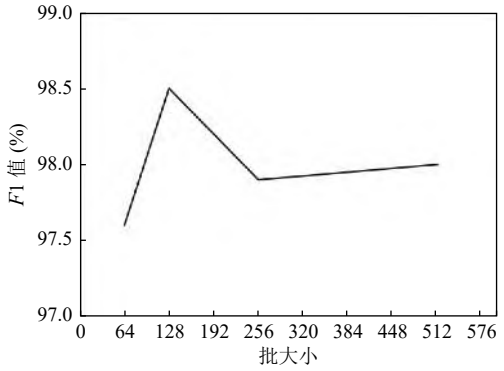

图 6 $F1$ 值随批大小变化图

综上所述, 我们将 Transformer 编码器层数设置为 2, 注意力头的个数设置为 12, 批大小设定为 128.

3.4 实验对比

为验证本文提出的 LTM-B 模型的效果, 本文设置了 6 组对比实验, 实验方法如下所示:

(1) BERT+截断法: 截取文档前 510 个字符, 两篇文档分别经过 BERT 模型处理得到文档向量, 两个向量经过拼接送入全连接层分类输出匹配结果, 计算结果靠近 0 则划分成同类, 计算结果接近 1 则为异类;

(2) Word2Vec+截断法: 截取文档前 510 个字符, 采用 Word2Vec 模型结合 jieba 分词来得到文档的矩阵表示, 后然后通过最大池化得到文档向量, 再将两个文档向量拼接后经过全连接层分类输出匹配结果;

(3) 分段法: 将文档分成多个分段, 每个分段 510 字符, 每个分段分别使用 BERT 模型处理成向量表示, 组成文档的矩阵表示, 然后直接简单地将两个文档矩阵输入到匹配层得到结果;

(4) 压缩法: 将文档中每个段落的第一句抽出组成一个新文本, 文本若超过 510 字符, 则只截取前 510 字符, 将此文本通过 BERT 模型处理成文本向量, 然后对两个文本向量依次经过拼接等操作输出结果;

(5) 分段法+BiLSTM: 文档截取前 4 段, 每段 510 字符, 然后使用 BERT 模型处理得到文档矩阵表示, 再由 BiLSTM 实现特征提取, 后经过匹配层得到结果;

(6) 分段法+Transformer: 相对本文提出的 LTM-B 模型, 该方法的输入层缺少位置矩阵.

实验结果如表 2 所示.

表 2 模型对比实验 (%)

编号	方法	*P*	*R*	*F*1	*A*
1	BERT+截断法	95.4	92.5	93.8	94.0
2	Word2Vec+截断法	70.6	71.0	70.8	71.1
3	分段法	95.5	95.3	95.4	95.4
4	压缩法	94.1	96.4	95.3	95.2
5	分段法+BiLSTM	97.3	97.8	97.5	97.5
6	分段法+Transformer	**98.9**	97.2	98.0	98.1
7	LTM-B	98.4	**98.6**	**98.5**	**98.5**

3.5 实验结论

由方法 1、2 可知, BERT 模型的文本表示能力比 Word2Vec 模型更加优秀; 由方法 1、3、4 可知, 截断法比分段法和压缩法效果都差一点, 主要是因为截断法丢失了更多的文本信息; 由方法 3、5、6 可知, 增加表示层能够提取更多的文本特征, 且 Transformer 编码器在特征提取方面要优于 BiLSTM; 由方法 6、7 可知, 加入位置信息确实能够提升模型的匹配效果. 本文提出的 LTM-B 模型在长文本拥有更好的表现, 其原因在于:

(1) BERT 模型是一种深层动态的文本表示方法, 融合文本的多层次特征, 很好地解决了一词多义问题;

(2) 输入层采用“字符-分段-文档”的分层思想, 最大程度保留了文本信息, 并利用 BiLSTM 产生位置信息, 解决了 Transformer 编码器不能获取文本时序特征的问题;

(3) 表示层利用了 Transformer 编码器强悍的特征提取能力, 能够捕获更多的文本特征;

(4) 匹配层添加了交互层, 使得两篇文档能够进行信息交互, 这有利于加强模型的匹配效果.

4 结论

本文紧紧围绕 BERT 模型对长文本匹配问题进行深入研究, 剖析 3 种常用方法后提出 LTM-B 模型. 该模型以孪生网络为基础, 在输入层对文档进行分段, 利用 BERT 模型得到文档矩阵, 并通过 BiLSTM 生成位置矩阵, 两个矩阵求和后送入由 Transformer 编码器构成的表示层做特征提取, 最后在匹配层进行交互、池化、拼接后输入到全连接层通过激活函数分类输出匹配结果. 实验证明, 本文提出的 LTM-B 模型在长文本匹配问题上具有不俗的表现.

参考文献

1 庞亮, 兰艳艳, 徐君, 等. 深度文本匹配综述. 计算机学报, 2017, 40(4): 985–1003.

2 余同瑞, 金冉, 韩晓臻, 等. 自然语言处理预训练模型的研究综述. 计算机工程与应用, 2020, 56(23): 12–22. [doi: 10.3778/j.issn.1002-8331.2006-0040]

3 秦永彬, 孙玉洁, 魏笑. 基于文本聚类与兴趣衰减的微博用户兴趣挖掘方法. 计算机应用研究, 2019, 36(5): 1469–1473.

4 王绍卿, 李鑫鑫, 孙福振, 等. 个性化新闻推荐技术研究综述. 计算机科学与探索, 2020, 14(1): 18–29. [doi: 10.3778/j.issn.1673-9418.1908024]

5 焦桐, 肖源. 大数据搜索引擎下的知识产出机制研究. 情报科学, 2018, 36(3): 33–38.

6 汤建明, 寇小强. 海量网络文本去重系统的设计与实现. 计算机应用与软件, 2018, 35(12): 33–37.

7 钟文康, 葛季栋, 陈翔, 等. 面向神经机器翻译系统的多粒度蜕变测试. 软件学报, 2021, 32(4): 1051–1066. [doi: 10.13328/

j.cnki.jos.006221]

8 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.

9 李洋, 董红斌. 基于 CNN 和 BiLSTM 网络特征融合的文本情感分析. 计算机应用, 2018, 38(11): 3075–3080. [doi: 10.11772/j.issn.1001-9081.2018041289]

10 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Neural Information Processing Systems. Los Angeles: Curran Associates Inc., 2017. 6000–6010.

11 付利华, 赵宇, 孙晓威, 等. 基于孪生网络的快速视频目标分割. 电子学报, 2020, 48(4): 625–630. [doi: 10.3969/j.issn.0372-2112.2020.04.001]

12 杨丽, 吴雨茜, 王俊丽, 等. 循环神经网络研究综述. 计算机应用, 2018, 38(S2): 1–6, 26.

13 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe: Curran Associates Inc., 2013. 3111–3119.

14 Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. Computer Science, 2014, 24(5): 1345–1349.

15 Frean M. The upstart algorithm: A method for constructing and training feedforward neural networks. Neural Computation, 1990, 2(2): 198–209. [doi: 10.1162/neco.1990.2.2.198]