简单而有效的跨语言信息检索神经排序与重排序基线

Jimmy Lin 加拿大滑铁卢大学

David Alfonso-Hermelo 加拿大华为诺亚方舟实验室

Vitor Jeronymo 巴西坎皮纳斯州立大学

Ehsan Kamalloo 加拿大滑铁卢大学

Carlos Lassance Naver Labs Europe，法国

Rodrigo Nogueira 坎皮纳斯州立大学，巴西

Odunayo Ogundepo 滑铁卢大学，加拿大

Mehdi Rezagholizadeh 华为诺亚方舟实验室，加拿大

Nandan Thakur 加拿大滑铁卢大学

Jheng-Hong Yang 加拿大滑铁卢大学

Xinyu Zhang 加拿大滑铁卢大学

arXiv:2304.01019v1 [cs.IR] 3 Apr 2023

摘要

多语言语言模型的出现重新激发了人们对跨语言信息检索（CLIR）的兴趣。跨语言信息检索是指使用一种语言的查询来检索另一种语言的文档。然而，研究进展迅速，方法种类繁多，令人困惑，而可复现性却落后于最新技术水平。在此背景下，我们的工作作出了两项重要贡献：首先，我们以单语检索的多阶段架构为框架，提出一个用于组织不同跨语言检索方法的概念框架。其次，我们在 Anserini 和 Pyserini 信息检索工具包中，为 TREC 2022 NeuCLIR 赛道的波斯语、俄语和中文测试集合实现了简单而有效、可复现的基线。我们的工作建立在两个团队的合作之上，这两个团队提交了 TREC 评测中效果最好的运行结果。这些贡献为未来的进展奠定了坚实基础。

1 引言

跨语言信息检索（CLIR）是指使用一种语言的查询检索另一种语言文档的任务，例如，使用英语查询检索俄语文档。通常，跨语言信息检索系统是整体流程的一部分，该流程涉及机器翻译、相关的人类语言技术，有时还包括人类专家；这些环节共同帮助用户利用他们可能无法阅读的语言内容来满足信息需求。跨语言信息检索的研究已有数十年历史 [11, 16, 31, 43],，但近年来这一领域重新受到关注 [13, 49], ，主要得益于 mBERT [10] 和 XLM-R [6]等多语言预训练 Transformer 模型的出现。

跨语言信息检索领域近期研究活动的一个枢纽是 TREC NeuCLIR Track。该赛道于 TREC 2022 首次举办，并计划于 2023 年继续，或许还会延续到以后。该赛道为整个社区提供了一个评估跨语言信息检索系统的平台，评估背景是现代文档集合和系统，而如今这些系统主要由神经方法驱动。NeuCLIR 主题（即信息需求）以英语表述，参赛系统的任务是从中文、波斯语和俄语语料库中检索相关文档。

或许是该领域飞速发展带来的一个副作用，我们认为，信息检索社区仍不够清楚不同检索方法之间的关系（例如，稠密检索与稀疏表示、“学习得到的”与“启发式的”与“无监督的”等），也不清楚这些方法应如何应用于不同的检索场景。此外，当今检索模型日益复杂，现代软件栈也愈发繁复，这给可复现性工作带来了严峻挑战。这不仅使研究人员和从业者难以公平地比较不同方法，也为新手入门设置了障碍。即使在文档和查询使用同一种语言的单语检索中，这些问题也已存在。跨语言需求带来了额外的复杂性，使设计选择成倍增加（模型选择、训练方案、翻译系统的应用等），进一步模糊了概念上的清晰度，也增加了实验复现的难度。

贡献。 我们的工作旨在应对这些挑战，具体聚焦于帮助研究人员和从业者梳理现代神经检索技术背景下种类繁多的跨语言信息检索方法，而这些技术主要由深度学习驱动。我们的贡献可分为“概念性”和“实践性”两个部分：

从概念上讲，我们基于单语检索的多阶段排序这一通用设计，为组织不同的跨语言检索方法提供了一个框架。这些架构由第一阶段检索器组成，直接在任意规模的文档集合上执行前-k 检索，随后经过一个或多个重排序阶段，对第一阶段生成的候选结果的排名顺序进行优化。

最近，Lin [23]提出，检索技术可以根据其处理的表示形式——稠密语义向量还是稀疏词汇向量——以及权重的分配方式——是像 BM25 那样采用启发式方法，还是由使用标注数据训练的神经网络来分配——加以刻画。将这一观点应用于跨语言场景，自然会引出三种主要的第一阶段检索方法：文档翻译、查询翻译，以及使用语言无关表示。尽管这些方法已有数十年历史，但基于学习得到的表示也出现了“现代变体”，能够利用强大的预训练Transformer模型。

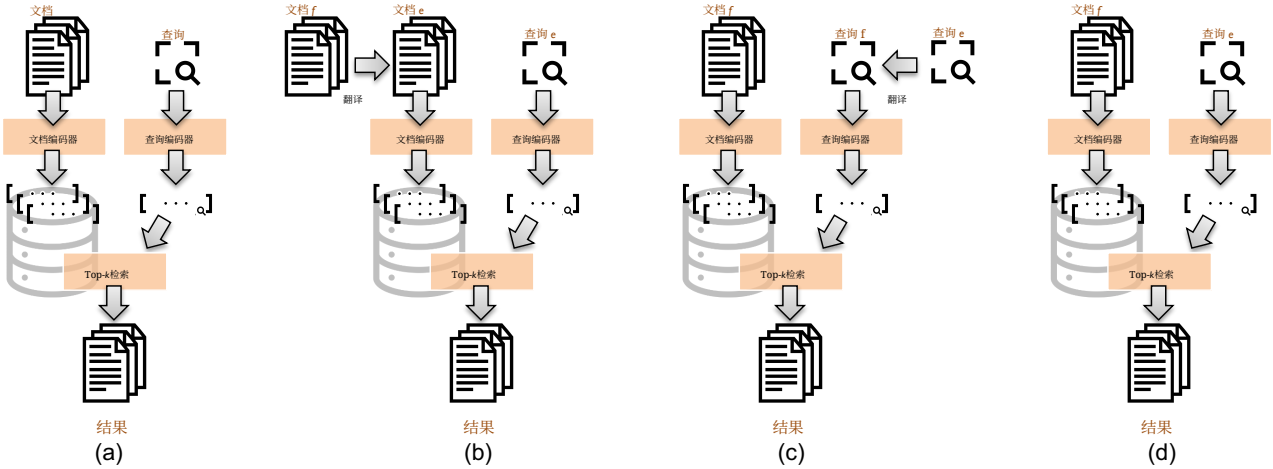

图 1：不同的检索架构：（a）一种单语双编码器架构，可同时实现稠密检索和稀疏检索方法；（b）适用于文档翻译的双编码器，其中所有文档都被翻译成 e，而查询仍使用 e；（c）适用于查询翻译的双编码器，其中查询 e 被翻译成 f，并针对 f 中的文档发起检索；（d）一种双编码器，其编码器可将多种语言的内容映射到同一表示空间。

对于单语检索，标准的多阶段架构会对第一阶段检索器的输出应用重排序器，例如上文所述的那些。在跨语言场景中，我们介绍如何利用现有的多语言模型来设计和构建跨语言重排序器。结果融合构成了我们概念框架的最后一个组成部分。在多阶段架构中，一个自然的问题是应在何时进行融合：这体现为我们所探讨的早期融合与后期融合技术。

具体而言，围绕上述概念框架，我们针对 TREC 2022 NeuCLIR 测试集提供了多个可复现的基线，其中包括提交至该评测的得分最高运行结果的变体。这些可复现的基线已纳入 Anserini 和 Pyserini 信息检索工具包。我们的工作建立在两个团队的合作之上，这两个团队提交了 TREC 评测中效果最好的运行结果。

我们希望这项工作能够为未来的研究奠定坚实基础，既提供概念框架，也提供参考实现，供学界进一步拓展。

2 单语检索概述

由于单语检索架构为跨语言检索提供了起点，因此先概述现代单语方法是合理的。在此，我们采用标准的（单语）检索任务表述（也称为即席检索）。给定一个有限但规模任意大的文档集合 $C = \{d_1, d_2 \ldots, d_n\}$，系统的任务是根据查询 q 返回文档排名前 k，使某种质量指标（如归一化折损累计增益或平均准确率）达到最大。

重排序器。 早在 21 世纪 00 年代中期，人们就开始在学习排序 [5]的背景下，以数据驱动的方式将神经网络应用于即席检索。此后，搜索引擎设计一直由多阶段排序架构 [30, 44]，主导：第一阶段检索器（通常就是 BM25 检索）会生成候选文档，再由一个或多个阶段对其进行重排序，通常使用机器学习模型。在“Transformer时代”，例如，可以用 BERT [32, 34] 和 T5 [33] 以完全相同的方式进行检索。使用预训练Transformer进行重排序时，需要向模型同时输入查询和候选文本；这种模型应用方式称为交叉编码器。

双编码器架构。 近期，段落检索领域的一项重要创新是引入了所谓的稠密检索模型，它们利用双编码器设计（与上文讨论的交叉编码器设计相对）：DPR [19] 和ANCE [45] 是早期的两个例子。有了足够的标注数据，我们便可以训练编码器（通常是基于 Transformer 的模型），将查询和文档投射到稠密（语义）表示空间（例如 768 维）中，从而将相关性排序转化为在表示向量上进行最近邻搜索。

在稠密检索模型问世后，研究人员很快意识到，也可以引导基于Transformer的编码器生成稀疏表示，例如，向量基底覆盖整个输入词汇空间。理解这些所谓的稀疏检索模型的另一种方式，是将它们与 BM25 进行对比：BM25 使用启发式评分函数分配词项权重，而稀疏检索模型则使用 BERT 等预训练 Transformer学习这些词项权重。这类学习得到的稀疏检索模型包括 DeepImpact [29], uniCOIL [24, 53],SPLADE [12], , 以及许多其他模型。

最近，Lin [23] 指出，稠密检索模型、稀疏检索模型和传统词袋模型（例如 BM25）都是双编码器架构的参数化变体，如图 1(a) 所示。在这三类模型中，“编码器”接收查询或文档，并生成向量表示。它们主要有两个差异维度，首先是表示向量的基础：稠密检索模型

生成稠密（语义）表示，而稀疏检索模型和词袋模型则将其表示向量建立在词汇空间中。另一个主要差异维度是这些表示是否经过学习：是，稠密检索模型和稀疏检索模型会学习这些表示；而否，传统词袋模型则不会。单语检索的概念框架为我们整理跨语言检索方法提供了基础，接下来我们将讨论这些方法。

3 跨语言检索方法

跨语言信息检索任务的形式化方式与单语检索任务类似。我们假设有一个由语言 f 的文档集合 Cf，其包含 $\{d_1, d_2 \ldots, d_n\}$。系统接收一个语言为 e 的查询 q，为清晰起见，我们将其记作 q_e；其任务是从 Cf 中返回文档排名前 k 的列表，以最大化某种质量指标，例如归一化折损累计增益或平均准确率。在本文中，e 指英语，f 指某种非英语语言（例如俄语），但一般情况下不必如此。

基于上一节介绍的单语检索架构设计，我们首先讨论第一阶段检索的三种可能设计：文档翻译、查询翻译，以及采用语言无关的表示。接着，我们概述用于对第一阶段检索器的输出进行重排序的交叉编码器，最后谈谈对融合技术的一些看法。

为进一步深入了解跨语言检索技术，我们介绍 TREC 2022 NeuCLIR 评测的一些细节。参与者需要根据英语查询，从三个独立的语料库中检索文档。这些语料库包含从 2016年8月1日到 2021年7月31日期间，整理自 Common Crawl 的波斯语、俄语和中文新闻通讯社文档。语料库规模适中：波斯语文档 223 万篇，俄语文档 463 万篇，中文文档 318 万篇。

信息需求（在 TREC 术语中称为主题）按照构建检索测试集的标准流程制定 [15, 42]。组织方发布了 114 个主题，最初以英语编写，随后由人工和 Google 翻译自动将其译成波斯语、俄语和中文。主题包含“标题”和“描述”字段，前者类似关键词查询，后者则用大致一个句子的篇幅阐述信息需求。根据设计，所有主题均相互对应，也就是说，每个主题都有三种语言的译文。然而，并非 所有主题都在所有语言中接受了评估：组织方总共发布了波斯语 46 个主题、俄语 45 个主题和中文 49 个主题的相关性判断。

3.1 文档翻译

一种非常简单的跨语言信息检索方法称为文档翻译：给定语言 e 中的查询 q_e 以及语言 f中的语料库 Cf，我们可以将整个语料库翻译成语言 e，即 $\{\text{Translate}(d_i)\}$，然后在语言 e中执行单语检索。图 1(b)展示了这一设计，其中主要新增的是一个文档翻译阶段，它会将翻译后的文档输入双编码器架构的文档端。

尽管翻译整个语料库可能耗时较长，但只需执行一次，因此可以视为一种开销较大的预处理步骤，类似于其他计算密集型文档扩展技术，例如 doc2query [35]。可以使用任何翻译技术，包括现成的机器翻译系统。一般而言，由于文档由结构完整的句子组成，自动翻译的输出通常相当流畅，具体效果取决于底层系统的质量。这与查询翻译（见下文）形成对比：查询翻译的质量往往较差，因为查询通常短得多（因而缺乏上下文），而且系统通常并未针对这类输入进行训练。

将 Cf 翻译为 Ce 后，任务便成为单语检索，因为查询也使用 e。在本研究中，三个语料库分别使用波斯语、俄语和中文；我们采用了 NeuCLIR 赛道组织者提供的英语译文，这些译文由 SockEye 机器翻译系统生成。我们从 NeuCLIR 主题中提取了三种类型的英语查询：仅使用“标题”字段、仅使用“描述”字段，以及同时使用两者。实验采用了两种检索模型和伪相关反馈：

BM25。 尽管近年来出现了许多神经排序模型，这种传统的“词袋”模型仍然是可靠的基线。

SPLADE。 我们选择了具备零样本能力的 SPLADE++ Ensemble Distil [12]。SPLADE 系列是一种稀疏神经检索模型，可学习文档扩展和查询扩展，并通过正则化项加以控制。

伪相关反馈（*PRF*）。 在 BM25 和 SPLADE 的结果之上，我们应用伪相关反馈。RM3 是一种广受欢迎的选择，也已在神经方法 [48], 领域得到充分研究；但本研究出于两个原因改用 Rocchio 反馈。第一，已有研究表明，Rocchio 反馈是稠密向量表示的一种有效伪相关反馈方法；将 Rocchio 用于词汇表示也能使概念保持统一。相比之下，稠密向量没有对应的 RM3 变体，因此更难比较稀疏和稠密伪相关反馈。第二，先前研究表明，Rocchio 的效果至少不逊于 RM3 [26], ，因此我们在不牺牲检索效果的前提下，获得了更简洁、一致的方法。

3.2 查询翻译

文档翻译的另一面称为查询翻译：给定语言 e 中的 q_e 和语言 f 中的语料库 Cf，我们可以将查询翻译成语言 f，即 $\text{Translate}(q_e) = q_f$，然后在语言 f 中执行单语检索。这一设计如图 1(c) 所示，其中我们添加了一个查询翻译组件，为双编码器架构的查询端提供输入。

查询翻译的计算效率远高于文档翻译，但也有前文讨论过的缺点——查询可能不是结构完整的句子，因此翻译起来可能更困难。不过，这种方法便于更快地进行实验，因为引入新的翻译模型无需重新翻译整个语料库。

查询翻译面临的一项挑战是，我们需要在 f 中有一个优质的单语检索模型，而根据定义，该语言并非英语。BM25 可以提供一个基线（基于语言 f 的词袋空间），但要获得有效的学习式检索模型则更加困难

因为非英语语言的人工标注数据较少。

我们的实验同时考察了赛道组织者提供的主题的人工翻译和机器翻译。根据每种翻译，我们可以创建三种类型的查询："标题"、"描述"和"两者"（与上文的文档翻译情况类似）。因此，我们总共有六种变体：{人工翻译，机器翻译} × {标题，描述，两者}。在这些条件下，我们使用两种不同的检索模型以及伪相关反馈进行了实验：

BM25。同样，这种传统的"词袋"模型仍然是一个稳健的基线。

SPLADE。要在非英语语言中构建 SPLADE模型，我们首先需要为相应语言选用一个优质的预训练语言模型。因此，此处使用的模型首先从头开始训练，采用 MLM+FLOPS 损失 [20]，并使用以下语料库的拼接数据：(i) 目标语言的 NeuCLIR 语料库，(ii) 目标语言的 MS MARCO 机器翻译 [4]，以及 (iii) 目标语言的 Mr. TyDi [51] 语料库（如果有）。最后，我们在目标语言版本的 MS MARCO 上进行微调，期望获得与英语中类似实验相近的零样本特性。每种语言都单独创建了一个模型。[1]

伪相关反馈。与文档翻译的情况一样，我们可以在 BM25 或 SPLADE 的基础上应用伪相关反馈。出于上述相同的原因，我们选择 Rocchio 作为反馈方法。

3.3 语言无关表示

从图 1(a) 所示的单语检索双编码器设计出发，人们可能会想知道，文档编码器和查询编码器是否能够生成某种语言无关的语义表示，从而支持跨语言直接匹配相关性。随着预训练多语言 Transformer 的出现，这确实成为可能。例如，我们可以将文档编码器应用于 Cf 中的文档（使用f语言），并将查询编码器应用于 e中的查询，然后直接根据这些表示进行相关性排序。因此，无需显式翻译查询或文档，我们也能执行跨语言检索。如图 1(d) 所示。

这种方法最直接的实现方式是训练一个 DPR 模型 [19],，但以 mBERT 等多语言 Transformer 骨干网络为基础。据我们所知，Asai 等人 [1] 最早提出了这种方法。最近，Zhang 等人 [52] 基于这一基本设计，提出了不同的方法：先在英语数据上进行"预微调"，再使用非英语数据对目标语言进一步微调，以利用跨语言迁移。尽管 Zhang 等人关注的是非英语语言的单语检索，但其中许多经验同样适用于跨语言检索。

具体而言，在本研究中，我们使用 XLM-R [6] 骨干网络初始化了一个多语言 DPR 模型，并对其进行了预先微调，[2] 将其命名为 xDPR。该模型在 MS MARCO 段落数据集上训练， [2], 其中查询编码器和段落编码器共享参数。

使用这个训练好的模型，我们分别对波斯语、俄语和中文语料库进行了编码。或许值得强调的是，这三种情况下使用的都是同一个模型。对于查询编码，我们有多种设计选择。与文档翻译和查询翻译类似，我们可以使用"标题"、"描述"或"两者"。此外，我们可以用 e或 f对查询进行编码。第一种情况下，我们要求编码器直接将 e 查询投射到 f 文档所处的语义空间中。第二种情况下，查询最初使用的是 f，因此模型会将 f 中的序列编码到 f 文档所处的语义空间中。因此，对于每种语言，我们总共得到九种变体：{原始查询、人工翻译、机器翻译} × {标题、描述、两者}。

最后，我们可以在 xDPR 检索到的结果上，按照 Li 等人 [21, 22]的工作，使用 Rocchio 方法进行伪相关反馈。因此，结合 Liu [26], ，我们能够以一致的方式在稠密检索和稀疏检索模型中实现 Rocchio 反馈。

3.4 重排序

在标准的多阶段排序架构中，第一阶段检索器会生成一个候选项排序列表，然后由一个或多个旨在改善排序的重排序阶段进行处理。跨语言场景同样适用重排序，但候选查询/文档对可能取决于第一阶段检索器而使用 e或 f。如果查询和文档都使用 e，我们可以使用单语英语重排序器。

对于基于文档翻译的第一阶段检索器，我们的实验使用了 monoT5，它基于 T5 [37]。重排序使用英语并采用以下提示：

Que查询：{query_text} 文档：{doc_text} 相关性 性：

模型需要生成"true"或"false"标记，据此可以提取相关性概率，并用它对候选项进行排序。monoT5 模型在 MS MARCO 段落数据集上微调后，在 TREC 深度学习竞赛 [8, 9], 上取得了最先进的结果，并在 BEIR [17] 及许多其他数据集 [38–40, 50] 上展现出出色的零样本检索效果。

对于基于查询翻译的第一阶段检索器进行重排序，我们使用了一种基于 T5 多语言版本 mT5 的变体。mT5 在多语言 mC4 数据集 [46]上进行了预训练；除此之外，我们采用相同的重排序方法。为了微调 mT5 以进行重排序，我们采用了与 Bonifacio 等人[4]类似的策略，使用了多语言版本的 MS MARCO 数据集 mMARCO。实验中，我们使用了具有 130 亿参数的 XXL 模型。

3.5 融合

数十年来，研究人员早已发现，融合技术可以结合多个独立检索结果列表中的证据，从而提升检索效果 [3, 41]。当各个检索结果列表基于不同的底层技术时，融合尤其有效，例如稠密检索与稀疏检索模型的对比 [14, 27]。鉴于我们的第一阶段检索器均基于截然不同的方法，

[1]SPLADE 和预训练模型可在 https://huggingface.co/naver/ modelname 获取，其中 modelname 为 =neuclir22-{pretrained,splade}-{fa,ru,zh} [2] https://huggingface.co/xlm-roberta-large

我们原本预期融合能显著提升检索效果，但实验结果似乎并未证实这一点。

在多阶段架构中，一个自然的问题是应该何时进行融合。一种可能的方法是分别对每个第一阶段检索器的输出进行重排序，然后融合这些结果；我们称之为后期融合。另一种可能的方法是先融合第一阶段检索器的输出，再对合并后的结果进行重排序；我们称之为早期融合。两种方法在检索效果上的差异是一个需要通过实验回答的问题，但后期融合的计算开销更大，因为它需要进行更多的重排序。

4 实现细节

本文所述的所有第一阶段检索和融合检索设置均通过 Anserini [47] 和 Pyserini [25]实现。Anserini 是一个基于 Java 的工具包，以开源搜索库 Lucene 为核心，旨在支持可复现的信息检索研究。Pyserini 为 Anserini 提供了 Python 接口，并通过支持稠密检索模型进一步扩展其功能。这些工具包在信息检索和自然语言处理领域的研究人员中得到了广泛应用。

对于使用 BM25 的文档翻译，我们的实现采用 Lucene 的英语默认分析器，该分析器会执行词元化、词干提取等操作。检索使用 Pyserini 的默认 BM25 参数 ($k_1 = 0.9, b = 0.4$)。对于查询翻译，需要注意的是，由于我们索引的是非英语文本，因此需要使用 f中的分析器。幸运的是，Lucene 已为这三种语言实现了分析器，我们直接使用了这些分析器。使用了相同的 BM25 参数。

所有 SPLADE 模型均在 Lucene 中使用标准的“伪文档”技巧 [28]实现。我们使用词元权重生成合成文档：将词元重复若干次，次数等于其权重（将权重量化为整数后）。例如，如果编码器为“car”赋予的权重为十，我们就将该词元重复十次。随后，这些伪文档像往常一样使用 Anserini 建立索引，其中权重存储在倒排索引的倒排记录中的词频位置。Top-k检索通过在 Lucene 中使用“词频求和”评分函数实现，其输出与根据查询向量和文档向量之间的内积进行排序完全相同。Anserini 提供了适当的抽象接口，可隐藏所有这些实现细节。

Pyserini 通过 Faiss 工具包 [18] 提供对稠密检索的支持；所有 xDPR 运行均使用平坦索引。对于 BM25 和 SPLADE 模型，Anserini 提供了适用于在 Python 中执行检索的语言绑定，而 Pyserini 则提供了相应接口，对使用稠密检索和稀疏模型的检索进行抽象和统一（即，它们仅仅是命令行参数中的参数化变体）。此外，Pyserini 还提供了倒数排名融合的实现，因此，整个基础设施使得混合搭配不同实验条件变得相当容易。

5 结果

我们的结果按以下阶段组织：第一阶段检索、重排序和融合。我们以以下指标报告检索效果：nDCG@20（NeuCLIR 评测的官方指标）以及截断于 1000 条结果时的召回率（recall@1000），后者用于衡量重排序检索效果的上限。组织方还将平均准确率均值（MAP）作为补充指标；我们也遵循了这一做法。总体而言，nDCG@20 和 MAP 得出的结论一致，因此为简洁起见，我们在报告中省略 MAP 结果。

在第3节中，我们描述了一个庞大的设计空间，其中包含可供多种重排序和融合方法使用的第一阶段变体。穷举检验所有可能的组合并不现实，因此，我们的实验通过逐步筛除“不值得关注”的设置来开展，具体如下文所述。

最后，谈一下显著性检验：我们当然充分认识到其重要性，但同样清楚多重假设检验所带来的风险。由于我们考察的条件数量众多，标准技术（例如邦费罗尼校正）很可能过于保守，难以检测出显著差异，尤其是考虑到 NeuCLIR 的主题规模相对较小。在大多数实验中，我们没有 进行显著性检验，而是将重点放在从大量实验条件中显现出的一般趋势上。我们更加审慎地应用显著性检验，以回答有针对性的研究问题。需要明确的是，我们报告的结果就是我们所进行的全部检验——也就是说，我们没有挑选最有希望的结果。在所有情况下，我们都使用了配对 t-检验（$p \leq 0.05$），并采用邦费罗尼校正。

5.1 第一阶段检索器

我们首先考察各个第一阶段检索器的输出。表1和表2分别展示了以 nDCG@20 和召回率@1000 衡量的结果。每组行按总体方法组织。各列按语言分组显示指标；每组中，我们报告使用由“标题”字段、“描述”字段以及两者组成的查询时的结果。

文档翻译。请注意，在文档翻译条件下，我们索引的是 NeuCLIR 组织者提供的机器翻译文档，这些文档为英语。第 (1ab) 行中的 BM25 条件与第 (2ab) 行中的 SPLADE 条件，区别仅在于应用于翻译语料库的检索模型。对于 BM25，在波斯语和中文中，“标题”和“两者”查询条件的检索效果（两项指标）大致相同，但在俄语中，“两者”的效果较差。对于所有语言，“描述”查询的表现似乎都较差。有趣的是，对于 SPLADE，在波斯语和中文中，三种查询类型在两项指标上的检索效果差距似乎都不大。这可能是因为该检索模型包含查询扩展，因此，更丰富地描述信息需求所带来的益处会有所减弱。

（a）与（b）行之间的比较凸显了伪相关反馈的影响。我们看到，最理想的情况下，伪相关反馈仅能使 BM25 的 nDCG@20 略有提升，而对于 SPLADE，伪相关反馈实际上会降低检索效果。然而，从表 2 中的召回率数据来看，伪相关反馈确实能够提升召回率。这种现象在预期之中，因为伪相关反馈主要是一种增强召回率的手段。

查询翻译。 对于 BM25，如第 (3a)–(3d) 行所示，俄语查询的“标题”和“两者”条件总体表现相当

nDCG@20		波斯语			俄语			中文		
	PRF	标题	desc	both	标题	desc	both	标题	desc	both
文档翻译 — BM25										
(1a) 官方 Sockeye 翻译	✗	0.3665	0.2889	0.3670	0.3693	0.2060	0.3080	0.3705	0.3070	0.3723
(1b) 官方 Sockeye 翻译	✓	0.3532	0.3127	0.3720	0.3589	0.2627	0.3188	0.3802	0.3206	0.3806
文档翻译 — SPLADE										
(2a) 官方 Sockeye 翻译	✗	0.4627	0.4618	0.4802	0.4865	0.4193	0.4573	0.4233	0.4299	0.4236
(2b) 官方 Sockeye 翻译	✓	0.4438	0.4675	0.4645	0.4836	0.4243	0.4604	0.4204	0.4142	0.4206
查询翻译 — BM25										
(3a) 人工翻译（HT）	✗	0.3428	0.2843	0.3429	0.3668	0.3138	0.3665	0.2478	0.2068	0.2572
(3b) 机器翻译（MT）	✗	0.3331	0.2974	0.3700	0.3564	0.2972	0.3605	0.1830	0.1498	0.1754
(3c) 人工翻译（HT）	✓	0.3356	0.2885	0.3408	0.3572	0.3366	0.3630	0.2544	0.1985	0.2734
(3d) 机器翻译（MT）	✓	0.3374	0.3300	0.3612	0.3426	0.3257	0.3764	0.1861	0.1464	0.1785
查询翻译 — SPLADE										
(4a) 人工翻译（HT）	✗	0.4301	0.4413	0.4788	0.4594	0.3922	0.4214	0.3110	0.2935	0.3143
(4b) 机器翻译（MT）	✗	0.4437	0.4300	0.4728	0.4452	0.3792	0.4156	0.2843	0.2527	0.2929
(4c) 人工翻译（HT）	✓	0.4348	0.4232	0.4146	0.4322	0.4133	0.4316	0.3198	0.2926	0.3077
(4d) 机器翻译（MT）	✓	0.4193	0.4121	0.4444	0.4337	0.3965	0.4075	0.2920	0.2562	0.3029
语言无关表示 — xDPR										
(5a) d: 原始语料库, q: 英语	✗	0.1522	0.1847	0.1804	0.2967	0.2913	0.2866	0.2200	0.2192	0.2185
(5b) d: 原始语料库, q: 人工翻译	✗	0.2776	0.2900	0.2953	0.3350	0.3276	0.3307	0.3197	0.3129	0.3035
(5c) d: 原始语料库, q: 机器翻译	✗	0.2721	0.2968	0.3055	0.3619	0.3348	0.3542	0.3025	0.2785	0.3013
(5d) d: 原始语料库, q: 英语	✓	0.1694	0.1996	0.1993	0.3116	0.3085	0.3045	0.2442	0.2343	0.2312
(5e) d: 原始语料库, q: 人工翻译	✓	0.3083	0.2988	0.3197	0.3349	0.3544	0.3578	0.3376	0.3463	0.3380
(5f) d: 原始语料库, q: 机器翻译	✓	0.3136	0.3012	0.3181	0.3727	0.3690	0.3793	0.3268	0.3041	0.3345

表 1：主要结果表，报告了各种第一阶段检索器的 nDCG@20。

在两项指标上，中文也是如此。对于 SPLADE，如第 (4a)–(4d) 行所示，似乎没有一致的结论：有些情况下，“两者”优于“标题”，而另一些情况下则相反。不过，就 nDCG@20 而言，单独使用“描述”似乎通常效果较差。

对于查询翻译，可以自然地比较人工翻译和机器翻译。在第 (3) 行和第 (4) 行中，(a) 和 (c) 条件对应于 (b) 和 (d) 条件。对于波斯语和俄语，无论是使用 BM25 还是 SPLADE，机器翻译查询的效果似乎都并非总是不如人工翻译。在某些情况下，我们甚至观察到机器翻译查询的表现优于相应的人工翻译查询。对于 BM25，需要注意的是，由于查询是词袋，只要包含正确的内容词，翻译是否流畅并不重要。对于 SPLADE，该模型似乎能够适应可能不够流畅的翻译。然而，在中文中，人工翻译与机器翻译之间似乎存在明显差距，人工翻译通常能带来更好的结果。

最后，与文档翻译的情况一致，伪相关反馈似乎并未提升 nDCG@20，但确实提高了召回率。这同样在预期之中。

语言无关表示。 表 1 和表 2 的最后几行展示了 xDPR 的检索效果。回顾一下我们的实验设计：在文档端，f中的原始语料库使用该模型进行编码。在查询端，有三种选择：直接编码英语查询、编码人工翻译（HT）查询，或编码机器翻译（MT）查询。它们分别显示在第 (5a)、(5b) 和 (5c) 行。我们看到，(5a) 行与 (5b) 行的检索效果差异很大，这表明，与将查询编码到 f 中相比，尝试将 e 中的查询直接编码到 f 中文档所占据的语义空间会造成很大的损失。显然，该模型无法将不同语言中含义相同的文本（即查询翻译）充分编码到同一语义空间中。无论采用哪种配置，在两种翻译类型和两项指标下，稠密检索模型的效果似乎都远不如 BM25 和 SPLADE 模型。不过，我们发现伪相关反馈似乎确实能提升检索效果，这与此前关于向量伪相关反馈的研究一致 [21, 22]。

5.2 重排序

在上一节中，我们针对每种语言考察了 $18 \times 3 = 54$ 不同条件下的第一阶段检索场景。逐一报告每种条件下的重排序结果并不现实，因此我们做出了一些取舍，以便集中关注重点：我们只考虑同时利用标题字段和描述字段的条件，因为这类查询似乎比仅使用标题字段的查询更稳健。我们

recall@1000		波斯语			俄语			中文		
	PRF	标题	desc	both	标题	desc	both	标题	desc	both
文档翻译 — BM25										
(1a) 官方 Sockeye 翻译	✗	0.7335	0.6319	0.7652	0.7409	0.5780	0.7255	0.7567	0.6639	0.7567
(1b) 官方 Sockeye 翻译	✓	0.8111	0.7638	0.8248	0.7908	0.6780	0.7798	0.8129	0.7404	0.8011
文档翻译 — SPLADE										
(2a) 官方 Sockeye 翻译	✗	0.8478	0.8796	0.8860	0.8538	0.8376	0.8513	0.7997	0.7597	0.7922
(2b) 官方 Sockeye 翻译	✓	0.8592	0.8735	0.8703	0.8686	0.8238	0.8544	0.8038	0.7623	0.8067
查询翻译 — BM25										
(3a) 人工翻译（HT）	✗	0.7128	0.7027	0.7373	0.7125	0.6655	0.7421	0.4759	0.4577	0.4940
(3b) 机器翻译（MT）	✗	0.7254	0.6815	0.7424	0.7332	0.6210	0.7373	0.3829	0.2989	0.4028
(3c) 人工翻译（HT）	✓	0.7691	0.7520	0.8092	0.7381	0.7276	0.7770	0.5230	0.5113	0.5327
(3d) 机器翻译（MT）	✓	0.7672	0.7033	0.7829	0.7439	0.7136	0.7959	0.4361	0.3748	0.4341
查询翻译 — SPLADE										
(4a) 人工翻译（HT）	✗	0.7652	0.8173	0.8239	0.7739	0.7200	0.7612	0.6803	0.6602	0.6551
(4b) 机器翻译（MT）	✗	0.8045	0.8172	0.8437	0.7725	0.7150	0.7669	0.6424	0.5919	0.6312
(4c) 人工翻译（HT）	✓	0.7897	0.8175	0.8245	0.7946	0.7209	0.7776	0.7100	0.7205	0.7029
(4d) 机器翻译（MT）	✓	0.8099	0.8117	0.8350	0.7918	0.7090	0.7590	0.6861	0.6096	0.6535
语言无关表示 — xDPR										
(5a) d: 原始语料库, q: 英语	✗	0.4910	0.5445	0.5393	0.5704	0.5627	0.5834	0.4161	0.4359	0.4386
(5b) d: 原始语料库, q: 人工翻译	✗	0.6288	0.6780	0.7088	0.6196	0.5825	0.6368	0.5773	0.5841	0.6031
(5c) d: 原始语料库, q: 机器翻译	✗	0.6333	0.6453	0.6850	0.6285	0.5649	0.6300	0.5420	0.5382	0.5873
(5d) d: 原始语料库, q: 英语	✓	0.4702	0.4981	0.5347	0.6251	0.5971	0.6212	0.4330	0.4714	0.4593
(5e) d: 原始语料库, q: 人工翻译	✓	0.6409	0.6612	0.7212	0.6541	0.5915	0.6346	0.6088	0.5939	0.6310
(5f) d: 原始语料库, q: 机器翻译	✓	0.6686	0.6516	0.7071	0.6784	0.6032	0.6475	0.5744	0.5375	0.6109

表 2：主要结果表，报告了各种第一阶段检索器的 recall@1000。

还重点关注不使用伪相关反馈（PRF）的运行结果，因为伪相关反馈会带来额外的计算开销（包括延迟和索引大小）。

对于每种语言，这会将纳入考量的第一阶段检索器数量减少到九个。我们对这些运行结果进行了重排序，并在重排序模型的输入模板中加入标题和描述字段。我们也非正式地考察了其他条件，但并未穷尽所有可能；这些条件似乎没有改变我们的整体发现。例如，我们尝试用伪相关反馈对第一阶段检索结果进行重排序，但结果并没有明显改善（尽管召回率更高）。

重排序结果见表 3。在第一阶段检索器的检索效果（“1st”列）下，我们报告（nDCG@20，recall@1000）：前者衡量候选项排序质量，后者衡量重排序器检索效果的上限。我们发现，重排序大幅提升了检索效果；这在意料之中，因为交叉编码器在各种设置下的检索效果早已广为人知（见第 3.4 节）。

然而，一个有趣的观察是，重排序缩小了最佳与最差第一阶段检索器之间的检索效果差距。例如，从明显不如 SPLADE 有效的 BM25 开始，重排序器能够弥补候选结果质量较低的问题，使端到端检索效果与对 SPLADE 结果进行重排序相差不大（至少在归一化折损累计增益方面如此）。事实上，在某些情况下，对 xDPR 结果进行重排序所得的分数甚至高于对 BM25 结果进行重排序所得的分数。尽管文献中此前已指出第一阶段检索器与重排序器之间存在“耦合效应”[14, 36], ，这一发现仍表明有必要对此进行进一步探索。

5.3 融合

融合时，可能组合的设计空间极其庞大，无法穷尽探索。为保持连贯性，我们仅关注重排序实验中的第一阶段检索器。在各种融合技术中，我们最终选择了倒数排名融合，这是一种简单、有效且稳健的方法 [7]。

基于这些考量，我们在表4中尝试了以下融合方案：(6a) 将文档翻译与 BM25 和 SPLADE 结合；(6b) 将查询翻译与 BM25 和 SPLADE 结合；(6c) 将文档翻译和查询翻译与 BM25 结合；(6d) 将 SPLADE 的文档翻译和查询翻译结合；(6e) 结合所有词汇方法；(6f) 结合两种稠密方法；(6g) 将所有方法结合。为方便起见，表4上半部分重复列出了第一阶段检索器的检索效果。表格下半部分用红色标出融合结果劣于最佳输入结果的情况。在这些情况下，融合并不比直接选择效果最佳的单次检索结果更有价值。

nDCG@20	波斯语		俄语		中文	
	1st	重排序	1st	重排序	1st	重排序
文档翻译 — BM25						
(1a) 官方 Sockeye 翻译	(0.3670, 0.7652)	0.5350	(0.3080, 0.7255)	0.5662	(0.3723, 0.7567)	0.4955
文档翻译 — SPLADE						
(2a) 官方 Sockeye 翻译	(0.4802, 0.8860)	0.5545	(0.4573, 0.8513)	0.5714	(0.4236, 0.7922)	0.5026
查询翻译 — BM25						
(3a) 人工翻译（HT）	(0.3429, 0.7373)	0.5346	(0.3665, 0.7421)	0.5745	(0.2572, 0.4940)	0.4300
(3b) 机器翻译（MT）	(0.3700, 0.7424)	0.5551	(0.3605, 0.7373)	0.5742	(0.1754, 0.4028)	0.3831
查询翻译 — SPLADE						
(4a) 人工翻译（HT）	(0.4788, 0.8239)	0.5722	(0.4214, 0.7612)	0.5823	(0.3143, 0.6551)	0.4980
(4b) 机器翻译（MT）	(0.4728, 0.8437)	0.5932	(0.4156, 0.7669)	0.5767	(0.2929, 0.6312)	0.5132
语言无关表示 — xDPR						
(5a) d: 原始语料库, q: 英语	(0.1804, 0.5393)	0.4630	(0.2866, 0.5834)	0.5305	(0.2185, 0.4386)	0.4440
(5b) d: 原始语料库, q: 人工翻译	(0.2953, 0.7088)	0.5614	(0.3307, 0.6368)	0.5617	(0.3035, 0.6031)	0.5008
(5c) d：原始语料库，q：机器翻译	(0.3055, 0.6850)	0.5644	(0.3542, 0.6300)	0.5337	(0.3013, 0.5873)	0.5087

表 3：对各种第一阶段检索器进行重排序的结果（nDCG@20）。在“1st”列中，为方便起见，我们重复列出第一阶段检索器的（nDCG@20，召回率@1000）指标。在所有情况下，我们均使用标题和描述作为第一阶段检索和重排序的查询（未使用伪相关反馈）。

		nDCG@20			召回率@1000		
		波斯语	俄语	中文	波斯语	俄语	中文
(1a)	文档翻译–BM25	0.3670	0.3080	0.3723	0.7652	0.7255	0.7567
(2a)	文档翻译–SPLADE	0.4802	0.4573	0.4236	0.8860	0.8513	0.7922
(3b)	查询翻译–BM25	0.3700	0.3605	0.1754	0.7424	0.7373	0.4028
(4b)	查询翻译–SPLADE	0.4728	0.4156	0.2929	0.8437	0.7669	0.6312
(5a)	稠密检索–e	0.1804	0.2866	0.2185	0.5393	0.5834	0.4386
(5c)	稠密检索–f	0.3055	0.3542	0.3013	0.6850	0.6300	0.5873
(6a)	倒数排名融合(1a, 2a)：文档翻译–BM25、文档翻译–SPLADE	0.4462	0.4180	0.4189	0.8936	0.8670	0.8536
(6b)	倒数排名融合(3b, 4b)：查询翻译–BM25、查询翻译–SPLADE	0.4610	0.4598	0.2981	0.8703	0.8368	0.6692
(6c)	倒数排名融合(1a, 3b)：文档翻译–BM25、查询翻译–BM25	0.3795	0.3635	0.2736	0.7901	0.7686	0.7366
(6d)	倒数排名融合(2a, 4b)：文档翻译–SPLADE，查询翻译–SPLADE	0.5165	0.4921	0.4178	0.9009	0.8508	0.7938
(6e)	倒数排名融合(1a, 2a, 3b, 4b)：文档翻译、查询翻译	0.4897	0.4857	0.4397	0.9285†	0.8880	0.8637†
(6f)	倒数排名融合(5a, 5c)：稠密检索	0.2640	0.3469	0.2731	0.6814	0.6493	0.5693
(6g)	倒数排名融合(1a, 2a, 3b, 4b, 5a, 5c)：文档翻译、查询翻译、稠密检索	0.4926	0.5142†	0.4541	0.9291†	0.8818	0.8704†

表4：不同融合组合的结果。为方便起见，重复列出了各个第一阶段检索器的分数。所有情况下，我们都使用标题和描述作为查询，且未使用伪相关反馈。红色表示融合效果不如最佳单一输入检索结果的情况。† 表示相较于 (2a) 有显著提升。

从这些结果来看，对于波斯语和俄语，将文档翻译和查询翻译的 SPLADE 模型融合起来似乎能取得最佳检索效果，即第 (6d) 行；但对于中文，这种融合的效果略逊于仅使用文档翻译的 SPLADE。融合所有词汇检索结果（第 (6e) 行），在波斯语和俄语上的效果略逊于仅融合 SPLADE 检索结果，但能提升中文效果。最后，与单纯融合词汇检索结果相比，纳入语言无关的稠密检索技术所提供的证据似乎更有价值，这一点可从比较 (6g) 和 (6e)。鉴于稠密检索结果本身相当差，这一点令人意外。

总体而言，我们对融合未能像预期那样稳定地提升检索效果这一发现感到有些意外。表4中，红色数字标出了融合实际上损害了检索效果的所有情况，即融合效果差于表现最好的单次输入运行结果。我们认为，这一发现源于各次运行的检索效果差异很大：如果某个融合输入的效果远胜其他输入，倒数排名融合（RRF）的表现就不那么理想。

		波斯语			俄语			中文		
		1st	早期	late	1st	早期	late	1st	早期	late
(4a)	QT–SPLADE = 最佳单项	0.4728	0.5932		0.4156	0.5767		0.2929	0.5132	
(6c)	RRF(1a, 3b)：DT–BM25、QT–BM25	0.3795	0.5869	0.5723	0.3635	0.5788	0.5890	0.2736	0.5257†	0.4150
(6d)	RRF(2a, 4b)：文档翻译–SPLADE，查询翻译–SPLADE	0.5165	0.5823	0.6122	0.4921	0.5729	0.5915	0.4178	0.5379	0.5272
(6e)	RRF(1a, 2a, 3b, 4b)：文档翻译、查询翻译	0.4897	0.5901	0.5911	0.4857	0.5728	0.5853	0.4397	0.5394	0.5058
(6f)	RRF(5a, 5c)：稠密检索	0.2640	0.5621†	0.4573	0.3469	0.5438	0.5162	0.2731	0.5077†	0.4470
(6g)	倒数排名融合(1a, 2a, 3b, 4b, 5a, 5c)：文档翻译、查询翻译、稠密检索	0.4926	0.5893	0.5626	0.5142	0.5676	0.5840	0.4541	0.5340	0.5295

表4：比　　较早期融合与后期融合。† 表示相较于后期融合有显著提升　　.

为了更严谨地检验这一观察结果，我们进行了显著性检验，将文档翻译SPLADE模型（表4第(2a)行）与SPLADE模型融合（第(6d)行）、所有词汇模型融合（第(6e)行），以及所有词汇模型与稠密检索模型融合（第(6g)行）进行比较。这些比较从单个最佳第一阶段检索器出发，回答以下问题：SPLADE融合能否带来额外收益？BM25呢？稠密检索呢？

表4中的结论（以符号†标示）是，大多数融合组合在统计上并未显著优于采用SPLADE的文档翻译，而SPLADE是表现最佳的单一第一阶段检索器。就nDCG@20而言，最大的集成仅在俄语上显著优于DT–SPLADE；就recall@1000而言，我们观察到更多显著提升，但仅出现在波斯语和中文上。值得注意的是，将文档翻译和查询翻译两者与SPLADE结合（第6d行），并未显著优于单独使用DT–SPLADE。

在最后一组实验中，我们针对表4中的部分条件比较了早期融合与后期融合的检索效果。结果见表5。在这种情况下，我们以 QT–SPLADE 作为比较基准，因为它似乎是效果最佳的单阶段检索器与重排序组合。对于波斯语，除（6f）外，后期融合的效果似乎与早期融合相当或略好；俄语的情况似乎也类似，不过后期融合的提升幅度看起来较小。中文的结果则不太一致，有些情况下早期融合优于后期融合。为了更严格地比较早期融合与后期融合，我们对所有成对组合进行了显著性检验。只有少数差异达到显著水平，而且仅出现在早期融合优于后期融合的情况下。不过，其中三种情况有两种出现在稠密检索模型中，而这类模型的初始效果就较差。总体而言，这些实验并未明确指出哪种融合策略更优。

作为补充背景，NeuCLIR 2022 评测中的最佳结果由我们团队成员取得，但这些结果是在截止日期的时间压力下生成的，因此无法像表 5 中那样仔细考察所有配置。最佳结果（nDCG @20 分数）如下：（i）波斯语：p2.fa.rerank，0.588；（ii）俄语：p3.ru.mono，0.567；（iii）中文：p2.zh.rerank，0.516。将这些结果与本文报告的最佳条件进行比较后，我们确认，仅通过仔细研究不同系统组件的各种影响，就有可能在所有语言上取得改进，并实现新的最先进检索效果：（i）波斯语：6d 后期融合，0.612（+0.024）；（ii）俄语：6d 后期融合，0.592（+0.025）；（iii）中文：6e 早期融合，0.539（+0.023）。

6 结论

TREC 2022 的 NeuCLIR 评测标志着人们在“神经网络时代”重新关注跨语言信息检索挑战。总体而言，本文展现了两个提交了最有效运行结果的团队与一个拥有构建检索工具包经验的研究小组之间的合作。我们共同以更具原则性的方式梳理了评测中采用的各类方法，并基于单语检索提出了一个组织性概念框架。

总体上有哪些启示？查询翻译和文档翻译这两种已有数十年历史的通用方法，似乎都能很好地适应神经网络时代。尤其是，SPLADE 的效果似乎非常出色，展现了稀疏学习表示的潜力。尽管语言无关表示的效果似乎不如查询翻译或文档翻译，但我们对这类技术的探索才刚刚开始，因为 xDPR 目前只能算是一种基线。不过，考虑到同一个 xDPR 模型适用于所有三种语言，我们可以看到其巨大的潜力。

在自然语言处理和信息检索领域，我们才刚刚开始探索大型预训练变换器模型的应用。我们认为，未来前景令人振奋，并相信我们的概念框架、实验结果和软件基础设施为进一步探索奠定了坚实基础。

致谢

本研究部分得到了加拿大自然科学与工程研究理事会（NSERC）的资助。

参考文献

[1] Akari Asai、Jungo Kasai、Jonathan Clark、Kenton Lee、Eunsol Choi 和 Hannaneh Hajishirzi。2021年。XOR QA：跨语言开放检索问答。载于*2021*年计算语言学协会北美分会会议论文集：人类语言技术。计算语言学协会，线上，547–564。[2] Payal Bajaj、Daniel Campos、Nick Craswell、Li Deng、Jianfeng Gao、Xiaodong Liu、Rangan Majumder、Andrew McNamara、Bhaskar Mitra、Tri Nguyen、Mir Rosenberg、Xia Song、Alina Stoica、Saurabh Tiwary 和 Tong Wang。2018年。MS MARCO：人工生成的机器阅读理解数据集。*arXiv:1611.09268v3*（2018年）。

[3] Brian T. Bartell、Garrison W. Cottrell 和 Richard K. Belew。1994年。多个排序检索系统的自动组合。载于第17届 ACM SIGIR 信息检索研究与发展大会论文集（SIGIR 1994）。爱尔兰都柏林，173–181。[4] Luiz Bonifacio、Vitor Jeronymo、Hugo Queiroz Abonizio、Israel Campiotti、Marzieh Fadaee、Roberto Lotufo 和 Rodrigo Nogueira。2021年。mMARCO：MS MARCO 段落排序数据集的多语言版本。arXiv:2108.13897（2021年）。[5] Christopher J. C. Burges、Tal Shaked、Erin Renshaw、Ari Lazier、Matt Deeds、Nicole Hamilton 和 Greg Hullender。2005年。使用梯度下降进行学习排序。载于第22届国际机器学习会议论文集（ICML 2005）。德国波恩，89–96。[6] Alexis Conneau、Kartikay Khandelwal、Naman Goyal、Vishrav Chaudhary、Guillaume Wenzek、Francisco Guzmán、Edouard Grave、Myle Ott、Luke Zettlemoyer 和 Veselin Stoyanov。2020年。大规模无监督跨语言表示学习。载于计算语言学协会第58届年会论文集。计算语言学协会，线上，8440–8451。[7] Gordon V. Cormack、Charles L. A. Clarke 和 Stefan Büttcher。2009年。倒数排名融合优于孔多塞法和单独的排序学习方法。载于第32届 ACM SIGIR 信息检索研究与发展大会论文集（SIGIR 2009）。美国马萨诸塞州波士顿，758–759。[8] Nick Craswell、Bhaskar Mitra、Emine Yilmaz 和 Daniel Campos。2020年。TREC 2020 深度学习赛道概述。载于第二十九届文本检索会议论文集（TREC 2020）。美国马里兰州盖瑟斯堡。[9] Nick Craswell、Bhaskar Mitra、Emine Yilmaz、Daniel Campos 和 Ellen M. Voorhees。2019年。TREC 2019 深度学习赛道概述。载于第二十八届文本检索会议论文集（TREC 2019）。美国马里兰州盖瑟斯堡。[10] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。BERT：用于语言理解的深度双向变换器预训练。载于2019年计算语言学协会北美分会会议论文集：人类语言技术，第1卷（长篇和短篇论文）。计算语言学协会，美国明尼苏达州明尼阿波利斯，4171–4186。[11] Marcello Federico 和 Nicola Bertoldi。2002年。使用 N最佳查询翻译的统计跨语言信息检索。载于第25届 ACM SIGIR 信息检索研究与发展大会论文集（SIGIR 2002）。芬兰坦佩雷，49–56。[12] Thibault Formal、Carlos Lassance、Benjamin Piwowarski 和 Stéphane Clinchant。2022年。从蒸馏到困难负样本采样：提升稀疏神经信息检索模型的效果。载于第45届 ACM SIGIR 信息检索研究与发展大会论文集。西班牙马德里，2353–2359。[13] Petra Galuščáková、Douglas W. Oard 和 Suraj Nair。2021年。跨语言信息检索。arXiv:2111.05988（2021年）。[14] Luyu Gao、Zhuyun Dai、Tongfei Chen、Zhen Fan、Benjamin Van Durme 和 Jamie Callan。2021年。以语义残差嵌入补充词汇检索。载于第43届欧洲信息检索会议（ECIR 2021）论文集，第一部分。146–160。[15] Donna Harman。2011年。信息检索评估。Morgan & Claypool 出版社。[16] David A. Hull 和 Gregory Grefenstette。1996年。跨语言查询：多语言信息检索的基于词典的方法。载于第19届 ACM SIGIR 信息检索研究与发展大会论文集（SIGIR 1996）。瑞士苏黎世，49–57。[17] Vitor Jeronymo、Luiz Bonifacio、Hugo Abonizio、Marzieh Fadaee、Roberto Lotufo、Jakub Zavrel 和 Rodrigo Nogueira。2023年。InPars-v2：将大型语言模型用作高效的信息检索数据集生成器。arXiv 预印本 arXiv:2301.01820（2023年）。[18] Jeff Johnson、Matthijs Douze 和 Hervé Jégou。2019年。使用图形处理器进行十亿规模的相似性搜索。《IEEE大数据汇刊》7卷3期（2019年），535–547。[19] Vladimir Karpukhin、Barlas Oguz、Sewon Min、Patrick Lewis、Ledell Wu、Sergey Edunov、Danqi Chen 和 Wen-tau Yih。2020年。用于开放域问答的稠密段落检索。载于2020年自然语言处理实证方法会议（EMNLP）论文集。计算语言学协会，线上，6769–6781。[20] Carlos Lassance、Hervé Déjean 和 Stéphane Clinchant。2023年。关于从头预训练变换器用于信息检索的实验研究。arXiv:2301.10444（2023年）。[21] Hang Li、Ahmed Mourad、Shengyao Zhuang、Bevan Koopman 和 Guido Zuccon。2021年。深度语言模型与稠密检索器的伪相关反馈：成效与陷阱。arXiv:2108.11044（2021年）。[22] Hang Li、Shuai Wang、Shengyao Zhuang、Ahmed Mourad、Xueguang Ma、Jimmy Lin 和 Guido Zuccon。2022年。是否进行插值：伪相关反馈、稠密与稀疏检索器。载于第45届 ACM SIGIR 信息检索研究与发展大会论文集（SIGIR 2022）。西班牙马德里，2495–2500。[23] Jimmy Lin. 2021. 信息检索表征方法的概念框架提案。《SIGIR Forum》55, 2 (2021), 4:1–29. [24] Jimmy Lin 和 Xueguang Ma. 2021. 关于 DeepImpact、COIL 及信息检索技术概念框架的几点简短说明。arXiv:2106.14807 (2021). [25] Jimmy Lin、Xueguang Ma、Sheng-Chieh Lin、Jheng-Hong Yang、Ronak Pradeep 和 Rodrigo Nogueira. 2021. Pyserini：用于稀疏表示与稠密表示可复现信息检索研究的 Python 工具包。载于第44届ACM SIGIR信息检索研究与发展国际年会论文集（SIGIR 2021）。2356–2362. [26] Yuqi Liu. 2022. 使用 Rocchio 技术和文本分类实现简单而有效的伪相关反馈。硕士论文。滑铁卢大学。[27] Xueguang Ma、Kai Sun、Ronak Pradeep、Minghan Li 和 Jimmy Lin. 2022. 再看 DPR：训练复现与检索复现。载于第44届欧洲信息检索会议（ECIR 2022）论文集，第一部分。挪威斯塔万格，613–626. [28] Joel Mackenzie、Andrew Trotman 和 Jimmy Lin. 2021. 学习得到的稀疏表征中的权重乱象，以及逐分数查询评估的回归。arXiv:2110.11540 (2021). [29] Antonio Mallia、Omar Khattab、Torsten Suel 和 Nicola Tonellotto. 2021. 为倒排索引学习段落影响分数。载于第44届ACM SIGIR信息检索研究与发展国际年会论文集（SIGIR 2021）。1723–1727. [30] Irina Matveeva、Chris Burges、Timo Burkard、Andy Laucius 和 Leon Wong. 2006. 利用多层嵌套排序器实现高精度检索。载于第29届ACM SIGIR信息检索研究与发展国际年会论文集（SIGIR 2006）。华盛顿州西雅图，437–444. [31] Jian-Yun Nie. 2010. 跨语言信息检索。Morgan & Claypool 出版社。[32] Rodrigo Nogueira 和 Kyunghyun Cho. 2019. 使用 BERT 进行段落重排序。arXiv:1901.04085 (2019). [33] Rodrigo Nogueira、Zhiying Jiang、Ronak Pradeep 和 Jimmy Lin. 2020. 使用预训练序列到序列模型进行文档排序。载于计算语言学协会 EMNLP 2020 论文集：Findings。708–718. [34] Rodrigo Nogueira、Wei Yang、Kyunghyun Cho 和 Jimmy Lin. 2019. 使用 BERT 进行多阶段文档排序。arXiv:1910.14424 (2019). [35] Rodrigo Nogueira、Wei Yang、Jimmy Lin 和 Kyunghyun Cho. 2019. 基于查询预测的文档扩展。arXiv:1904.08375 (2019). [36] Ronak Pradeep、Yuqi Liu、Xinyu Zhang、Yilin Li、Andrew Yates 和 Jimmy Lin. 2022. 从石头中榨出水分：进一步提升交叉编码器重排序效果的多种技巧。载于第44届欧洲信息检索会议（ECIR 2022）论文集，第一部分。挪威斯塔万格，655–670. [37] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu. 2020. 探索统一文本到文本 Transformer 的迁移学习极限。《机器学习研究杂志》21, 140 (2020), 1–67. [38] Kirk Roberts、Dina Demner-Fushman、Ellen M. Voorhees、William R. Hersh、Steven Bedrick、Alexander J. Lazar、Shubham Pant 和 Funda Meric-Bernstam. 2019. TREC 2019 精准医学赛道概述。载于第二十八届文本检索会议（TREC 2019）论文集。马里兰州盖瑟斯堡。[39] Guilherme Moraes Rosa、Luiz Bonifacio、Vitor Jeronymo、Hugo Abonizio、Roberto Lotufo 和 Rodrigo Nogueira. 2022. 参数量以十亿计的模型胜过域内训练数据：法律案例蕴含任务案例研究。arXiv:2205.15172 (2022). [40] Guilherme Moraes Rosa、Ruan Chaves Rodrigues、Roberto de Alencar Lotufo 和 Rodrigo Nogueira. 2021. 是否进行微调？用于法律案例蕴含任务的零样本模型。载于第十八届人工智能与法律国际会议论文集。295–300. [41] Christopher C. Vogt 和 Garrison W. Cottrell. 1999. 通过分数的线性组合进行融合。《信息检索》1, 3 (1999), 151–173. [42] Ellen M. Voorhees. 2002. 信息检索评估哲学。载于《跨语言信息检索系统评估：第二届跨语言评估论坛研讨会》，《计算机科学讲义》第2406卷。355–370. [43] Jianqiang Wang 和 Douglas W. Oard. 2006. 结合双向翻译与同义关系进行跨语言信息检索。载于第29届ACM SIGIR信息检索研究与发展国际年会论文集（SIGIR 2006）。美国华盛顿州西雅图，202–209. [44] Lidan Wang、Jimmy Lin 和 Donald Metzler. 2010. 学习高效排序。载于第33届ACM SIGIR信息检索研究与发展国际年会论文集（SIGIR 2010）。瑞士日内瓦，138–145. [45] Lee Xiong、Chenyan Xiong、Ye Li、Kwok-Fung Tang、Jialin Liu、Paul N. Bennett、Junaid Ahmed 和 Arnold Overwijk. 2021. 面向稠密文本检索的近似最近邻负对比学习。载于第9届国际学习表征会议论文集（ICLR 2021）。

[46] Linting Xue、Noah Constant、Adam Roberts、Mihir Kale、Rami Al-Rfou、Aditya Siddhant、Aditya Barua 和 Colin Raffel. 2021. mT5：大规模多语言预训练文本到文本 Transformer。载于*2021*年计算语言学协会北美分会人类语言技术会议论文集。计算语言学协会，线上，483–498.[47] Peilin Yang、方辉和 Jimmy Lin. 2018. Anserini：使用 Lucene 实现可复现的排序基线。《数据与信息质量杂志》10, 4 (2018), 文章 16.[48] Wei Yang、陆旷、Peilin Yang 和 Jimmy Lin. 2019. 批判性审视“神经网络炒作”：神经排序模型的弱基线与效果增益的可加性。载于第*42*届*ACM SIGIR*信息检索研究与发展国际年会论文集（*SIGIR 2019*）。法国巴黎，1129–1132. [49] Puxuan Yu 和 James Allan. 2020. 跨语言信息检索中神经匹配模型的研究。载于第*43*届*ACM SIGIR*信息检索研究与发展国际会议论文集（*SIGIR 2020*）。1637–1640. [50] Edwin Zhang、Nikhil Gupta、Rodrigo Nogueira、Jimmy Lin 和 Kyunghyun Cho. 2020. 快速部署面向新冠肺炎开放研究数据集的神经搜索引擎。载于*ACL 2020* 第一届新冠肺炎自然语言处理研讨会论文集。[51] Xinyu Zhang、Xueguang Ma、Peng Shi 和 Jimmy Lin. 2021. Mr. TyDi：面向稠密检索的多语言基准。载于第一届多语言表征学习研讨会论文集。多米尼加共和国蓬塔卡纳，127–137. [52] Xinyu Zhang、Kelechi Ogueji、Xueguang Ma 和 Jimmy Lin. 2022. 探索训练多语言稠密检索模型的最佳实践。*arXiv:2204.02363* (2022). [53] Shengyao Zhuang 和 Guido Zuccon. 2021. 使用上下文化精确词项匹配和高效段落扩展实现快速段落重排序。*arXiv:2108.08513* (2021).