中国科学院大学
University of Chinese Academy of Sciences

信息检索课程报告

大作业实验报告

姓名　周龚、王宏光、官奕琳、胡羽昊、付柏韬

院所　计算机科学与技术学院

2023 年 12 月 31 日

1 引言

信息检索的重排任务是指在检索系统返回的初始排名结果的基础上，通过重新排序（re-ranking）来提高结果的质量。初始排名可能是通过一些传统的检索算法，如基于关键词的检索所生成的。重排任务的目标是根据用户的需求和查询的上下文，优化排名结果，使得排名更符合用户的期望，提高系统的准确性和用户满意度。

为了在不同排名模型之间实现效率和准确性的权衡，将检索分为多个阶段是一种自然而然的选择：速度快但准确性较低的模型（例如 BM25）从整个语料库中检索，而速度较慢但更准确的模型（例如 BERT）则在顶部候选列表中优化排名。

启发式检索器如 BM25（Best Matching 25）是一种用于信息检索的概率模型，旨在改进传统的 TF-IDF（Term Frequency-Inverse Document Frequency）模型。BM25 考虑了文档长度对权重的影响，引入了饱和函数和可调参数，使得它更适应不同的检索任务。该模型通过计算查询词在文档中的出现频率以及文档长度来评估文档的相关性，进而产生一个用于排序的得分。BM25 在文本搜索领域广泛应用，是许多搜索引擎和信息检索系统的基础模型之一。然而，它们在评分方面受到文档统计的限制。为了解决这个问题，深度语言模型可以用于重新估计搜索索引中的术语权重 [1, 2]，或者将可能的查询术语添加到文档中 [3]。

预训练的深度语言模型，如 BERT（Bidirectional Encoder Representations from Transformers）是一种基于 Transformer 架构的深度学习模型，由 Google 于 2018 年提出。BERT 的创新之处在于采用了双向（bidirectional）的预训练策略，通过同时考虑输入序列中左右两个方向的上下文信息，使得模型能够更好地理解词语的语境。该模型在重排名任务上展现出强大的监督迁移性能。已有的研究工作通过使用二元分类目标对 BERT 进行微调，并显示其在性能上明显优于先前的模型 [4, 5]。然而，已有文献指出，二元分类对于生成包含更难负例的高性能深度检索器来说并不是一个足够有效的方法。为了解决该问题，在本实验中，我们在训练基于 BERT 的排序模型时引入了更流行的局部对比估计（Localized Contrastive Estimation，LCE）学习。该方法能使模型更有效地区分标准答案/正样本和负样本之间的区别，并给出一个更适用于排序的关联分数，可与负样本扩展结合使用。此外，受最新研究的启发，我们也尝试采用了 docT5query 模型来扩展查询，从查询和答案的角度同时实现了数据增强。重排结果的 NDCG 指标明显优于测试集，说明了该实验的可行性。

2 实验方法

对于重排任务，简单地计算 BERT 关于查询文档对的输出和对应标签的二元交叉熵，并以此作为损失函数优化网络参数的效果往往不好。这是因为重排任务处理的是检索器查询结果靠前的部分，其中的不相关文档有较多的混淆特征难以区分，甚至会导致训练崩溃。为此，下面我们使用对比学习和数据增强两种不同角度的方法解决这个问题。

2.1 对比学习

给定一个初始的简单检索器关于一组训练查询在语料库中的查询结果，对于每个查询 q，从排名前 m 个文档的集合 R_q^m 中随机选择 n 个不相关的文档作为负样本，将这 n 个负样本与一个作为正样本的相关文档 d^+ 结合得到集合 G_q。由此我们可以为每个查询 q 定义对比损失函数：

$$\mathcal{L}_q := -\log\frac{\exp(score(q,d^+))}{\sum_{d\in G_q}\exp(score(q,d))} = -score(q,d^+) + \log\sum_{d\in G_q}\exp(score(q,d)) \tag{1}$$

其中 $score$ 函数是 BERT 对于查询文档对的得分函数。因此，我们可以为一组训练查询 Q 定义局部对比估计 (Localized Contrastive Estimation, LCE) 损失：

$$\mathcal{L}_{LCE} := \frac{1}{|Q|}\sum_{q\in Q}\sum_{d\in G_q} -\log\frac{\exp(score(q,d^+))}{\sum_{d\in G_q}\exp(score(q,d))} \tag{2}$$

2.2 数据增强

2.2.1 hard negative sample

在 2.1 节中，选择负样本的方法是随机的，为了进一步发挥对比学习的作用，尽可能排除掉混淆特征的干扰，我们有针对性地选择 hard negative sample(模型更难区分的负样本，即与正样本更接近的文本) 进行训练。具体地说，我们先随机选择 n' 个文档 (passage) 作为负样本，根据 BERT 最后一层隐藏层的输出计算各个负样本与正样本之间的余弦相似度，选取其中最相似的 n 个负样本进行训练。

2.2.2 docT5query

为了扩展文档语义相关词项，提高部分词项的权重，以及缓解词汇失配问题，我们基于 Nogueira 等提出的 docT5query 方法，使用 T5 模型 (Text-To-Text Transfer Transformer) 从给定的文档中生成与信息检索相关的查询，并将生成的查询附加到原始文档上进行索引，从而提高信息检索的效果。模型的输入是'prefix: passage' 的形式，这里我们采用的前缀是'text2query'。

2.3 训练过程

总结上面的内容，我们模型的训练过程如图 1所示。基于增强的数据使用对比学习方法训练模型，得到更新的模型后根据其输出分数对结果进行重排。

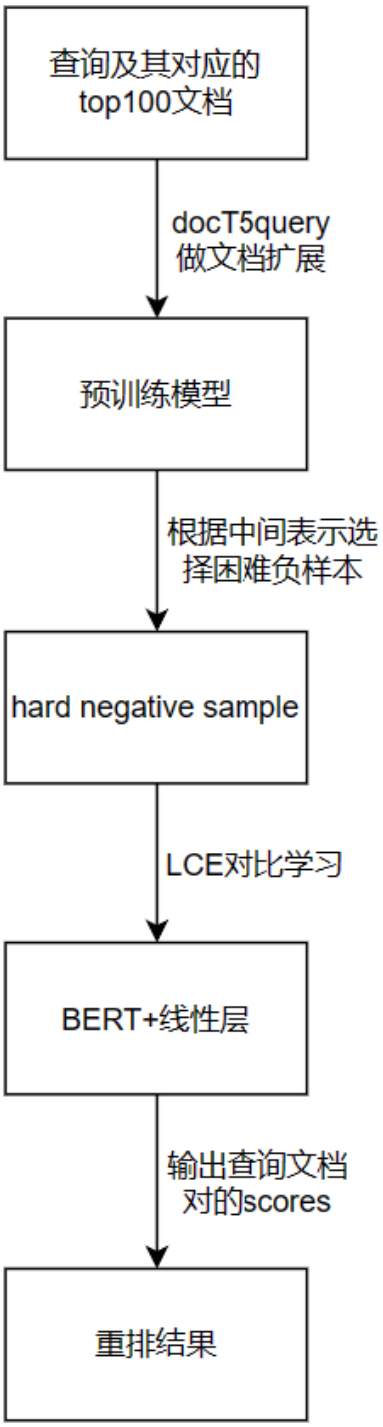
查询及其对应的
top100文档
docT5query
做文档扩展
预训练模型
根据中间表示选
择困难负样本
hard negative sample
LCE对比学习
BERT+线性层
输出查询文档
对的scores
重排结果

图 1: 模型训练过程

3 实验设计

3.1 数据集

2022 Deep Learning Passage Ranking 抽样数据集。拥有语料库、训练集、验证集、测试集的数据。

pid	passage
文档 id	文件内容

表 1: 语料库格式

其中，训练集、验证集、测试集有以下几种类型的文件:

qid	query
查询 id	查询内容

表 2: 查询格式

qid	zero	pid	rating
查询 id	0	文档 id	评分

表 3: 标准返回

qid	qzero	pid	rank	score	sys_id
查询 id	Q0	文档 id	排名	得分	系统 id

表 4: 来自简单 IR 系统的 100 篇文章初始排名

数据处理方面，首先是按照文件结构进行读入，然后将读入的数据进行处理，分词生成数据的词汇库，以及直接提取关键词生成待处理数据。

3.2 预训练模型

模型方面,我们使用了预训练的 Bert 和线性层的方法。其中初始模型使用了 Reranker 模型 [6]，对线性层结构做出了部分调整。具体来说，该模型采用了 huggingface 的 Bert 模版，并在这个基础上对预训练模型使用了 LCE 进行微调。

我们在该预训练模型的基础上，将 BERT 模型的参数冻结，并更新了线性层，训练目标是降低 LCE 损失，训练效果通过验证集和测试集上的 NDCG 指标说明。实验组的区别主要在于负样本的选择方式，一个是直接随机抽取 (1 个正样本和 4 个负样本)，一个是扩大抽取容量后再筛选 hard negative sample(随机保留 9 个负样本中与正例最相似的 4 个负样本)。具体来说，对于每个 query-passage 对，我们都获取其 BERT 最后一层

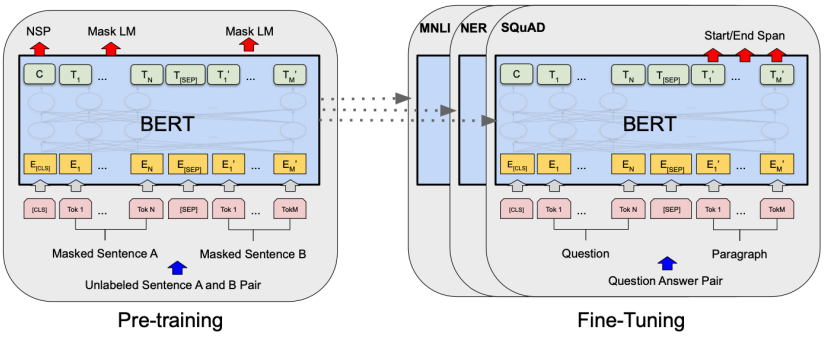

图 2: bert 模型示意

的 hidden states 作为高维表征，与正例计算余弦相似度，选取最相似的 k 个作为最终训练所用的负样本。这里 k 的取值为 4，与随机抽取个数同样是可调的超参数。

3.3 实验结果

两个模型在验证集的平均 loss 如表 5所示，尽管我们使用了低学习率，在第四个 epoch 开始 model-10 的 loss 仍开始明显增长，所以停止了训练。最终在测试集的 76 个查询上的结果如表 6所示，其中第一行是计算了 test set 的 NDCG 指标作为评价基础；第二行是 Reranker-BERT 开箱即用的结果，可以表明我们训练的真实性，以及检验实验设计是否合理，同时也能说明我们仅使用了训练集数据进行训练；第三行是使用了预训练 bert 加线性层的结果，对应的是直接抽取负样本的实验；第四行是采用增强数据训练的方法，是另一组实验。

表 5: 验证集平均 loss

Model	epoch1	epoch2	epoch3	epoch4	epoch5
model-5	**3.85**	**0.98**	**0.81**	**1.33**	**1.08**
model-10	2.34	1.37	0.80	—	—

首先，model-5 和 model-10 的重排结果相较于测试集有明显提升，说明了 BERT 模型的可用性；其次，我们的重排结果与 RerankerBERT 的结果较为接近，证明了我们确实没有在测试集上进行训练；然后，仅改变线性层结构并简单微调，重排结果得到了略微的提升，说明模型仍有优化空间；最后，hard negative sample 的方法没有额外提升，可能受实验的随机性扰动影响，不如直接从 topK 结果中抽取负样本稳定。

3.4 发布地址

- Github: https://github.com/truman0102/UCASIR2023WI

表 6: 实验结果

Model	NDCG@5	NDCG@10	NDCG@15	NDCG@20	NDCG@30	NDCG@100
test set	0.2888	0.2692	0.2561	0.2524	0.2432	0.2133
bert-base	0.4917	0.4365	0.4111	0.3912	0.3630	0.2622
model-5	**0.4938**	**0.4415**	**0.4169**	**0.3942**	**0.3646**	**0.2630**
model-10	0.4714	0.4289	0.4055	0.3826	0.3580	0.2589

4 结论

信息检索行为中的马太效应是，排名靠前的检索结果往往会受到更多的关注，而排名靠后的检索结果几乎不会被浏览。在优化前部检索结果的顺序时，深度学习方法具有重大的潜力。本实验使用 BERT 模型与对比学习的方法，实现了 TREC 竞赛的重排子任务，证明了深度学习检索方法的可用性。后续的研究可以在数据处理，模型设计和其他先验知识方向继续探索。

参考文献

[1] Zhuyun Dai and Jamie Callan. Context-aware document term weighting for ad-hoc search. In *Proceedings of The Web Conference 2020*, pages 1897–1907, 2020.

[2] Zhuyun Dai and Jamie Callan. Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 1533–1536, 2020.

[3] Rodrigo Nogueira, Wei Yang, Jimmy Lin, and Kyunghyun Cho. Document expansion by query prediction. *arXiv preprint arXiv:1904.08375*, 2019.

[4] Zhuyun Dai and Jamie Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 985–988, 2019.

[5] Rodrigo Nogueira and Kyunghyun Cho. Passage re-ranking with bert. *arXiv preprint arXiv:1901.04085*, 2019.

[6] Luyu Gao, Zhuyun Dai, and Jamie Callan. Rethink training of bert rerankers in multi-stage retrieval pipeline, 2021.