计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210300266

基于预训练和深度哈希的大规模文本检索研究

邹 傲 郝文宁 靳大尉 陈 刚 田 媛
陆军工程大学指挥控制工程学院 南京 210000
(3231954713@qq.com)

摘 要 针对文本检索中存在的检索效率和准确率不高的问题,提出一种基于预训练语言模型和深度哈希方法的检索模型。该模型首先通过迁移学习的方法引入预训练语言模型中所包含的文本先验知识,之后进行特征提取,将输入转化为高维的向量表示。在整个模型的后端加入哈希学习层,通过设计特定的优化目标对模型的参数进行微调,从而在训练中动态地学习哈希函数和每个输入的唯一哈希表示。实验表明,该方法的检索准确率相较于其他基准模型在 top-5 和 top-10 指标上分别有至少 21.70%和 21.38%的提升,哈希码的引入使得模型在仅损失 4.78%准确率的前提下将检索速率提升了 40 倍,因此该方法能够显著提升检索准确率和效率,且在文本检索领域有着潜在应用前景。

关键词:深度学习;相似性检索;预训练语言模型;深度哈希
中图法分类号 TP391.1

Study on Text Retrieval Based on Pre-training and Deep Hash

ZOU Ao,HAO Wen-ning,JIN Da-wei,CHEN Gang and TIAN Yuan
Command & Control Engineering College,Army Engineering University of PLA,Nanjing 210000,China

Abstract Aiming at the problem of low retrieval efficiency and accuracy in text retrieval,a retrieval model based on pre-trained language model and deep hash method is proposed. Firstly,the prior knowledge of text contained in the pre-trained language model is introduced by transfer learning,and then the input is transformed into high-dimensional vector representation by feature extraction. A hash learning layer is added to the back end of the whole model to fine tune the parameters of the model by designing specific optimization objectives,so as to dynamically learn the hash function and the unique hash representation of each input in the training. Experimental results show that the retrieval accuracy of this method is at least 21.70% and 21.38% higher than that of other benchmark models in top-5 and top-10,respectively. The introduction of hash code makes the model improve the retrieval speed by 40 times under the premise of only losing 4.78% accuracy. Therefore,this method can significantly improve the retrieval accuracy and efficiency,and has a potential application prospect in the field of text retrieval.

Keywords Deep learning,Similarity retrieval,Pre-trained language model,Deep hash

1 引言

近年来,由于各种文本形式网络数据的快速增长,在自然语言处理领域,大规模文本检索的研究受到了广泛关注。文本检索,亦称为自然语言检索,主要研究如何从无结构的文档集中找到与查询语句相关的文档子集,并根据相关度排行把检索结果返回给用户。文本检索需要满足准确率(Accuracy)、查全率(Recall)和查准率(Precision)等衡量检索结果准确性的评价指标,同时也要兼顾检索效率,尽量减少检索过程所花费的时间。

绝大多数传统文本检索方法的原理是统计查询项与文档词项之间的重复程度,查询与文档文本之间的精确词项匹配是许多检索系统的基础[1]。不同的权重和正则化机制的使用催生了一系列无监督方法,如 TF-IDF 和 BM25[2]等。在有监督的方法被引入后,LTR(Learning to Rank)方法逐渐成为主流,融入语义的文本检索方法不断出现,超越了传统的基于词项频率统计的方法[3]。随着神经网络模型和深度学习的兴起,面向文本检索的深度神经网络模型开始出现[4-6]。有监督方法的引入、复杂的文本语义提取模式以及深度学习基于数据驱动的特点导致深度神经网络检索模型在训练和使用中需要耗费更多的时间成本,从而造成检索效率不高的问题。

为保证文本检索系统同时具有较高的准确率和效率,本文提出了一种基于预训练语言模型和深度哈希编码的检索技术。通过预训练语言模型可以引入丰富的语义先验知识,从而使得检索模型能够更快地收敛,大幅节约训练成本并提高检索准确率。与此同时,在模型的后端引入一个哈希学习层,

到稿日期:2021-03-26 返修日期:2021-05-23
基金项目:国家自然科学基金(61806221)
This work was supported by the National Natural Science Foundation of China(61806221).
通信作者:郝文宁(hwnbox@163.com)

能够将最终输出的高维浮点型特征向量转化为相应维度的哈希码,使用输出的哈希码作为特征能够减少部分存储空间,并显著提升检索速度。

2　相关研究

本文提出的新的检索方法涉及深层次语义挖掘和深度哈希两方面内容。

近年来,对文本语义挖掘的研究经历了神经网络语言模型—词向量—预训练语言模型3个阶段。预训练语言模型是当前自然语言处理领域中最成熟、性能最强、应用最广泛的解决方案[7],相较于传统的词向量方法[8-10]其能够更充分地挖掘文本中蕴含的语义信息。预训练语言模型方法不仅在学术界各种任务的公开评测数据集,如SQuAD[11]和RACE[12]上取得了亮眼效果,而且在工业界也有不少成熟的落地应用。

主流的预训练模型以Transformer Encoder[13]为基本单元构建复杂的深度神经网络结构,通过构造特定的预训练任务训练模型参数,训练后的模型在下游任务中只需进行参数微调便能够取得很好的结果,代表性的预训练模型有BERT[14],XLNet[15],RoBERTa[16]等。

具体来讲,Transformer Encoder内部发挥作用的主要是多头注意力机制[17]以及残差连接结构[18]。一系列的研究表明Transformer结构相较于卷积神经网络能够捕捉更多的语义信息,且有着比循环神经网络更快的训练速度,是当前深度学习领域使用最广泛的特征提取器。

基于哈希技术进行检索的方法的基本思想是:构造一系列哈希函数,将每个文本对象映射成一个二进制特征向量,从而将语义上相似的样本映射成相似的二进制码。将高维实数特征向量编码成低维紧致二进制码,可以在检索过程中显著加速相似度计算,并能节省内存中的存储空间。现有的基于哈希的方法主要可分为两类:data-independent(数据无关)和data-dependent(数据相关)。第一种方法通常使用随机映射函数将样本映射到特征空间,再进行二值化计算二进制码,代表性的方法有LSH[19](局部敏感哈希)及其拓展方法E^2LSH[20](欧氏局部敏感哈希)等,之前的研究也有使用LSH进行文本检索的思路[21]。第二种方法基于数据驱动,利用统计学习的方法学习哈希函数,将样本映射成二进制码,代表方法有谱哈希[22]等。2014年以来,图像检索领域出现了一系列将深度神经网络和哈希函数相结合的方法[23-26]:深度哈希方法。该方法同样属于数据驱动,通过深度神经网络在训练过程中自动学习生成哈希函数,相较于人工设计或通过传统机器学习方法哈希函数的学习效果更好,其查找准确率和效率都有较大提升,且在检索之外的其他领域也有应用[27]。

采用深度神经网络的方法已经在多个信息检索任务中取得了较大的性能提升[28],自预训练语言模型成为自然语言处理的主流方法以来,基于BERT和Transformer的研究[29-32]在文档检索领域取得了较好的效果。Yan等[30]采用了篇章级别的BERT模型,通过相关性估计进行文档的检索。Hofstatter等[31]针对深度模型在文本检索中效率不高的问题,提出了一种Transformer-Kernel结构,对包含多层Transformer的预训练语言模型进行简化,在保证一定检索准确率的基础上提升了检索效率。Macavaney等[33]将经过预训练的上下文嵌入用于下游的检索任务,取得了一定成果。尽管深度模型的使用带来了一定的性能提升,但基于自注意力机制方法的时间复杂度和空间复杂度较高,假设输入序列长度为L,则其时间复杂度和空间复杂度都是$O(L^2)$。若直接将基于Transformer的模型应用于文本检索,则需要在内存中始终维持全部文档集之间的注意力信息,这会造成难以承受的空间开销,因此需要对其进行改进。

预训练语言模型、深度哈希已经分别在文本语义表示、大规模图像检索等领域取得了较好的实验效果。受到之前研究的启发,本文针对文本检索问题,设计了一种模型,能够综合预训练语言模型包含大量语义信息以及深度哈希在检索效率方面的优势,从而同时获得较高的检索准确率和效率。

综上,本文的工作主要包含以下方面:

(1)提出了一种基于预训练语言模型和有监督深度哈希算法的文本检索模型。

(2)通过实验验证该模型在相应的数据集上经过训练和参数微调能够获得较高的检索准确率。

(3)经过对比实验验证该模型比传统检索方法在检索准确率和效率方面具有较大优势。

3　模型结构

本文提出的方法本质上是一种基于语义的检索模型,在成对标签(pair-wise labels)的训练数据上进行有监督的哈希学习,该模型总体结构如图1所示。为方便说明,表1列出了本文所用符号标记及含义。

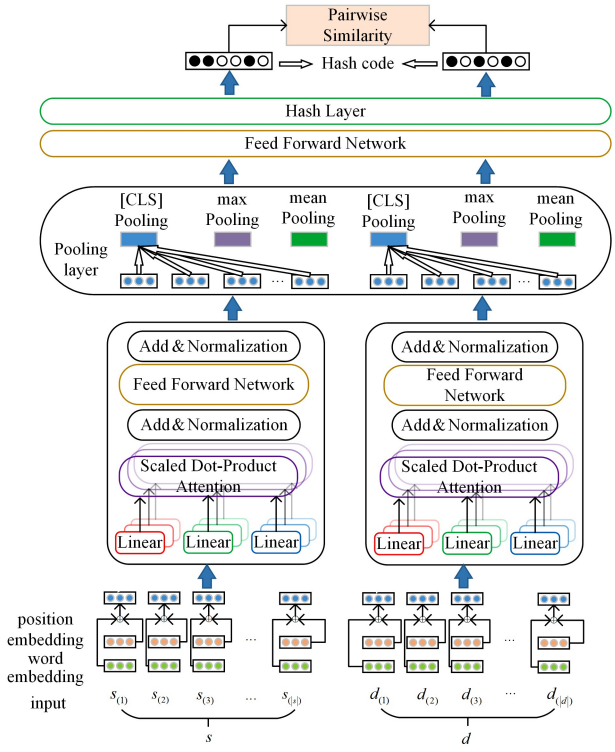

图1　基于预训练语言模型和深度哈希的检索模型的结构图

Fig. 1　Structure diagram of retrieval model based on pre-trained language model and deep hash

表 1 本文所用符号

Table 1 Notations used in this paper

文本描述	符号标记
单个查询语句	s
单个文档	d
查询语句集合	S
文档集合	D
查询语句 s 中的词项	t_s
文档 d 中的词项	t_d
查询语句 s 的检索结果的集合	R_s
结果元组(文档 d 排在第 i 位)	$\langle i,d\rangle$, where $\langle i,d\rangle \in R_s$
文档 d 对查询语句 s 的相关性标签	$rel_s(d)$

训练过程中,模型的输入为查询语句 s 与文档 d 组成的成对定长文本序列(s,d)和相关性标签 $rel_s(d)$,$rel_s(d)\in\{0,1\}$。$rel_s(d)=1$ 表示输入的语句 s 与文档 d 在语义上相似,$rel_s(d)=0$ 则表示输入的语句 s 与文档 d 在语义上不相似。有监督深度哈希的训练目标是为每一个语句 s 与文档 d 都学习一个哈希码 $b\in\{0,1\}^c$,最终哈希码的集合 $B=\{b_i\}_{i=1}^{|D|+|S|}$ 应能保存每一个标签 $rel_s(d)$所蕴含的相似度信息。具体地,若 $rel_s(d)=1$,s 与 d 的哈希表示 b_s 和 b_d 应该有较小的汉明距离;相反,若 $rel_s(d)=0$,s 与 d 的哈希表示 b_s 和 b_d 应该有较大的汉明距离。

对输入到模型的语句 s 与文档 d 设置输入序列长度上界,对输入文本序列中多余和不足部分进行截断和填充。经过分词和向量化后,s 和 d 分别转化为 $\boldsymbol{S}\in\mathbb{R}^{|s|\times n_{\dim}}$ 和 $\boldsymbol{D}\in\mathbb{R}^{|d|\times n_{\dim}}$。对于模型的输入 $\boldsymbol{S}$ 和 $\boldsymbol{D}$,使用 3 个线性变换 $\boldsymbol{W}_K$,$\boldsymbol{W}_Q$,$\boldsymbol{W}_V$ 分别与输入 $\boldsymbol{S}$ 和 $\boldsymbol{D}$ 相乘,可得矩阵 $\boldsymbol{K},\boldsymbol{Q},\boldsymbol{V}$,用以计算注意力权重:

$$Attention_{\boldsymbol{S}}(\boldsymbol{K},\boldsymbol{Q},\boldsymbol{V})=\mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{n_{\dim}}}\right)\boldsymbol{V} \tag{1}$$

$$Attention_{\boldsymbol{D}}(\boldsymbol{K},\boldsymbol{Q},\boldsymbol{V})=\mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{n_{\dim}}}\right)\boldsymbol{V} \tag{2}$$

首先将输入 $\boldsymbol{S}$ 和 $\boldsymbol{D}$ 通过 Transformer Encoder 的注意力层,得到:

$$\widetilde{\boldsymbol{S}}=LayerNorm(\boldsymbol{S}+Attention(\boldsymbol{K},\boldsymbol{Q},\boldsymbol{V})\times\boldsymbol{S}) \tag{3}$$

$$\widetilde{\boldsymbol{D}}=LayerNorm(\boldsymbol{D}+Attention(\boldsymbol{K},\boldsymbol{Q},\boldsymbol{V})\times\boldsymbol{D}) \tag{4}$$

其中,$LayerNorm()$为正则化操作 Layer Normalization[34]。$\widetilde{\boldsymbol{S}}$ 和 $\widetilde{\boldsymbol{D}}$ 再经过一个前馈网络层得到最终输出:

$$\widetilde{\widetilde{\boldsymbol{S}}}=LayerNorm(\widetilde{\boldsymbol{S}}+FFN(\widetilde{\boldsymbol{S}})) \tag{5}$$

$$\widetilde{\widetilde{\boldsymbol{D}}}=LayerNorm(\widetilde{\boldsymbol{D}}+FFN(\widetilde{\boldsymbol{D}})) \tag{6}$$

其中,$FFN()$为一个由两层线性变换层和一个 ReLU 函数组成的前馈神经网络层,若输入为 $\boldsymbol{x}$,则该前馈神经网络层的计算方法如下:

$$FFN(\boldsymbol{x})=\max(0,\boldsymbol{x}\boldsymbol{W}_1+\boldsymbol{b}_1)\boldsymbol{W}_2+\boldsymbol{b}_2 \tag{7}$$

其中,$\boldsymbol{W}_1$,$\boldsymbol{b}_1$,$\boldsymbol{W}_2$,$\boldsymbol{b}_2$ 为线性网络的参数。

预训练语言模型由多层的 Transformer Encoder 结构堆叠而成,将上文计算得出的输出$\widetilde{\widetilde{\boldsymbol{S}}}$和$\widetilde{\widetilde{\boldsymbol{D}}}$作为新的输入进行计算,在经过所有 Transformer Encoder 的计算后再采取不同的池化策略得到整个预训练语言模型的输出。池化的方式有多种,本文尝试了[CLS] pooling,max pooling,mean pooling 3 种池化策略并进行了比较实验,发现直接采用输入语句的[CLS] pooling 池化策略效果最好,因此将其作为模型的池化策略。

在经过预训练模型后,输入的文本序列对(s,d)转变为包含语义的特征向量 $\boldsymbol{s}$ 和 $\boldsymbol{d}$,其中哈希学习层由堆叠的非线性变换层和一个二值化层组成。堆叠的非线性变换层由全连接网络堆叠而成,接收预训练模型的输出 $\boldsymbol{s}$ 和 $\boldsymbol{d}$,将其扩展到更高维度从而增强模型的表达能力,最后再降维输出,得到原始输入的实向量输出。

二值化层则通过设置阈值的方式,对全连接学习层的实向量输出进行映射,将实向量中每一位的值根据阈值映射成"0"或"1",从而将实向量转换成哈希码。具体来讲,对于每一个输入的高维实向量 $\boldsymbol{x}=(x_1,x_2,x_3,\cdots,x_d)$,二值化层将对该向量每一维度的实数值 $x_i(1\leqslant i\leqslant d)$做如下操作:

$$x_i=\begin{cases}1, & x_i\geqslant\dfrac{1}{n}\sum_{i=1}^{n}x_i\\ -1, & x_i<\dfrac{1}{n}\sum_{i=1}^{n}x_i\end{cases} \tag{8}$$

由此可生成 S 和 D 的哈希码集合 $B=\{b_i\}_{i=1}^{|D|+|S|}$,根据相关性标签 $rel_s(d)$定义似然函数如下:

$$p(rel_s(d)\mid B)=\begin{cases}\sigma(\Omega_{s,d}), & rel_s(d)=1\\ 1-\sigma(\Omega_{s,d}), & rel_s(d)=0\end{cases} \tag{9}$$

其中,$\Omega_{s,d}=\frac{1}{2}b_s^{\mathrm{T}}b_d$,$\sigma(\Omega_{s,d})=\frac{1}{1+\mathrm{e}^{-\Omega_{s,d}}}$。根据负对数似然优化 $rel_s(d)$,可得如下优化问题:

$$\begin{aligned}\min_B J_1 &= -\log p(rel_s(d)\mid B)\\ &= -\sum_{rel_s(d)\in L}\log p(rel_s(d)\mid B)\\ &= -\sum_{rel_s(d)\in L}(rel_s(d)\Omega_{s,d}-\log(1+\mathrm{e}^{\Omega_{s,d}}))\end{aligned} \tag{10}$$

对于输入 s 和 d,高维实特征 $\boldsymbol{s}$ 和 $\boldsymbol{d}$ 相较于 b_s 和 b_d 包含更多的语义信息,因此可设计如下优化问题:

$$\min_W J_2=-\log\sum_{s,d\in S,D}(\cos\, sim(\boldsymbol{s},\boldsymbol{d})-rel_s(d))^2 \tag{11}$$

其中,W 是预训练语言模型中的全部参数。

最终的训练目标是由 J_1 和 J_2 构成的联合训练目标:

$$J=J_1+J_2 \tag{12}$$

4 实验过程及结果分析

4.1 实验环境

本文所有实验均采用表 2 所列的实验环境。

表 2 实验环境

Table 2 Experimental environment

操作系统	Ubuntu18.04
CPU	Intel Xeon(R) Silver 4214R CPU @ 2.40GHz×48
GPU	Tesla V100-PCIE-32GB
Python	3.6.10
Pytorch	1.2.0
RAM	64GB

4.2 数据集

本文的实验数据来源是 GLUE[35] 任务集中的 MRPC[36] 数据集和 STS[37] 数据集,格式均为成对数据标签的形式(pair-wise labels)。

上述数据集在预处理后总共获得的数据量为 5173 条。

实验中将数据集按照 8∶1∶1 的比例划分为训练集、验证集和测试集。

4.3　实验设计及思路

基于预训练模型的下游任务应用主要通过两种方法实现，即 Feature-based（基于特征）和 Fine-tune（参数微调）。Feature-based 指利用预训练模型的最后隐层输出结果作为特征直接提取，引入到具体的下游任务中作为输入；Fine-tune 是在已经训练好的预训练模型的基础上，根据具体的下游任务对模型的输出层进行修改，并添加少量与任务相关的参数，然后将整个模型在新的下游任务中进行重新训练。实验中，首先采用 Feature-based 的方法，利用模型最后时刻输出[CLS]的向量表示作为输入文本的向量表示（实验中也选取了输出层所有 token 的向量表示在每个维度上的最大值和平均值来获得相应向量表示的方法进行实验，但效果较差），然后再通过设定阈值的方式，将实数域的向量表示转换为哈希码。在进行检索任务时，通过比较不同语段哈希码之间的汉明距离，便可找出与其语义最接近的语段。根据实验结果，采取这种方法取得的效果并不理想。

本实验的重点在于第二种方法，即采用 Fine-tune 的方法进行参数调优。通过第 3 节设计的模型优化策略，模型将不断对所有参数进行更新，从而能够捕捉到文本更深层次的语义信息，哈希层的作用是根据阈值将实向量映射到汉明空间中，以提高检索效率。后续实验结果表明，此操作会丢失小部分语义信息，造成一定的性能损失，但通过比较哈希码之间汉明距离的检索方式能够极大地提升检索效率。

本文主要进行了如下实验：模型在 Feature-based 和 Fine-tune 两种模式下的性能比较、Feature-based 模式的基准模型与传统无监督检索方法及其他神经网络模型的对比实验、模型的消融研究以及模型的鲁棒性研究。

实验中采用 Huggingface 的 Transformers 项目[38]所提供的开源预训练语言模型调用框架，所使用的模型已在大规模文本语料上进行过大规模训练，有着较强的语义先验知识。以实验中所采用的模型之一 RoBERTa-base 为例，该模型使用 16 GB 的英文语料进行了 10 万次的迭代训练，实验中所采用的其他预训练模型也采用了类似级别的数据量和训练量进行预训练。

4.4　参数设置

实验中，对采用预训练语言模型方法的基准模型的参数做如下规定：批大小设置为 32，学习率设置为 10^{-4}，epoch 设为 3，在学习中采用学习率逐渐递减的策略，每个 epoch 的学习率设为上一个 epoch 的 1/10。优化器采用动量设为 0.9 的 Adam，dropout 设为 0.1。无论采用何种预训练语言模型，最终每个输入所得到的哈希码均为 768 位。通过模型生成的哈希码在测试集上的准确率可以评估训练模型的性能。

本文采用预训练词向量 Word2vec[8]和 GloVe[9]以及一个随机初始化的双向 LSTM[39]作为基准模型，为方便比较，LSTM 的隐层输出同样设为 768 维。其中 Word2vec 和 GloVe 为预训练方法，因此采用与预训练语言模型方法相同的微调 epoch 数，对于 LSTM 模型的训练采用早停策略使其收敛。除此之外，本文选取当前文本检索中较新的研究成果 BERT[CLS][29]和 TKL[32]进行比较实验，所有方法均取 10 次实验的平均值作为该方法的最终结果。

4.5　结果分析

4.5.1　基准模型结果比较

本文主要进行了两方面实验。首先对预训练语言模型的两种调用策略即 Fine-tune 和 Feature-based 进行了对比实验，结果如表 3 所列。随后，在得到 Fine-tune 策略更优的结论后，采用该策略训练本文所提出的模型，并与多种基准方法进行了比较实验，结果如表 4 所列。表 3 和表 4 的实验结果中的 top-5 和 top-10 分别表示采用得分最高的前 5 个或 10 个作为检索返回结果。

表 3　Feature-based 与 Fine-tune 两种策略的比较

Table 3　Comparison of Feature-based and Fine-tune strategies

Model	Feature-based		Fine-tune			
	Top-5	Top-10	Top-5	Imporvement	Top-10	Imporvement
BERT-base-uncased	34.53	36.75	83.35	2.41×	86.57	2.36×
BERT-base-cased	35.69	37.14	79.25	2.41×	83.45	2.25×
RoBERTa-base	29.07	30.55	71.67	2.47×	72.97	2.39×
XLNet	31.48	33.63	76.32	2.42×	82.35	2.45×

由表 3 可以看出，模型采用 Fine-tune 的方法所取得的性能远优于采用 Feature-based 的方法，对于多种预训练语言模型，基于 Fine-tune 的方法比基于 Feature-based 的方法在 top-5 和 top-10 准确率上都能有 2.4 倍左右的提升，该实验结果与预期相符。此外，BERT-base-uncased 模型的效果要优于 BERT-base-cased，这与预期不符，因为 BERT-base-cased 模型的表示能力要强于 BERT-base-uncased，初步分析认为是数据集内的单一文本长度过短，使得在相同训练条件下 BERT-base-cased 模型出现了一定程度的过拟合。

之后，我们以 BERT-base-uncased 模型作为本文方法的基准模型，与其他方法如 Word2vec 等进行比较，实验结果如表 4 所列。结果表明，本文提出的方法在准确率方面已经接近当前研究的最佳水平。在检索效率方面，本文提出的方法相较于全部基线模型都有着较大优势，这是以牺牲部分性能损失获得的效率提升。若将模型中最后的哈希学习层省去，则不存在实向量到哈希码的信息损失，准确率能够得到一定改善，能够与最新的方法相当，但是效率会下降很多。如表 4 所列，本文方法相比传统以及最新研究的模型，其准确率能够达到当前第一梯队水平，检索效率则优于全部参与比较的基准模型，但与最好结果还存在较小差距。

若从模型训练的角度来看，本文提出的方法在算法复杂度方面不占优势，假设输入文本长度为 l，批大小为 b，模型中

注意力头的个数为 n_h，则模型在训练阶段的空间复杂度为 $O(\max(bn_h ln_{\dim}, bn_h l^2))$，时间复杂度为 $O(\max(bn_h ln_{\dim}, bn_h l^2))$，其复杂度与当前主流的基于预训练语言模型的方法（如 BERT[CLS][29]）相当。本文模型在效率方面的优越性主要体现在使用训练好的模型进行预测，其时间复杂度为 $O(\max(|D| n_h ln_{\dim}, |D| n_h l^2))$，而主流方法如 BERT[CLS][29] 的时间复杂度则为 $O(\max((|D| n_h ln_{\dim})^2, (|D| n_h l^2)^2))$，尤其在文档长度较短的场景中本文方法有很大优势，实验结果也证明了这一点。

表 4 本文基准模型与其他方法在测试集上的准确率

Table 4 Results of benchmark model and other methods on test set

（单位：%）

Model	Top-5	Top-10	Time/s	Speed up
TF-IDF	51.24	53.73	43.40	1.53×
BM25	53.76	59.55	38.65	1.72×
Word2vec+FFN	57.35	58.43	18.38	3.61×
GloVe+FFN	61.73	64.48	16.53	4.01×
BiLSTM+FFN	68.49	71.32	43.54	1.52×
BERT[CLS]	82.65	84.23	51.72	1.28×
TKL	**85.42**	84.91	5.92	11.20×
Our method	80.34	82.43	**1.63**	**40.68×**
- pretrained	74.52	76.45	69.42	—
- Fine-tune	34.53	36.75	65.43	—
- hash layer	83.35	**86.57**	66.31	1

4.5.2 消融实验

本文所提出的方法在准确率和效率的优越性上主要依赖于 3 个部分：经过充分预训练的语言模型、基于特定目标的微调步骤以及哈希学习层的引入。为探究模型不同部分的作用，在基准对比实验后进行了消融实验，结果如表 4 的后 4 行所列。

由实验结果可以发现，当模型不采用经过预训练的语言模型，而对模型参数进行随机初始化时，在相同的 Fine-tune 训练过程后，会造成一定的性能损失。另外，若只去掉 Fine-tune 步骤，则模型退化为 Feature-based 模式，前文已经说明该操作会对准确率有较大影响，实验结果也证明了这一点。

模型所生成的高维实向量表示相较于哈希码包含更多的语义信息，直接使用高维实向量作为检索依据能取得更好的效果，但大规模的余弦相似度求解运算比较耗时，相比之下，经过深度哈希学习得到的语句哈希码表示有着更高的查询效率。

为探究哈希层对检索效率的影响，对带有哈希学习层的模型和删去哈希学习层的模型进行了比较实验。实验结果表明，本文模型在不采用哈希码时的检索准确率最高，但耗费时间也最长，比传统方法 TF-IDF 等还要费时。相比之下，采用深度哈希方法的准确率虽然要略低一些(4.9%)，但却能显著提升检索效率。基于 Fine-tune 和深度哈希的方法所消耗的时间主要是在训练过程中，一旦模型训练完毕，将大幅提高下游检索任务的查询效率，而传统方法的优势在于不需要经过额外的训练步骤，只需少量的词频统计即可，但在检索过程中的速度要落后于本文提出的方法。在准确率方面，深度哈希的方法也要远高于传统方法，一方面是由于深度哈希在数据集上经过了学习训练，学习到了更多的语义信息；另一方面，传统方法基于词袋模型，更加适用于较长文本的检索，而本实验采用的数据集的文本单元规模较小，在一定程度上限制了传统方法性能优势的发挥。

4.5.3 模型鲁棒性研究

在文本检索中，算法的性能会被噪声数据所影响，该问题可看作一个噪声鲁棒性问题。为检验模型性能在含噪声数据下的鲁棒性，本文在混有不同比例噪声的数据上进行了对比实验。

具体地，本文采用 WNLI 数据集[40]中的数据作为噪声数据。实验依然在测试集上进行，但会分别引入 5%，10%，15%的噪声数据。从表 5 的结果可看出，本文方法有较好的抗噪鲁棒性，在存在少量噪声数据的情况下并未造成明显的性能损失。

表 5 模型抗噪鲁棒性实验

Table 5 Experiment on anti-noise robustness of the model

（单位：%）

Proportion of noise data	Top-5	Top-10
0	80.34	82.43
5	80.27	82.15
10	78.43	81.94
15	78.15	80.55

综上，实验结果揭示了在预训练语言模型上将 Fine-tune 和深度哈希相结合的方式在文本检索领域有着较好的应用前景。实验证明，在短文本级别的文本检索上，其能够取得不错的准确率和效率。但是本实验中采用的数据集规模依然较小，且短文本级别的检索和实际应用中长文本的检索在细节和实现上面有些许区别，尽管如此，本文的工作为未来文本检索问题的解决提出了一个新的思路，值得继续探究。

结束语 本文针对文本检索的准确率和效率问题，采用预训练语言模型加 Fine-tune 的方式作为解决框架，配合哈希学习层，能够达到较高的检索效率和准确率，即该方法在文本检索领域有着较好的应用前景。相较于传统方法，所提方法的优势主要有以下两点：1）通过预训练语言模型进行迁移学习，能够利用前期所学习到的大量语法语义知识，加深模型对文本的理解程度，使其在检索中能够获得较高的准确率；2）基于深度哈希学习所获得的哈希码进行检索，能够显著提升检索效率。

与此同时，本文的研究也存在一些问题：首先，模型的生成需要花费不少的时间进行相关训练任务的设计以及训练，且如果有大量新的文本集加入可能需要对模型进行补充训练。其次，本文实验中所采用的数据集规模较小，与文本检索的实际情况存在差别。最后，本文方案需要将待检索文本全部转化成哈希码的形式存储在内存中，实验中发现会存在较大的内存开销，这一点在数据量规模增加时较为明显。

针对已有工作的不足，后续可以参照如下思路进行进一步研究：

（1）构建一个针对文本检索任务的通用模型，模型参数由多个数据集联合训练得到，训练好的模型在遇到新数据时应有较好的泛化能力，且在新数据上经过少量的 Fine-tune 过程后能得到更好的结果。

(2)在使用传统 NLU 数据集的基础上，使用更多文本检索领域的真实数据进行实验，对比模型在新老数据集上的性能差异，分析原因并对模型做进一步改进。

(3)可以对模型输出哈希码的位数进行调整，在时间和空间的开销方面寻求折衷。

参 考 文 献

[1] MITRA B, CRASWELL N. An introduction to neural information retrieval[M]. Now Foundations and Trends, 2018: 1-126.

[2] ROBERTSON S, ZARAGOZA H. The probabilistic relevance framework: BM25 and beyond[M]. Now Publishers Inc, 2009: 1-32.

[3] LI H, XU J. Semantic matching in search[J]. Foundations and Trends in Information Retrieval, 2014, 7(5): 343-469.

[4] XIONG C Y, DAI Z N, JAMIE C, et al. End-to-End Neural Ad-hoc Ranking with Kernel Pooling[J]. ACM Sigir Forum, 2017, 51(cd): 55-64.

[5] HUI K, YATES A, BERBERICH K, et al. Co-PACRR: A Context-Aware Neural IR Model for Ad-hoc Retrieval[C]// Eleventh ACM International Conference. ACM, 2017.

[6] MITRA B, DIAZ F, CRASWELL N. Learning to match using local and distributed representations of text for web search [C]// Proceedings of the 26th International Conference on World Wide Web. 2017: 1291-1299.

[7] LI Z J, FAN Y, WU X J. Survey of Natural Language Processing Pre-training Techniques[J]. Computer Science, 2020, 47(3): 162-173.

[8] MIKOLOV T. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26: 3111-3119.

[9] PENNINGTON J, SOCHER R, MANNING C. GloVe: Global Vectors for Word Representation[C]// Conference on Empirical Methods in Natural Language Processing. 2014.

[10] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification[J]. arXiv: 1607.01759, 2016.

[11] RAJPURKAR P, ZHANG J, LOPYREV K, et al. Squad: 100000+ questions for machine comprehension of text[J]. arXiv: 1606.05250, 2016.

[12] LAI G, XIE Q, LIU H, et al. Race: Large-scale reading comprehension dataset from examinations[J]. arXiv: 1704.04683, 2017.

[13] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need[C]// Neural Information Processing Systems. 2017: 5998-6008.

[14] DEVLIN J, CHANG M W, LEE K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv: 1810.04805, 2018.

[15] YANG Z, DAI Z, YANG Y, et al. Xlnet: Generalized autoregressive pretraining for language understanding[J]. arXiv: 1906.08237, 2019.

[16] LIU Y, OTT M, GOYAL N, et al. Roberta: A robustly optimized bert pretraining approach[J]. arXiv: 1907.11692, 2019.

[17] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv: 1409.0473, 2014.

[18] HE K, ZHANG X, REN S, et al. Deep Residual Learning for Image Recognition[C]// Computer Vision and Pattern Recognition. 2016: 770-778.

[19] SLANEY M, CASEY M A. Locality-Sensitive Hashing for Finding Nearest Neighbors [Lecture Notes][J]. IEEE Signal Processing Magazine, 2008, 25(2): 128-131.

[20] DATAR M, IMMORLICA N, INDYK P, et al. Locality-sensitive hashing scheme based on p-stable distributions[C]// Symposium on Computational Geometry. 2004: 253-262.

[21] CAI H, LI Z J, SUN J, et al. Fast Chinese Text Search Based on LSH[J]. Computer Science, 2009, 36(8): 201-204.

[22] WEISS Y, TORRALBA A, FERGUS R. Spectral hashing[C]// Conference on Neural Information Processing Systems. 2008: 1-4.

[23] LIN K, YANG H, HSIAO J, et al. Deep learning of binary hash codes for fast image retrieval[C]// Computer Vision and Pattern Recognition. 2015: 27-35.

[24] YAO T, LONG F, MEI T, et al. Deep semantic-preserving and rank-ing-based hashing for image retrieval[C]// International Joint Conference on Artificial Intelligence. 2016: 3931-3937.

[25] LU J, LIONG V E, ZHOU J, et al. Deep Hashing for Scalable Image Search[J]. IEEE Transactions on Image Processing, 2017, 26(5): 2352-2367.

[26] ZHANG S, LI J, ZHANG B, et al. Semantic Cluster Unary Loss for Efficient Deep Hashing[J]. IEEE Transactions on Image Processing, 2019, 28(6): 2908-2920.

[27] ZENG Y, CHEN Y L, CAI X D. Deep Face Recognition Algorithm Based on Weighted Hashing[J]. Computer Science, 2019, 46(6): 277-281.

[28] GUO J, FAN Y, PANG L, et al. A deep look into neural ranking models for information retrieval[J]. Information Processing & Management, 2019, 6: 102067-102086.

[29] NOGUEIRA R, CHO K. Passage Re-ranking with BERT [J]. arXiv: 1901.04085.

[30] YAN M, LI C, WU C, et al. IDST at TREC 2019 Deep Learning Track: Deep Cascade Ranking with Generation-based Document Expansion and Pre-trained Language Modeling[C]// TREC 2019. 2019.

[31] HOFSTATTER S, ZLABINGER M, HANBURY A. Interpretable & time-budget-constrained contextualization for re-ranking [J]. arXiv: 2002.01854.

[32] HOFSTATTER S, ZAMANI H, MITRA B, et al. Local self-attention over long text for efficient document retrieval[C]// Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 2021-2024.

[33] MACAVANEY S, YATES A, COHAN A, et al. CEDR: Contextualized embeddings for document ranking[C]// Proceedings of

the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2019:1101-1104.
[34] BA J L, KIROS J R, HINTON G E. Layer normalization[J]. arXiv:1607.06450, 2016.
[35] WANG A, SINGH A, MICHAEL J, et al. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding[J]. arXiv:1804.07461, 2018.
[36] DOLAN W B, BROCKETT C. Automatically constructing a corpus of sentential paraphrases[C]// Proceedings of the Third International Workshop on Paraphrasing. 2005.
[37] CER D, DIAB M, AGIRRE E, et al. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation[J]. arXiv:1708.00055, 2017.
[38] WOLF T, CHAUMOND J, DEBUT L, et al. Transformers: State-of-the-art natural language processing[C]// Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. 2020:38-45.
[39] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation, 1997, 9(8):1735-1780.
[40] LEVESQUE H, DAVIS E, MORGENSTERN L. The winograd schema challenge[C]// Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning. 2012.

ZOU Ao, born in 1997, postgraduate. His main research interests include machine learning and natural language processing.

HAO Wen-ning, born in 1971, Ph. D, professor, Ph. D supervisor. His main research interests include big data and machine learning.