T²排序：大规模中文段落排序基准

Xiaohui Xie
xiexiaohui@mail.tsinghua.edu.cn
清华大学直流输电技术研究中心。
中关村实验室。
北京，中国

Qian Dong
dq22@mails.tsinghua.edu.cn
清华大学直流输电技术研究中心。
中关村实验室。
北京，中国

王炳宁
bryantwwang@tencent.com
腾讯公司
北京，中国

吕飞扬
feiyanglv@tencent.com
腾讯公司
北京，中国

Ting Yao
tessieyao@tencent.com
腾讯公司
北京，中国

甘维南
carrygan@tencent.com
腾讯公司
北京，中国

Zhijing Wu
wuzhijing.joyce@gmail.com
北京理工大学
北京，中国

Xiangsheng Li
lixsh6@gmail.com
腾讯公司
北京，中国

李海涛
liht22@mails.tsinghua.edu.cn
清华大学 DCST。
中关村实验室。
中国北京

Yiqun Liu
yiqunliu@tsinghua.edu.cn 清华大学 DCST。中关村实验室。中国北京

Jin Ma
daniellwang@tencent.com 腾讯公司。中国北京

arXiv:2304.03679v1 [cs.IR] 7 Apr 2023

摘要

段落排序涉及两个阶段：段落检索和段落重排序，这两个阶段对于信息检索（IR）领域的学术界和产业界而言，都是重要且具有挑战性的课题。然而，常用的段落排序数据集通常聚焦于英语。对于中文等非英语场景，现有数据集在数据规模、细粒度相关性标注和假阴性问题方面存在局限。为解决这一问题，我们推出了T²排序，这是一个大规模中文段落排序基准。T²排序包含来自真实世界搜索引擎的超过300K个查询和超过2M个唯一段落。我们招募了专家标注员，为查询-段落对提供4级分级相关性分数（细粒度），而不是二元相关性判定（粗粒度）。为缓解假阴性问题，在进行相关性标注时，我们考虑了更多样的段落，尤其是在测试集中，以确保更准确的评估。除文本查询和段落数据外，我们还提供了其他辅助资源，例如查询类型和文档的XML文件

允许为个人或课堂用途免费制作本作品全部或部分内容的数字副本或纸质副本，但不得为营利或商业目的制作或传播副本，且副本必须附有本声明及首页上的完整引文。本作品中由 ACM 以外的其他方拥有版权的部分，必须尊重其版权。注明出处后可摘录。若要以其他方式复制、再版、发布到服务器或分发给列表，须事先获得具体许可和/或支付费用。请通过 permissions@acm.org 申请许可。

*SIGIR '23，2023*年7月*23*日至*27*日，中国台湾台北 © 2023 Association for Computing Machinery。ACM ISBN 978-1-4503-XXXX-X/18/06...$15.00
https://doi.org/XXXXXXX.XXXXXXX

这些段落的来源，以便促进后续研究。为评估该数据集，我们实现了常用的排序模型，并在 T²Ranking 上进行测试，将其作为基线。实验结果表明，T²Ranking 具有挑战性，仍有改进空间。完整数据 [1] 及所有代码均可在 https://github.com/THUIR/T2Ranking/ 获取

关键词

测试集、段落检索、段落重排序、段落排序、搜索评估

ACM参考文献格式：

Xiaohui Xie、Qian Dong、Bingning Wang、Feiyang Lv、Ting Yao、Weinan Gan、Zhijing Wu、Xiangsheng Li、Haitao Li、Yiqun Liu 和 Jin Ma。2023 年。**T²段落排序**：大规模中文段落排序基准。载于 第 *46* 届 *ACM SIGIR* 国际信息检索研究与开发会议论文集（*SIGIR '23*）。ACM，美国纽约州纽约市，10 页。https://doi.org/XXXXXXX.XXXXXXX

1 引言

段落排序是信息检索系统的关键组成部分。段落排序的良好表现能够提升搜索用户的满意度，并使多个与信息检索相关的应用- 程序受益，例如问答 [1] 和阅读理解 [17]。通常，段落排序包含两个连贯的阶段，即段落检索和段落重排序。段落排-序的目标是编制一个按照相关性排序的搜索结果列表

[1]该数据集根据 Apache 2.0 许可证授权

针对来自大型段落集合的查询。第一阶段，即段落检索，需要从庞大的段落语料库中召回相关段落。因此，除了有效性之外，还应考虑效率 [10]。第二阶段，即段落重排序，可以采用更加注重有效性的模型，对第一阶段检索到的段落进行重排序。

为支持段落排序研究，已经构建了各种基准数据集。其中一些同时支持第一阶段检索（FR）和第二阶段重排序（SR）任务，而另一些则专注于SR任务。我们在表1 中总结了部分常见数据集的数据统计。常用数据集主要面向英语场景。例如，TREC 复杂答案检索（CAR） [6], TriviaQA [11] 和 MS-MARCO [16]。其中，MS-MARCO 是一个拥有 880 万个段落的大规模数据集。查询以问题为基础，并由标注员提供人工生成的答案。随后，可以通过判断段落中是否存在与查询相关的答案来获得二元相关性得分；也就是说，包含答案的段落为相关（1），不包含答案的段落为不相关（0）。继 MS-MARCO成功之后，非英语社区也构建了类似的数据集，例如中文数据集。例如，mmarco-Chinese [3] 借助机器翻译实现了原始 MS-MARCO 的中文版本。DuReader检索 [20]采用了类似范式，根据人工生成的答案为查询-段落对生成二元相关性判断。Multi-CPR [15] 是一个用于段落检索的多领域中文数据集，包含三个不同领域以及一定数量的人工标注查询-段落对。此外，Sogou-SRR [29],Sogou-QCL [30] 和 TianGong-PDR [27] 基于来自 Sogou2（一款热门中文搜索引擎）的用户日志提供。

尽管现有数据集促进了段落排序应用的发展，但仍存在若干需要解决的局限性：

(1) 这些数据集既不是大规模的，也不是人工生成的，尤其是在中文社区中。Sogou-SRR 和 Tiangong- PDR 涉及的查询数量有限。尽管mMarco-Chinese 和 Sogou-QCL 规模较大，但前者基于翻译，而后者仅采用点击标签。最近，构建了两个数据规模可观的段落排序数据集，即 DuReader检索 和 Multi-CPR。

（2）细粒度人工标注有限。大多数数据集采用二元相关性标注。Roitero 等人 [24] 证明了细粒度相关性等级的优势，因此近期研究也开始探索超越二元（粗粒度）范式的细粒度相关性标注。然而，细粒度标注的数量相当有限，例如，Sogou-SRR 和 TianGong-PDR 中均不到 10 万条。

（3）假阴性问题会损害评估的准确性。正如 Arabzadeh 等人 [2] 指出的，现有的段落排序数据集存在假阴性问题，即相关结果被标记为不相关。这个问题主要是由于大规模数据集中的人工标注有限所导致的，会损害评估的准确性。例如，对于 Multi-CPR 中的每个查询，只有一个段落会被标记为正例，而其他段落则被视为负例。近期的一个数据集 DuReader检索试图通过要求标注员手动检查并重新标注从多个检索器汇总的排名靠前的检索结果中的段落，来缓解这一问题。

为了确保段落排序模型的训练和评估具有高质量，我们构建并发布了一个新的中文数据集，名为 T^{2}Ranking，其中包含超过 307K 个基于问题的查询，以及从 180 万篇网络文档中提取的 230 多万个段落。具体而言，我们从中国流行搜索系统搜狗搜索引擎的用户日志中采样查询，并进行查询预处理，例如过滤色情查询和非疑问式查询、移除相似查询，从而获得包含 307K 个查询（测试集 50K 个）的干净、高质量查询集。对于每个查询，我们从不同搜索引擎中提取对应文档的内容，并移除垂直搜索结果（例如图片搜索结果和视频搜索结果）及重复结果，以供后续处理。为确保每个段落的语义完整性，我们训练并使用段落切分模型从每篇文档中提取段落，平均每篇文档得到约 1.3 个段落。随后，我们使用段落聚类方法剔除高度相似的段落，并生成查询-段落池。此外，我们还记录了查询类型和文档的其他辅助资源，以便开展拓展研究（例如多模态任务和域外（OOD）任务）。对于给定的查询及其对应段落，我们聘请专家标注员为每个查询-段落对提供 4 级相关性判定，并采用基于主动学习的数据采样，提高标注的效率和质量。所有受聘标注员均为从事标注工作的全职员工。

我们对所提出的数据集进行了全面分析，并给出了全面的统计数据。此外，我们在 T^{2}Ranking 上开展了全面实验，以评估多个段落检索模型以及段落重排序模型的性能。实验结果表明，T^{2}Ranking 是一项极具挑战性的任务，性能仍有进一步提升的潜力。

总而言之，我们作出如下贡献：

- 我们构建了一个名为 T^{2}Ranking 的大规模中文数据集，用于段落排序（检索和重排序）。T^{2}Ranking 包含超过 30 万条查询和超过 200 万个独特段落，还提供细粒度相关性标注，以及作为多模态信息的查询类型、文档标题和 XML文件。
- 我们采用多种策略来确保数据集的高质量，例如使用段落切分模型和段落聚类模型来增强段落的语义完整性和多样性，并采用主动学习标注方法来提高数据标注的效率和质量。
- 我们在T^{2}Ranking上开展了大量实验，以评估现有段落检索和重排序模型的性能。实验结果表明，该任务仍有进一步提升的空间，未来更复杂的模型或许能够带来改进。

[2]https://www.sogou.com/

表1：段落排序中常用数据集的数据统计。Qrys（Psgs）：查询（段落）。FR（SR）：段落排序的第一阶段（第二阶段），即段落检索（重排序）。

数据集	Lang	#查询	#段落	查询来源	段落来源	标注	Task
Trec Car [6]	EN	2M	30M	Wiki 文档。	Wiki 文档。	二元	SR
TriviaQA [11]	EN	95K	650K	Trivia 问答。	维基百科/网页文档。	二元	法语、SR
MS-MARCO [16]	EN	516K	8.8M	用户日志	网页文档。	二元	法语，SR
Sogou-SRR [29]	CN	6K	63K	用户日志	网页文档。	细粒度	SR
搜狗-QCL [30]	CN	537K	9M	用户日志	网页文档。	点击标签	SR
天工-PDR [27]	CN	70	11K	用户日志	新闻文档	细粒度	法语，SR
mMarco-中文 [3]	CN	516K	8.8M	用户日志	网页文档。	二元	法语，SR
Multi-CPR [15]	CN	303K	3M	用户日志	结果标题	二元	法语，SR
DuReaderretrieval [20]	CN	97K	8.9M	用户日志	网页文档。	二元	法语，SR
T²Ranking（我们的）	CN	307K	2.3M	用户日志	网页文档。	细粒度	法语、SR

2 相关工作

目前已有多个用于段落排序的基准数据集。对于那些为所有查询—段落对提供相关性标注的数据集，可以同时测试段落检索和段落重排序任务。然而，其他数据集仅关注段落重排序任务，只为查询—段落对提供相关性标注；这些段落已由第一阶段检索器从初始结果列表中提取出来。如表 1 所示，我们用 FR 表示段落排序的第一阶段，即段落检索；用 SR 表示段落排序的第二阶段，即段落重排序。

常用的段落排序数据集是为英语社区构建的。Trec 复杂答案检索（CAR） [6] 使用从维基百科中提取的主题、提纲和段落。对于训练集，如果某个段落出现在主题对应的维基百科页面中，则视为相关，否则视为不相关。测试集包含 113 个复杂主题，每个主题有 50 个经人工标注的段落。TriviaQA [11] 从 14 个问答和知识竞赛网站收集问答对，并从维基百科和网页文档中收集段落。MS-MARCO [16] 因其规模庞大而得到广泛应用。与 Trec Car 和 TriviaQA 不同，MS-MARCO 中的查询来自必应搜索引擎[3]的用户生成查询，且以问题为基础。段落取自同一搜索引擎返回的真实网页文档。随后，研究人员招募人工编辑，并要求他们针对特定查询，仅根据所提供的段落提取正确的信息，撰写自然语言答案。TriviaQA 和 MS-MARCO 中段落的相关性均以二元方式判定，依据是段落是否包含给定查询的真实答案要点。

对于中文社区而言，已有多个用于训练和评估段落排序模型的数据集。基于搜狗搜索引擎，已建立了三个数据集，分别是 Sogou-SRR [29], Sogou-QCL [30] 和 TianGong-PDR [27]。Sogou-SRR（搜索结果相关性）包含6K 个查询及其对应的前 10 条搜索结果。对于每条搜索结果，均提供截图、标题、摘要、HTML 源代码、解析树、URL，以及四级相关性评分和结果类型。Sogou-QCL 是一个大规模数据集，由 537K 个查询和超过 900 万个中文网页组成。与人工生成的相关性判定不同，查询-结果对的相关性等级是基于点击标签进行评估的。Tiangong-PDR 中的查询收集自搜狗的搜索日志，而段落则来自新浪新闻网站的网页数据[4]。此外，每个查询-段落对都有四级人工评估的相关性标签。除此之外，mMarco-Chinese [3] 是通过对 MS-MARCO进行机器翻译构建的。然而，这些数据集并非大规模和/或人工生成的。最近，Qiu 等人 [20]提出了一个名为DuReader检索的新数据集，用于基准测试百度搜索[5]中的段落检索模型。与MS-MARCO 类似，DuReader检索中的查询基于问题，并收集人工生成的答案来评估段落的相关性等级。Long等人 [15] 构建了 Multi-CPR，这是一个用于段落排序的多领域数据集。Multi-CPR 的查询和段落来自三个不同的垂直搜索系统：电子商务、娱乐视频和医疗。Multi-CPR 中的段落并非从网页文档中提取，而是指搜索结果的标题，例如电子商务搜索中的商品标题，因此段落长度更短。研究人员招募了人工标注员来判断查询-段落对的相关性等级（二元）。对于每个查询，语义上最相关的段落被标记为正例，而其他段落则被标记为负例。

3 任务定义

本节将正式定义 T²Ranking 中的任务。我们提出的数据集聚焦于段落排序的两个阶段，即段落检索和重排序。这与现代信息检索系统的处理流程一致，该流程遵循先检索、后重排序的范式。

段落检索的目标是响应给定的查询，检索候选段落。给定一个查询 q，使用检索模型从大型语料库 $\mathbf{G} = \{\mathbf{p}_i\}_{i=1}^{G}$中检索候选段落集 $\mathbf{K} = \{\mathbf{p_q}_j\}_{j=1}^{K}$，其中 $K \ll G$。具体而言，一个段落由词序列 $\mathbf{p} = \{w_p\}_{p=1}^{|\mathbf{p}|}$组成，其中 $|\mathbf{p}|$ 表示其长度

[3]https://www.bing.com

[4]https://www.sina.com.cn/ [5] https://www.baidu.com/

为段落 $\mathbf{p}$ 的长度。同样，查询是一个词序列 $\mathbf{q} = \{w_q\}_{q=1}^{|\mathbf{q}|}$。段落检索的主要挑战在于，面对语料库中数量庞大的段落，如何高效地为查询检索相关段落。在检索之后，通常采用重排序，在 K 上生成一个排列，使相关性更高的段落在列表中的排名更靠前。与检索任务相比，重排序任务要求模型具备更强的相关性建模能力，以捕捉候选集 K 中相关段落之间细微的语义差异。

4 数据集构建

本节介绍 T²Ranking的构建细节。我们首先介绍数据集构建的整体流程，其中包括查询采样、段落提取和相关性标注。随后，我们提供数据构建中使用的重要技术细节，例如基于模型的段落分割、基于聚类的段落去重，以及基于主动学习的数据采样。

4.1 总体流程

构建 T²Ranking 的整体流程包含多个步骤，包括查询采样、文档检索、段落抽取和细粒度相关性标注。

查询采样。 我们从搜狗的查询池中采样真实用户查询，并进行预处理（例如去重，以及规范化多余的空格和问号），以获得干净的查询数据集。然后，我们使用意图分析算法，从 T² Ranking 中过滤掉色情查询、非疑问型查询、资源请求型查询，以及可能包含用户信息的查询，以确保数据集仅由高质量的、基于问题的查询组成。请注意，资源请求型查询用于搜索特定的音乐、电影资源等。

文档检索。我们从搜狗、百度和 Google 等热门搜索引擎中为每个查询检索一组全面的文档，充分利用它们在网页内容索引和排名方面的丰富资源与专业能力。这有助于减少假阴性问题，因为每个系统覆盖的网络部分不同，并且能够返回不同的相关文档，从而提高我们数据集的整体覆盖率。

段落提取。T²Ranking中段落的构建涉及分段和去重。我们没有采用启发式方法从给定文档中划分段落（例如，通常通过换行符确定段落的起始或结束位置），而是采用基于模型的方法进行段落分段，以最大限度地保留每个段落中的完整语义（详见第 4.2 节）。此外，我们引入了一种基于聚类的技术，以提高标注效率，并保持已标注查询-段落对的多样性（详见第4.3节）。该方法能够有效去除由特定查询检索到的几乎相同的段落。经过分段和去重的段落随后被合并到 T²Ranking 的段落集合中。

细粒度相关性标注。 所有受聘的标注员都是提供搜索相关任务标注的专家，并且长期从事标注工作。每个查询-段落至少由三名标注员提供4级细粒度标注对。具体来说，如果某个查询-段落对的前三位标注员之间的标注不一致（三位标注员给出了三个不同的分数），将请第四位标注员对其进行评估。如果四位标注员的标注仍不一致，则认为该查询-段落对过于含糊，无法确定所需信息，并将其从数据集中排除。每个查询-段落对的最终相关性标签由多数投票决定。遵循 TREC基准测试集的标准 [5], 我们还将四级相关性标注的说明定义如下：

- **Level-0.** 查询与段落的内容完全不匹配。
- **第 1 级**。 该段落与查询相关，但未能满足该查询所需的信息需求。
- **二级**。 该段落与查询相关，且部分满足其信息需求。
- **第 3 级**。 段落内容经过定制，以满足查询的信息需求，并准确包含查询的答案。

我们在表 2 中展示了几个示例。细粒度的 4 级标注能够准确评估段落重排序任务。值得注意的是，对于检索任务，我们将**第 2 级**和 **第 3 级**段落视为相关段落，其他所有段落均视为不相关段落。

值得注意的是，在处理测试查询时，我们采用了在段落分割过程之后标注所有查询-段落对的策略，该策略旨在缓解测试集中的假阴性问题，从而对检索和重排序性能提供更精确的评估。对于训练查询，我们采用上述基于聚类的方法对段落进行去重，然后将去重后的段落呈现给招募的专家标注员，以获得四级细粒度标注。这一策略不仅提高了标注效率，还保持了所标注查询-段落对的多样性。此外，主动学习策略的成功促使我们通过选择具有信息量的查询-段落对进行标注，来丰富训练样本中包含的信息。主动学习背后的核心思想是，允许模型选择其希望学习的训练样本，并专注于那些对提升其性能最有价值的样本，从而实现更高效的标注。在 T²Ranking中，我们设计了一种基于主动学习的方法，以迭代方式对训练数据进行标注（详见第 4.4节）。总体而言，T²Ranking 的数据构建流程在算法 1 中进行了形式化定义。

4.2 基于模型的段落分割

通常，现有数据集中的段落是根据自然段或固定长度的滑动窗口从文档中切分出来的。然而，考虑到大多数网页文档的行文并不严谨，基于自然段的分割通常会产生过长的段落，其中包含多个主题。此外，基于滑动窗口的分割往往会导致段落语义不完整 [4, 20], , 从而降低数据集用于评估段落检索和重排序的可靠性。

为了解决这一问题，我们提出一种用于段落分割的基于模型的方法。我们使用精心撰写的文本训练分割模型

表2：查询-段落对标注示例。

#标注	查询	段落	解释
0	焚烧茶叶和菊- themums 会产生茶多酚吗?	茶和菊花可以被允许用于 混合浸泡。菊花 其本身具有.....的效果。	文中没有提到查询。 段落，并且它不符合信息- 根本不需要查询的信息需求。
1	宫颈癌 IIB 期是什么意思 均值?	IB 期宫颈癌意味着 癌症局限于宫颈，并且 阴道镜检查显示病变范围更大.....	此段落涉及宫颈癌- cer，但它不适用于给定的 查询，用于请求有关.....的信息 IIB期。
2	儿童出汗过多的原因是什么- 德伦?	儿童在床上出汗是由于夜晚 儿童出汗.....	查询并不局限于.....的主题 仅在睡眠期间出汗。
3	鲢鱼喜欢什么口味的饵料?	钓雅罗鱼的最佳诱饵是饵料，里面含有 芳香、鱼腥或难闻的气味以及 新鲜.....	答案恰好包含在 该段落。

算法 1： 数据集构建流程

输入： 查询池 Q，文档池 D，段落分割模型 $Seg(\cdot)$、交叉编码器 $CE(\cdot)$ 和专家标注员 H

输出： 细粒度相关性标签 L

```
数据集构建(Q, D,H) 开始
    Q = Sample(Q); % 对查询集合进行采样Q;
    L = ∅,P = ∅; % 初始化标签集和段落集;
    for q ∈ Q do
        %通过以下方式为查询 q 检索一组文档D
          多个搜索引擎;
        D = MultiSearchEngines(q,D);
        对于 d ∈ D，执行。
            P ∪ Seg(d);% 段落分割;
        end
        如果q是一个训练查询那么
            % 基于聚类的段落去重用于
              训练查询;
            P =去重(P);
            如果 CE(·) 已准备就绪则
                % 筛除确定性较高的查询-段落对
                  ;
                P = 过滤(CE(q, P));
            end
        end
        % 测试查询的段落均已标注，以
          缓解假阴性问题;
        对 p ∈ P 执行
            % 查询-段落对的标注;
            L ∪ H (q, p);
        end
    end
    返回 L
end
```

使用序列标注任务处理网页文档。具体而言，我们我们使用搜狗百科、百度百科[7] 和中文维基百科[8] 作为训练数据，因为这些网页文档通常撰写得较为规范，且自然段落划分清晰。图 1 展示了一个英文版维基百科示例。给定一篇网页文档 $\mathbf{d} = \{w_d\}_{d=1}^{|\mathbf{d}|}$，我们使用分割模型 $Seg(\cdot)$ 来判断给定词语 w_d 是否应当分开。形式上，序列标注任务可定义为

$$\hat{y}_d = Seg(\mathbf{d})_{w_d}, \tag{1}$$

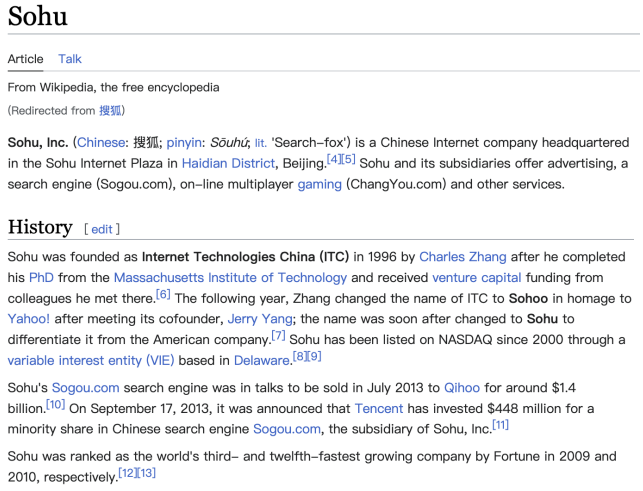
$$\mathbf{L}_s = \text{CrossEntropy}(y_d, \hat{y}_d), \tag{2}$$

其中，$\hat{y}_d$ 是分割的预测分数。真实标签 y_d 表示单词 w_d 是否为段落的最后一个单词。分割模型 $Seg(\cdot)$ 基于公式 2 中定义的损失进行训练。如果 $\hat{y}_d \geq \sigma$，则从第 d 个单词处将段落从其文档中分割出来。σ 是控制分割程度的超参数。σ 的值越小，分割出的段落就越多。

4.3 基于聚类的段落去重

在网上标注大量高度相似的段落既多余又毫无意义。本文提出一种基于聚类的段落去重方法，从而提高标注效率。具体而言，我们采用层次聚类算法 Ward [26]，以无监督的方式将相似段落聚集在一起。同一簇中的段落被视为近乎重复。因此，我们只从每个簇中选取一个段落进行标注。值得注意的是，我们只在训练集上进行去重。对于测试集中的查询，为了尽可能缓解假阴性问题，我们会标注由段落分割模型得到的所有段落。直观来说，对于排序模型而言，在特定查询下几乎完全相同的段落，与差异显著的段落相比，提供的信息增益很少。在实际操作中，与已标注为真阳性的段落属于同一簇的假阴性，可以通过跨编码器 [21]轻松过滤。因此，我们采用

[6]https://baike.sogou.com/[7] https://baike.baidu.com/ [8] https://zh.wikipedia.org/



Sohu

Article Talk

From Wikipedia, the free encyclopedia

(Redirected from 搜狐)

Sohu, Inc. (Chinese: 搜狐; pinyin: *Sōuhú*; lit. 'Search–fox') is a Chinese Internet company headquartered in the Sohu Internet Plaza in Haidian District, Beijing.[4][5] Sohu and its subsidiaries offer advertising, a search engine (Sogou.com), on–line multiplayer gaming (ChangYou.com) and other services.

History [edit]

Sohu was founded as **Internet Technologies China (ITC)** in 1996 by Charles Zhang after he completed his PhD from the Massachusetts Institute of Technology and received venture capital funding from colleagues he met there.[6] The following year, Zhang changed the name of ITC to **Sohoo** in homage to Yahoo! after meeting its cofounder, Jerry Yang; the name was soon after changed to **Sohu** to differentiate it from the American company.[7] Sohu has been listed on NASDAQ since 2000 through a variable interest entity (VIE) based in Delaware.[8][9]

Sohu's Sogou.com search engine was in talks to be sold in July 2013 to Qihoo for around $1.4 billion.[10] On September 17, 2013, it was announced that Tencent has invested $448 million for a minority share in Chinese search engine Sogou.com, the subsidiary of Sohu, Inc.[11]

Sohu was ranked as the world's third– and twelfth–fastest growing company by Fortune in 2009 and 2010, respectively.[12][13]

图 1：维基百科中一篇行文流畅、段落划分清晰的网页文档示例。

通过去重节省标注成本，同时保留更多样化的训练样本，以提升模型性能。

4.4 基于主动学习的数据采样

在实践中，我们发现，并非所有训练样本都能进一步提升排序模型的性能。模型能够轻松准确预测的训练样本，不太可能为模型训练提供有用信息。

为了解决这一问题，我们借鉴主动学习的思路 [22]，使用模型挑选信息量更大的训练样本，以便进一步进行标注。主动学习是一种让模型参与数据标注过程的框架。主动学习旨在尽可能减少所需的已标注数据量，同时维持或提升模型性能。形式上，主动学习是一个迭代过程：模型对一批未标注样本进行预测，标注员选取其中不确定性最高或信息量最大的样本进行标注，再将已标注样本加入训练数据。随后，使用新标注的数据更新模型。主动学习框架如图 2 所示。具体而言，首先使用初始阶段构建的数据训练查询-段落重排序模型，具体来说是交叉编码器。在第二阶段，获取未标注的查询-段落对，并由交叉编码器评估其相关性。置信度分数较高的样本对会被筛除，因为它们无法为进一步提升性能提供显著信息；而置信度分数较低的样本对通常被视为噪声样本，也会被剔除。剩余的样本对将提交给标注员进行细粒度标注。随后，将已标注的查询-段落对加入训练集，并使用新获取的样本更新交叉编码器。

5 数据统计

本节介绍 T^{2}Ranking 的数据统计。

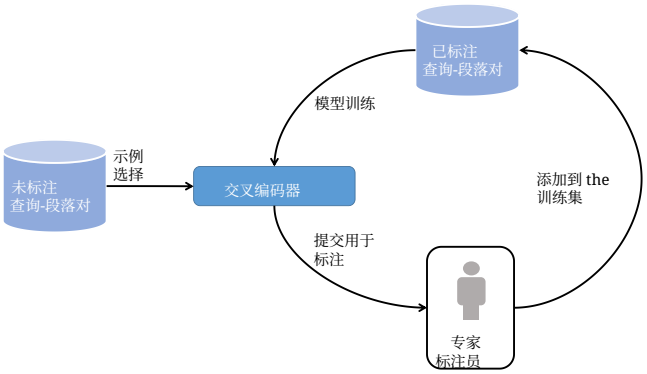


图 2：主动学习框架示意图。

表 3：T^{2}Ranking中的查询统计。

	数量	最大长度	平均长度
训练集	258,042	40	11.1
测试集	49,662	38	10.99

查询。 表 3 总结了 T^{2}Ranking 中查询的统计信息。训练集和测试集中查询的最大长度和平均长度几乎相同。我们进一步分析了训练集和测试集中查询的领域分布，如图 3 所示。领域标签由搜狗搜索引擎提供。训练集和测试集中的查询领域分布一致，并且查询涵盖了广泛的领域。我们还通过采用度量指标列表内相似度（ILS）[31] 来展示查询的多样性程度，该指标可定义为

$$s(\mathbf{q}_i, \mathbf{q}_j) = \frac{\text{BERT}(\mathbf{q}_i)_{[cls]} \cdot \text{BERT}(\mathbf{q}_j)_{[cls]}}{\|\text{BERT}(\mathbf{q}_i)_{[cls]}\| \|\text{BERT}(\mathbf{q}_j)_{[cls]}\|}, \tag{3}$$

$$\text{ILS}_Q = \frac{\sum_{i=1}^{|Q|} \sum_{j=i+1}^{Q} s(\mathbf{q}_i, \mathbf{q}_j)}{\sum_{i=1}^{Q} \sum_{j=i+1}^{Q} 1}, \tag{4}$$

其中，BERT [13] 是一种预训练语言模型，通常被用作各种任务 [7–9, 12, 18]的骨干模型。ILS分数越低，表示基准中的查询之间的相似度越低，也就意味着多样性越高。我们计算了T^2 Ranking 的 ILS分数，以及多个热门数据集（如 MSMARCO、Multi-CPR 和DuReader检索）的ILS 分数。结果如表 4 所示。从表中可以明显看出，T^{2}Ranking中的查询更加多样，这体现为更低的 ILS 分数。需要注意的是，T^2 Ranking 包含的查询甚至比 Multi-CPR中的查询具有更高的多样性，而 Multi-CPR 包含来自不同垂直搜索应用的查询。

文档 & 段落。 T^{2}排名包含从1,752,482份网页文档中提取的段落，分段后总计2,303,643个段落。平均而言，每份网页文档被划分为1.31个段落，平均长度为632.6。

相关性标注。 我们在图 4 中展示了 4 级相关性标注的分布。在训练集中，平均每个查询标注了 6.25 个段落；而在测试集中，平均每个查询标注了 15.75 个段落。

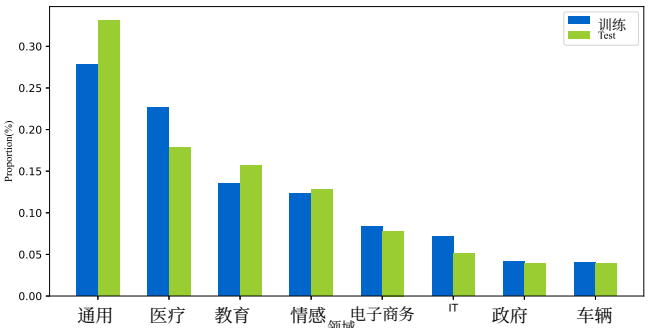
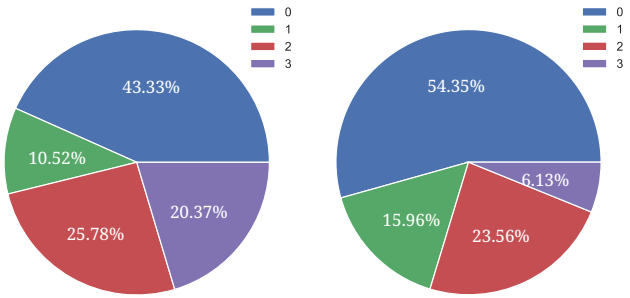


图3：T2Ranking 中训练和测试查询的领域统计信息。

表 4：不同数据集的 ILS 得分。ILS 得分越低，查询的多样性越高。

数据集	ILS 分数
MS-MARCO [16]	0.227
Multi-CPR [15]	0.186
DuReaderretrieval [20]	0.152
T²Ranking（我们的）	**0.144**



(a) 相关性的分布训练集中的标注。 **（b）相关性测试集中的标注。**

图 4：标注分布饼图。

6 实验与结果

与现代信息检索系统一致，我们的实验采用了检索-然后重排序范式。本节将考察常用检索器和重排序器在 T²Ranking 上的性能。

6.1 检索性能

基线。 现有的检索模型大致可分为稀疏检索模型和密集检索模型。稀疏检索模型侧重于利用精确匹配信号设计相关性评分函数，其中 BM25 凭借出色的性能成为最具代表性且应用最广泛的基线。此外，密集检索模型利用深度神经网络，为查询和文档学习低维密集嵌入。

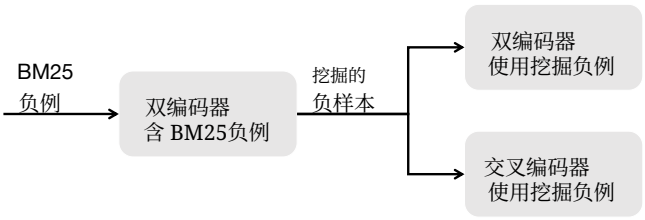
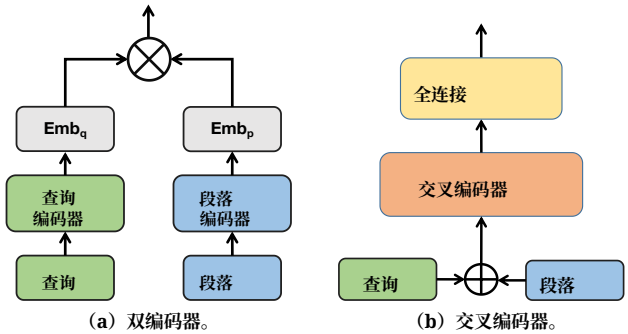


图5：我们实验中所使用基线的训练过程示意图。首先，我们使用 BM25负例训练双编码器，这与 DPR [12]类似。其次，我们使用多项研究 [15, 20]提出的全局负例采样策略，训练双编码器和交叉编码器。



（a）双编码器。 **（b）交叉编码器。**

图6：双编码器和交叉编码器架构示意图。

通常，大多数现有的稠密检索方法都遵循级联训练范式 [15, 20, 21]。因此，为便于未来研究基于我们数据集进行比较，我们按照图5所示的方式简化训练过程，如 [15, 20] 所示。具体而言，我们采用双编码器（DE）作为稠密检索模型的架构，其示意图见图6（a）。以下方法被用作我们的基线，以评估 T²Ranking 上的检索性能。

- **QL**（查询似然） [19] 是一种具有代表性的统计语言模型，通过对查询的生成过程进行建模来衡量段落的相关性。• **BM25** [23] 是一种广泛使用的稀疏检索基线。
- **DE w/ BM25 Neg** 等同于 DPR [12], ，后者是首个将预训练语言模型用作段落检索任务骨干网络的工作。
- **DEw/ 挖掘负例** 通过从整个语料库中全局采样困难负例来提升 DPR 的性能，如ANCE [28] 和 RocketQA [21] 所采用的方法。
- **DPTDR** [25] 是首个采用提示调优进行密集检索的工作。

其中，QL 和 BM25 属于稀疏检索模型，而其他模型属于稠密检索模型

实现细节。 BM25 使用 Pyserini 实现， [14] 采用默认参数。双编码器模型由.....实现。

表 5：检索模型在T²排名测试集上的性能。

	MRR@10	召回率 @50	召回率@1K
QL	.2803	。3915	.6858
BM25	.3579	。4918	.7426
带 BM25负例 的 DE	.4877	.7123	。9104
使用挖掘负例的 DE	。5191	.7357	.9147
DPTDR	**.5285**	**.7423**	**.9211**

通过深度学习框架 PyTorch，在最多8 张NVIDIA TeslaA100 GPU（配备 80G显存）上进行训练。我们使用现成的中文BERTbase 初始化双编码器。查询和段落的最大长度分别设置为32和 256。负例从BM25或DEw/ BM25负例召回的排名前200个段落中采样。正例:负例的比例设置为 1:1。我们使用 3e-5 的学习率训练双编码器 100个训练轮次。**指标**。在T²Ranking上，我们在实验中使用以下评估指标来考察基线模型的检索性能：(1) 排名前 10的检索段落的平均倒数排名（MRR@10），(2) 排名前-K的检索段落的召回率（Recall@K）。值得注意的是，对于检索任务，我们将二**级** 和三**级**段落视为相关段落，而所有其他段落均视为不相关段落。为进行全面比较，我们在测试查询上报告 Recall@50 和 Recall@1K。遵循 MS-MARCO 和 DuReader检索的评估设置，MRR 定义为一组查询中第一个相关段落倒数排名的平均值。MRR的取值范围为 0 到 1，数值越高表示系统越擅长将最相关的段落排在列表的更高位置。同时，Recall定义为在所有相关段落中被检索到的相关段落所占的比例，其取值范围同样为 0 到 1；数值越高表示系统越擅长检索出所有相关段落。MRR和 Recall衡量的是检索性能的不同方面。MRR@K 和 Recall@K 可表示为:

$$MRR@K = \frac{1}{|Q|} \sum_{q \in Q} \frac{\mathbf{I}(rank \leq K)}{rank}, \tag{5}$$

$$Recall@K = \frac{\mathbf{I}(rank_p^{\mathcal{K}^q} \leq K)}{\sum_{q \in Q} \sum_{p \in R_q} 1}. \tag{6}$$

其中，$\mathbf{I}(\cdot)$是指示函数。式 5 中的 $rank$ 表示查询 q的检索候选项中第一个相关段落的位置。 R_q 和 $rank_{\mathcal{K}_p}{}^q$ 分别表示查询 q 的相关段落，以及段落 p 在候选列表 K^q 中的位置。

检索性能。我们在表 5 中报告了基线模型的检索性能。与传统的稀疏检索方法 BM25 相比，双编码器模型显著提升了我们数据集上的检索性能。这一改进可归因于两种不同知识来源的融合，即在海量语料库上对语言模型进行无监督预训练所获得的潜在知识，以及通过在我们的大规模已标注数据集上进行监督训练所获取的相关性知识。结合近期研究中提出的负例挖掘策略 [28],，双编码器模型的检索性能可以在 T²Ranking 上进一步提升。值得注意的是，T²Ranking 中观察到的 Recall@K 指标低于其他采用粗粒度标注的基准测试集所报告的结果。例如，BM25 在 MS-MARCO- DEV Passage 和 DuReader检索上的 Recall@50 分别为 .601 和 .700，而在我们的数据集上为 0.4918。在 T²Ranking 的测试集中，我们有更多段落采用细粒度相关性标签进行标注，因此每个查询平均有 4.74 个正段落，这使检索任务更加困难，并在一定程度上缓解了假阴性问题。这凸显了 T²Ranking 所面临的挑战，以及未来进一步改进的潜力。

6.2 重排序性能

基线。 由于重排序器处理的段落数量较少，因此通常采用交叉编码器架构，而非双编码器架构。交叉编码器方法能够实现查询与文档之间更细致的交互，从而提升性能，但效率会有所降低。我们报告了交叉编码器模型的重排序性能。该模型使用从整个语料库中挖掘出的困难负例进行训练，如图5所示。交叉编码器架构如图6(b)所示。

实现细节。 交叉编码器与双编码器在相同的实验环境中实现，最大输入长度为288。负例从双编码器检索出的排名前256的段落中采样，并将正例与负例的比例设为1:128。随后，交叉编码器以3e-5的学习率训练5个训练轮次。

指标。 为了评估交叉编码器的重排序性能，我们使用两种排序指标：MRR@10 和 nDCG@K。在 T²Ranking 的测试集中，每个查询平均有 15.7 个标注段落，最多有 100 个标注段落。我们报告测试查询上的 nDCG@20 和 nDCG@100。nDCG@K通过将 DCG@K 除以 iDCG@K（即段落理想排序的 DCG@K）来对 DCG@K 进行归一化。DCG@K 根据段落出现的排名对其分级相关性值进行折扣，定义如下:

$$DCG@K = \frac{1}{|Q|} \sum_{q \in Q} \sum_{p \in \mathcal{K}^q} \frac{L(p)}{\log_2(rank_p^{\mathcal{K}^q} + 1)}, \tag{7}$$

$$nDCG@K = \frac{DCG@K}{iDCG@K}, \tag{8}$$

其中，$L(p)$ 是段落 p的分级相关性。

重排序性能。 交叉编码器的重排序性能如表6所示。结果表明，对双编码器检索出的候选项进行重排序，显著优于对使用BM25方法检索出的候选项进行重排序。这种性能提升归因于双编码器方法实现了更高的召回率，这与此前在其他基准测试集上开展的研究结果一致 [15, 20]。然而，与其他基准测试集相比，T²Ranking 上的重排序性能较低 [15, 20]。这可以归因于 T²Ranking 中存在更多细粒度标注的相关段落和多样性更高的查询，这使其成为更具挑战性的基准，同时也能更准确地反映重排序性能。

表6：交叉编码器在T²Ranking测试集上使用挖掘负例的性能。

候选项	MRR@10	nDCG@20	nDCG@100
BM25的排名前1000的段落。	。5184	。4401	.4696
DE的排名前1000的段落。	.5520	.5149	。5571

7 结论

在本研究中，我们介绍了 T²Ranking，这是一个用于中文段落排序的大规模基准，涵盖检索和重排序任务。为了构建高质量数据集，我们采用了多种策略，包括基于模型的段落分割、基于聚类的段落去重，以及基于主动学习的数据采样。具体而言，我们在 T²Ranking 中采用基于模型的方法进行段落分割，旨在最大限度地保留每个段落的完整语义。为兼顾标注效率与已标注查询-段落对的多样性，我们在 T²Ranking 中引入基于聚类的技术，移除针对特定查询检索到的高度相似段落，从而在不损害数据集整体质量的前提下简化标注流程。在构建 T² Ranking 时采用主动学习策略，提高了标注更多信息丰富的训练样本的效率。主动学习框架能够以最少的标注量实现最佳性能，同时持续使用最有价值的样本更新数据集。此外，为确保标注质量，专家标注员参与实施了针对 T²Ranking 训练集和测试集的四级细粒度标注方案。该方案能够在训练期间对信息检索模型进行更细致的建模，并在测试期间更准确地评估模型。总之，T²Ranking 包含超过 30 万个查询和 200 多万个唯一段落，其中约 240 万个查询-段落对由专家标注员标注了细粒度相关性标签。据我们所知，T²Ranking 是规模最大的用于段落排序且采用细粒度标注的中文基准。我们相信，该数据集将为信息检索领域的发展和信息检索技术的进步作出重要贡献。

致谢

本工作得到清华大学-腾讯天工智能计算研究院、北京人工智能研究院（BAAI）和泉城实验室的支持。

参考文献

[1] Elif Aktolga、James Allan 和 David A Smith。2011。利用句法结构和答案类型进行问答段落重排序。载于信息检索进展：第 *33* 届欧洲信息检索研究会议，*ECIR 2011*，爱尔兰都柏林，*2011* 年 *4* 月 *18—21* 日。论文集 *33*。Springer，617–628。[2] Negar Arabzadeh、Alexandra Vtyurina、Xinyi Yan 和 Charles LA Clarke。2022。稀疏标签的浅层池化。《信息检索期刊》25，4（2022），365–385。[3] Luiz Bonifacio、Vitor Jeronymo、Hugo Queiroz Abonizio、Israel Campiotti、Marzieh Fadaee、Roberto Lotufo 和 Rodrigo Nogueira。2021。mmarco：MS MARCO 段落排序数据集的多语言版本。*arXiv* 预印本 *arXiv:2108.13897*（2021）。

[4] Anfeng Cheng、Yiding Liu、Weibin Li、Qian Dong、ShuaiqiangWang、Zhengjie Huang、Shikun Feng、Zhicong Cheng 和 DaweiYin。2023。《布局感知的网页质量评估》。arXiv 预印本 arXiv:2301.12152（2023）。[5] Nick Craswell、Bhaskar Mitra、Emine Yilmaz、Daniel Campos、Ellen M Voorhees和IanSoboroff。2021。《TREC 深度学习赛道：大数据场景中的可复用测试集》。载于第四十四届国际 ACM SIGIR 信息检索研究与发展会议论文集。2369–2375。[6] Laura Dietz、Manisha Verma、Filip Radlinski和NickCraswell。2017。《TREC复杂答案检索概述》。载于 TREC。[7] Qian Dong、YidingLiu、Suqi Cheng、Shuaiqiang Wang、ZhicongCheng、ShuziNiu 和Dawei Yin。2022。《在预训练语言模型中融入显式知识以进行段落重排序》。arXiv预印本 arXiv:2204.11673（2022）。[8] QianDong 和 Shuzi Niu。2021。《用于文档排序的潜在图循环网络》。载于《高级应用数据库系统：第26届国际会议，DASFAA 2021，中国台湾台北，2021 年 4 月 11–14 日，论文集》，第二部分 26。Springer，88–103。[9] Qian Dong、Shuzi Niu、TaoYuan 和 Yucheng Li。2022。《用于文档排序的解耦图循环网络》。数据科学与工程7，1（2022），30–43。[10] Yixing Fan、Xiaohui Xie、Yinqiong Cai、Jia Chen、Xinyu Ma、Xiangsheng Li、Ruqing Zhang、JiafengGuo等。2022。《信息检索中的预训练方法》。信息检索基础与趋势® 16，3（2022），178–317。[11] Mandar Joshi、Eunsol Choi、Daniel S Weld 和 Luke Zettlemoyer。2017。《TriviaQA：用于阅读理解的大规模远程监督挑战数据集》。arXiv预印本 arXiv:1705.03551（2017）。[12] VladimirKarpukhin、Barlas Oğuz、Sewon Min、Patrick Lewis、Ledell Wu、SergeyEdunov、DanqiChen和Wen-tauYih。2020。《用于开放域问答的密集段落检索》。arXiv 预印本 arXiv:2004.04906（2020）。[13] Jacob Devlin Ming-Wei ChangKenton和 Lee KristinaToutanova。2019。《BERT：用于语言理解的深度双向变换器预训练》。载于 NAACL-HLT 论文集，第 1卷。2。[14] JimmyLin、Xueguang Ma、Sheng-Chieh Lin、Jheng-Hong Yang、Ronak Pradeep 和 RodrigoNogueira。2021。《Pyserini：用于采用稀疏与密集表示开展可复现信息检索研究的 Python工具包》。载于第四十四届国际 ACMSIGIR 信息检索研究与发展会议论文集。2356–2362。[15] Dingkun Long、Qiong Gao、Kuan Zou、Guangwei Xu、Pengjun Xie、Ruijie Guo、Jian Xu、GuanjunJiang、LuxiXing和Ping Yang。2022。《Multi-CPR：用于段落检索的多领域中文数据集》。载于第四十五届国际 ACMSIGIR 信息检索研究与发展会议论文集。3046–3056。[16] Tri Nguyen、MirRosenberg、Xia Song、Jianfeng Gao、Saurabh Tiwary、Rangan Majumder 和 Li Deng。2016。《MS MARCO：人工生成的机器阅读理解数据集》。载于 CoCo@ NIPs。[17] Kyosuke Nishida、Itsumi Saito、Atsushi Otsuka、Hisako Asano和 Junji Tomita。2018。《检索与阅读：信息检索与阅读理解的多任务学习》。载于第二十七届 ACM 国际信息与知识管理会议论文集。647–656。[18] Rodrigo Nogueira、Wei Yang、KyunghyunCho和 Jimmy Lin。2019。《使用 BERT 的多阶段文档排序》。arXiv 预印本 arXiv:1910.14424（2019）。[19] Jay MPonte 和 WBruce Croft。2017。《信息检索的语言建模方法》。载于 ACMSIGIR Forum，第51 卷。美国纽约州纽约市 ACM，202–208。[20] Yifu Qiu、Hongyu Li、Yingqi Qu、Ying Chen、Qiaoqiao She、Jing Liu、Hua Wu 和Haifeng Wang。2022。《DuReader_检索：从网页搜索引擎进行段落检索的大规模中文基准》。arXiv 预印本 arXiv:2203.10232（2022）。[21] YingqiQu、Yuchen Ding、Jing Liu、KaiLiu、RuiyangRen、Wayne Xin Zhao、Daxiang Dong、Hua Wu 和 HaifengWang。2020。《RocketQA：用于开放域问答的密集段落检索优化训练方法》。arXiv 预印本 arXiv:2010.08191（2020）。[22] Pengzhen Ren、YunXiao、Xiaojun Chang、Po-Yao Huang、Zhihui Li、Brij BGupta、Xiaojiang Chen 和 Xin Wang。2021。《深度主动学习综述》。ACM计算调查（CSUR）54，9（2021），1–40。[23] Stephen Robertson、Hugo Zaragoza等。2009。《概率相关性框架：BM25及其他方法》。信息检索基础与趋势® 3，4（2009），333–389。[24] Kevin Roitero、Eddy Maddalena、GianlucaDemartini 和 StefanoMizzaro。2018。《论细粒度相关性等级》。载于第四十一届国际 ACMSIGIR 信息检索研究与发展会议。675–684。[25] ZhengyangTang、BenyouWang和Ting Yao。2022。《DPTDR：用于密集段落检索的深度提示调优》。arXiv 预印本 arXiv:2208.11503（2022）。[26] Joe H WardJr. 1963。《用于优化目标函数的层次聚类》。美国统计协会期刊 58，301（1963），236–244。[27] Zhijing Wu、Jiaxin Mao、Yiqun Liu、Jingtao Zhan、Yukun Zheng、MinZhang和 Shaoping Ma。2020。《利用段落级累积增益进行文档排序》。载于The Web Conference 2020 论文集。2421–2431。

[28] Lee Xiong、Chenyan Xiong、Ye Li、Kwok-Fung Tang、Jialin Liu、Paul Bennett、JunaidAhmed和Arnold Overwijk。2020。用于稠密文本检索的近似最近邻负对比学习。*arXiv* 预印本 *arXiv:2007.00808*（2020）。[29] Junqi Zhang、Yiqun Liu、ShaopingMa和 QiTian。2018。在搜索引擎结果页面上利用多种信息源进行相关性估计。在 第 *27* 届 *ACM* 国际信息与知识管理会议论文集 中。627–636。

[30] Yukun Zheng、Zhen Fan、Yiqun Liu、ChengLuo、Min Zhang和 Shaoping Ma。2018。Sogou-qcl：一个带有点击相关性标签的新数据集。载于 第*41*届国际 *ACMSIGIR*信息检索研究与发展会议。1117– 1120。[31] Cai-Nicolas Ziegler、Sean M McNee、Joseph A Konstan 和Georg Lausen。2005。通过主题多样化改进推荐列表。载于 第*14*届国际万维网会议论文集。22–32。