arXiv:2304.01974v1 [cs.CL] 4 Apr 2023

用于病史采集的对话上下文重排序

Jian Zhu[*]不列颠哥伦比
亚大学 jian.zhu@ubc.ca

Ilya Valmianski [†]
AuxHealth
ilya@auxhealth.io

Anitha Kannan
Curai Health
anitha@curai.com

摘要

人工智能驱动的病史采集是症状检查、自动化患者接诊、分诊及其他人工智能虚拟医疗应用的重要组成部分。由于病史采集的情况极为多样，机器学习模型需要大量数据进行训练。为应对这一挑战，现有系统采用间接数据或专家知识开发。这导致了训练-推理差距，因为模型训练所用的数据类型与其在推理时观察到的数据不同。在这项工作中，我们提出了一种两阶段重排序方法，通过使用对话上下文模型对第一阶段的候选问题进行重排序，帮助缩小训练-推理差距。为此，我们提出了一种新模型——全局重排序器，它将对话与所有问题同时进行交叉编码，并将其与若干现有神经基线模型进行比较。我们测试了基于 Transformer 和 S4 的语言模型主干网络。结果发现，与专家系统相比，我们提出的全局重排序器搭配 Transformer 主干网络取得了最佳表现，其归一化折损累积增益（nDCG）提高了 30%，平均精度均值（mAP）提高了 77%。作为这项工作的一部分，我们还发布了在维基百科和 PubMed 数据上训练的双向和自回归 S4 模型的预训练检查点。

1 引言

病史采集是医疗就诊中的关键环节 [15]。它包括收集相关的患者报告信息，例如主诉症状、患者关注事项，以及既往病史、心理史和社会史。这些信息构成后续患者分诊、诊断和医疗护理规划的基础。病史采集虽然是医疗就诊的重要组成部分，但也是最耗时的环节之一 [37, 6]；如果采集不完整，可能导致分诊、诊断和治疗错误 [15]。

五十多年来，人们一直在努力开发能够自动完成部分病史采集工作的工具 [45]。这类工具中最简单的是静态就诊前问卷，如今已在美国医疗保健领域广泛使用。然而，静态问卷往往篇幅很长，会提出不太相关的问题，也无法根据患者的需求进行定制。近来，有研究致力于构建能够根据患者回应调整问题的智能系统（见 [41, 9] 及其中的引文）。然而，开发这类系统所需的医学推理并非易事。现有方法包括

[*]在 Curai Health 实习期间完成的工作
[†]在 Curai Health 任职期间完成的工作

预印本。正在审稿。

使用模拟患者进行强化学习 [38, 19], 基于临床记录进行监督学习[41],以及专家系统 [9]。

在此前所有研究中，医学推理系统都是基于可替代真实医患互动的数据构建的。这是因为，一方面，可用的医患病史采集对话数据很少；另一方面，病史采集过程中可能提出的问题范围非常广。因此，训练病史采集模型需要大量标注的互动数据。然而，这类数据更容易从间接来源获取，例如医疗记录、专家知识或模拟。这造成了训练-推理差距：用于训练模型的数据，无法充分代表模型在推理时接触到的数据。

这种训练-推理差距会显著影响病史采集的质量，尤其是因为在一次就诊中只能提出少数几个问题。因此，我们需要一种方法，能够在直接训练数据有限的情况下，协调两方面的需求：既要支持大量潜在问题，又要将其缩减为一小组与患者健康问题相关的恰当问题。

本文从一个专家系统出发，展示如何利用少量真实医患对话数据来弥合这一训练-推理差距。我们的灵感来自信息检索领域，其中“先检索再重排序”是一种广受欢迎的范式，可用于从大型语料库中高效检索文档 [28, 22]。在我们的方案中，“检索”环节由专家系统完成，它会检索出一系列可向患者提出的问题；随后，基于对话训练的重排序器会对这些候选问题进行“重排序”。由于重排序模型以原专家系统提供的候选问题为输入，因此无需在所有可能的问题中进行预测，而只需对规模小得多的子集进行重排序，从而大大简化了机器学习任务。我们的模型将先前的对话和可能的问题都作为自由文本条目输入，这意味着即使底层专家系统被替换为其他系统，该方案仍可运行。

我们的贡献如下：

1. 我们提出了一种两步法来选择病史采集问题：首先使用专家系统检索候选问题列表，然后使用机器学习得到的重排序器选出最适合提出的问题。2. 我们提出了一种新颖的“全局重排序器”，将前序对话和候选问题嵌入到一个长字符串中。随后，我们训练长上下文语言模型，同时预测每个问题的相关性。3. 我们针对这项任务，对其他重排序器进行了细致研究，包括双编码器、交叉编码器和自回归重排序器等不同架构。我们考察了不同的长上下文模型，包括 S4（双向和自回归）、Nystrom-former（双向，采用 Nystrom注意力和全注意力的变体）以及 LongT5（自回归）。我们还考察了逐点、成对和列表式损失函数的效果。最后，我们针对上下文长度和初始检索排序进行了一些消融研究。4. 我们发布了 S4 检查点，这些模型分别在维基百科[3] 和 Pubmed PMC开放获取子集[4] 数据集的英文子集上进行了双向和自回归预训练。5. 我们发现，我们的全局重排序器方法优于其他更传统的方法。此外，所有重排序器都显著提升了原始专家系统的性能。我们还发现，基于 S4 的方法虽然不如全注意力Transformer模型，但性能与 Nystrom注意力变换器相当。

医疗保健领域机器学习的 2 个可泛化见解

将深度学习应用于医疗保健的一项主要挑战，是缺乏大规模带标注的数据集。获取大量带标注的数据既昂贵又耗时，因为标注工作需要

[3]https://huggingface.co/datasets/wikipedia[4]
https://www.ncbi.nlm.nih.gov/pmc/tools/textmining/

由经过培训的医疗专业人员完成。近期研究已成功利用大语言模型的发展成果，这些模型是在网络规模数据上训练的。在许多任务中，包括本文讨论的病史采集，这种方法会带来训练与推理之间的差距：用于训练模型的数据无法充分代表模型在推理时接触的数据。在此背景下，我们的方法是先检索一组答案候选项（我们使用专家系统作为基础模型来提供候选项，但也可以由大语言模型提供），再使用基于少量标注数据训练的重排序器，这是一种很有前景的替代方案。正如本文所示，这种重排序器可以利用公共数据源进行训练，然后针对具体任务进行微调。

3 相关工作

我们研究基于医患对话对病史采集推荐进行重排序。这些对话往往很长，超出 Transformer 模型通常的词元长度限制。因此，与本研究相关的文献分为两类。§ 3.1 讨论能够编码完整医患对话的现代神经长程语言模型。§ 3.2 讨论能够利用编码后的对话对病史采集问题重新排序的重排序算法。

3.1 长程Transformer

Transformer [42] 已成为自然语言处理的主流架构。借助自注意力机制，Transformer 可以同时关注序列中的所有词元，因此比卷积网络 [21] 或长短期记忆（LSTM） [16]网络等经典架构更强大。然而，由于其 $O(n^2)$ 复杂度，原始 Transformer 无法高效处理长序列。BERT [10] 和 RoBERTa [24] 等流行的预训练语言模型最多只能处理 512 个词元。研究者已着手降低 Transformer 的计算复杂度，并提出了多种能够处理长文本序列的高效 Transformer，例如 Reformer [20], Linformer [44], Longformer [2], BigBird [49], Performer[8], Nystromformer [48], LongT5 [14], 等。除 Transformer 外，其他方法在处理超长序列方面也展现出良好前景，尤其是状态空间模型[12, 13, 26]。结构化状态空间序列模型（S4） [12] 在长程竞技场基准测试 [39]中显著优于许多长程 Transformer。本文采用 Nystromformer[48] 和 S4 模型[12]作为可能的非自回归骨干网络，并采用 LongT5 作为自回归骨干网络。

3.2 重排序

现代信息检索（IR）或问答（QA）系统通常分为两个阶段 [28, 22]。在第一阶段，给定一个查询，系统会使用高效的方法检索大量文档。在第二阶段，系统会使用计算量较大但更准确的方法，对第一阶段检索到的文档进行重排序。这与我们的问题设定类似：我们使用专家系统检索一组相关的病史采集问题，然后将重排序算法作为第二阶段。相关工作涉及重排序器的架构选择，以及用于训练这些重排序器的损失函数。

第二阶段重排序的架构。主要有三种架构：（1）双编码器 [22, 29, 36, 40] （2）交叉编码器 [22, 29, 17] 和（3）自回归重排序器 [30, 33,27]。在双编码器架构中，查询和候选文档由两个独立的编码器编码为向量表示（尽管这两个编码器可以共享相同的权重[36]）。相关性分数根据两个向量表示之间的距离计算。在交叉编码器中，查询和候选文档会拼接在一起，并一次性输入交叉编码器。在大多数使用场景中，交叉编码器在文档检索和排序方面的表现优于双编码器 [40, 17]；不过，双编码器通常比交叉编码器更高效，因为给定语料库中的所有文档都可以预先计算并存储为稠密嵌入，以供检索，从而避免重复计算 [36]。近期的序列到序列模型（如 T5 [35]）也已应用于自回归重排序。在这种方法中，查询和文档

通常由编码器编码，而解码器要么预测文档是否相关[30]，要么直接生成针对查询的检索文本 [27]。

用于排序的评分函数。 用于排序的评分范式也有多种可能：(1) “逐点”评分，即逐个文档计算查询的相关性（类似于交叉编码器） [29], (2) “成对”评分，即将文档两两比较并相互排序 [29], 以及 (3) “列表式”评分，即同时对候选项列表进行排序 [23]。先前研究表明，成对方法和列表式方法往往优于逐点方法[29, 33, 27, 50, 7]。在本研究中，我们比较了用于病史采集问题重排序的三种方法（“逐点”“成对”和“列表式”），但主要关注列表式方法。

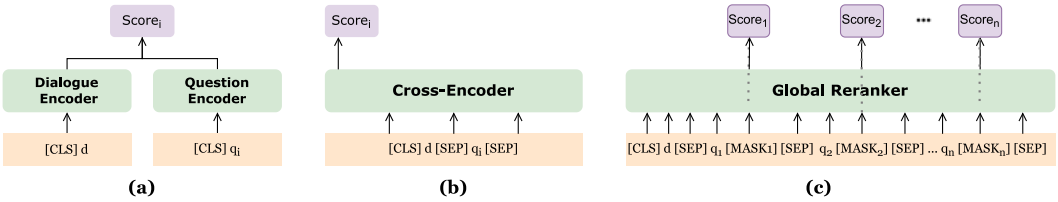


图 1：(a) 双编码器，(b) 交叉编码器，以及 (c) 所提出的全局重排序器。

与我们所提出的全局重排序器的联系。 在本文中，我们提出了一种新颖的重排序方法，称为“全局重排序器”。在该方法中，所有候选文档都会连接成单个输入，然后由长上下文语言模型进行处理。有关双编码器、交叉编码器和全局重排序器的示意性比较，请参见图 1。有关该方法的更详细说明，请参见 § 5。此前，已有研究在双向 [29] 和自回归 [33] 范式中，将文档对连接成单个字符串。由于预训练变换器施加的长度限制，这些模型只能进行成对评分。对于包含两个以上文档的列表式排序，先前的方法侧重于仅对提取的嵌入进行排序 [7, 1], ，这无法建模候选文档之间深层的语义关系。

4 通过重排序弥合训练与推理差距

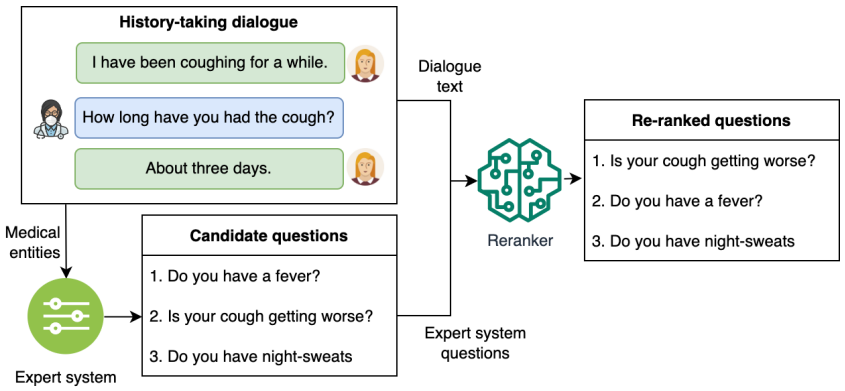


图2：所提出的两阶段病史采集工作流程概览。专家系统根据从对话中提取的相关实体提出候选问题。随后，机器学习得到的深度神经网络重排序器根据对话文本对候选问题进行重排序。

图2展示了我们弥合现有病史采集系统中训练与推理差距的方法概览。我们首先使用专家系统提出相关的病史采集问题，然后使用由完整医患对话提供上下文的深度神经网络，对专家系统的建议进行重排序。

重排序的目标是，给定先前对话上下文 $\boldsymbol{d}$ 和一组 n 候选病史采集问题 $Q = [\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n]$，生成一个由这些问题（可能经过重新排序）组成的新列表 Q'

使相关性较高的问题在序列中排得更靠前。就我们的情况而言，候选问题由内部专家系统生成，而真实标签$\boldsymbol{y} = [y_1, y_2, \ldots, y_n]$，$y_i \in \{0, 1\}$ 表示医生是否提出了某个推荐问题（提出记为 1，未提出记为 0）。医生可能会同时提出多个问题，因此 $\boldsymbol{y}$ 中的多个元素可能取值为 1。有关真实标签如何生成的更多详情，请参阅 § 6.1。最后，在本研究考察的所有模型中，重排序通过为 Q中的每个问题赋予分数 $\boldsymbol{s} = [s_1, s_2, \ldots, s_n]$ 来实现，然后根据 $\boldsymbol{s}$中的分数重新排序，构建 Q'。

5 全局重排序器

我们提出了全局重排序器，这是一种准确且高效的列表式重排序方法。在该方法中（示意图见图 1(c)），病史采集对话与所有候选病史采集问题被拼接成一段文本输入，模型据此同时为所有问题分配排序分数。全局重排序器通过语言模型直接编码所有文本，从而确保对话与候选问题之间，以及所有候选问题彼此之间，都能进行深入的语义交互。

全局重排序器的输入文本由对话上下文和所有候选问题拼接而成：[CLS] $\boldsymbol{d}$ [SEP] $\boldsymbol{q}_1$ [MASK$_1$] [SEP] $\boldsymbol{q}_2$ [MASK$_2$] [SEP] . . . $\boldsymbol{q}_n$ [MASK$_n$] [SEP],其中，[SEP] 词元用于标记候选问题的边界。[MASK$_i$] 词元是前一个问题 $\boldsymbol{q}_i$ 的池化标记。对于每个池化标记 [MASK$_i$]，全局重排序器都会预测一个分数 s_i，表示与 $\boldsymbol{q}_i$ 的相关性。我们还为每个输入词元添加了类型嵌入，以指明其属于对话还是候选问题。专家系统提供的候选问题实际数量为 3 到 40 个。

尽管自注意力机制本身并不假设输入序列具有任何固有顺序，但由于存在位置嵌入，预训练的Transformer模型通常会按顺序对文本进行编码。在当前任务中，我们期望语言模型学习 $\boldsymbol{d}$ 与 $\boldsymbol{q}_i$ 中词之间的顺序关系。通过消融实验（见 § 7.2），我们发现，当模型不受输入问题顺序 $[\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_n]$ 的影响时，性能最佳。为消除位置偏差，我们在每个新问题开始时重置位置嵌入。

我们选取了三种不同的神经网络架构来实现全局排序器，它们都能够处理长文本序列。前两种方法基于 Nystromformer [48],，该模型最初被提出作为一种高效 Transformer。我们分别在启用和停用 Nystrom注意力的情况下对 Nystromformer 进行实验（停用时使用完整注意力）。我们将 Nystromformer 用作“完整注意力”Transformer的基础模型，因为这样可以利用在长文本上训练的预训练 Nystromformer 检查点，同时保留完整注意力的良好性能。我们从试验性实验中了解到，其他高效 Transformer（例如 Longformer [2] ）未能收敛。第三种神经网络架构是状态空间模型 S4，已有研究表明，与许多 Transformer模型相比，它能更有效地处理长序列 [12]。

为训练全局重排序器，我们在学习排序框架 [23] 中比较了逐点式、成对式和列表式方法中的多种损失函数。逐点式基线采用二元交叉熵进行训练。对于成对损失函数，我们测试了 RankNet [3] 和 LambdaRank[4]。我们使用的列表式损失函数包括 ListNet [5], 、ListMLE [47], 、ApproxNDCG [34] 和 NeuralNDCG [32], ，其中后两种直接优化归一化折损累计增益（NDCG）评估指标。

6 实验

6.1 数据

医疗对话数据收集自我们基于文本的医疗服务平台上部分真实的医患互动。在一次典型的互动中，医生会提出一系列病史采集问题，这些问题既可以通过自由文本输入，也可以从推荐问题列表中选取。这些推荐问题由专家系统提供，它构成了我们所提出工作流程的第一阶段。在每个出现推荐问题的对话轮次中，医生选出的

	训练集	Dev	Test
就诊记录数	12105	311	655
样本数	26106	626	1361
对话平均长度	287.2	374.7	288.8
已选问题数	4.0	3.8	4.1
候选问题数	27.9	26.6	28.1
问题平均长度	8.0	8.1	8.0

表 1：不同数据划分的统计信息。文本长度按词数计算。

问题被标记为相关，未选出的问题则被标记为不相关。这形成了一个自然的数据集，其中包含医生标注的选择，我们以此训练重排序器。

该数据集包含 13071 条就诊记录。我们使用内部的对话分段模型筛除了非病史采集对话轮次，方法与 [43] 类似。数据的详细统计信息见表 1。

6.2 评估指标

我们采用了两种常见的排序指标进行评估：归一化折损累计增益（nDCG）[18] 和平均准确率均值（mAP） [22]。mAP 假设相关性为二元值，而 nDCG 既适用于二元相关性，也适用于连续相关性。对于全局重排序器，我们报告了 5 次重复评估的平均指标。每次评估中，都会随机打乱输入全局重排序器的候选问题顺序，以减轻位置偏差。

6.3 基线方法

In a除全局排序器外，我们还实现了三种广泛采用的基线排序方法 s:
bi-e编码器、交叉编码器和自回归重排序器。

双编码器。 在双编码器架构中（见图 1(a)），对话查询和候选问题分别由两个独立的编码器 f_D 和 f_Q进行编码，并通过余弦相似度计算得到的两个向量表示之间的相关性分数。双编码器学习一个嵌入空间，使对话表示接近最相关的问题，同时远离相关性较低的问题。训练目标是通过对比学习最小化 InfoNCE损失函数[31] ，其中从专家系统推荐的候选问题列表中随机抽取 7 个负例。训练过程中，InfoNCE损失的温度参数始终设为 0.05 [11]。

交叉编码器。 在交叉编码器架构中（见图 1(b)），先前对话与一个候选问题拼接在一起。交叉编码器 f_C 使用分类头，根据对话和查询的上下文表示为该候选问题分配相关性分数。我们考察了 Transformer模型和基于S4的模型。对于 Transformer模型，将 [CLS] 标记视为上下文表示。对于双向S4重排序器，我们对最后一层进行平均池化，以获得上下文表示。所有交叉编码器变体均采用二元交叉熵损失进行训练。

自回归重排序器。 我们还考虑了自回归重排序器 [30, 33]。对于 Transformer 基线，我们使用预训练的 LongT5 [14]。查询和文档拼接在一起，形成输入序列：查询： d 文档： q_i 相关：，然后将其输入编码器。随后，解码器会预测相关文档为 true，不相关文档为 false。在推理过程中，会对 true 和 false 词元的 logits 应用 softmax函数，以便在多个查询之间对结果进行归一化。

对于自回归 S4，按照 Long-T5 的方法进行实验时，我们发现其极不稳定且无法收敛，这与文献中的发现类似 [30] ，即它依赖于某些关键词，例如 true/false。因此，我们采用了与交叉编码器相同的设置，只是底层模型采用自回归而非双向结构。在此设置中，拼接后的对话和一个

候选问题被输入 S4 重排序器，最后一层的平均池化结果则通过分类头分类为相关或不相关。

6.4 实现

S4 模型预训练。 S4 模型基于 S4 语言模型的原始实现[12], ，其中 S4 层被用作典型 Transformer 中自注意力层的直接替代。我们实现了 12 层的双向 S4 模型和自回归 S4 模型。为匹配主流预训练 Transformer（如 BERT-base [10]）的参数量，我们将隐藏维度设为 768，并将状态空间模型（SSM）的数量设为 128，每个 SSM 有 64 个状态。

双向 S4 模型和自回归 S4 模型均在大规模文本上进行了预训练。自回归 S4 在维基百科完整英文子集上通过因果语言建模任务进行预训练。第二轮预训练以维基百科预训练检查点为初始化，并使用完整的 Pubmed PMC开放获取子集。双向 S4 模型在相同的数据集上进行预训练，但采用掩码语言建模任务，掩码设置与 BERT [10] 相同。预训练的最大序列长度设为 8192，有效批量大小为 256。

所有模型均使用 AdamW 优化器进行优化，学习率为 1e-4，并通过余弦调度器动态调整学习率，预热步数为 1000。预训练在 8×RTX 3090 GPU 上进行，该 GPU 配备 24GB 内存。当评估损失不再下降时，训练便会停止（所有模型均训练 ∼12k 步）。在这些数据集上预训练的自回归和双向检查点将与本文一同发布。

Transformer 实现。 所有 Transformer 模型均通过 Transformers 软件包实现， [46] 并采用默认维度。自回归模型为 LongT5 [14]，通过 long-t5-tglobal-base 检查点进行初始化。其他 Transformer 则基于 Nys- tromformer [48]，并通过公开检查点 uw-madison/nystromformer-4096 进行初始化。

重排序器训练。 对于全局重排序器，最大输入长度设为 4096，有效批量大小为 32。对于其他模型，有效批量大小为 64，最大长度为 2048，因为这一长度足以涵盖几乎所有数据样本。模型最多训练 5 个 epoch，仅保留验证表现最佳的模型。所有模型均使用 AdamW 优化器 [25] 进行训练，学习率为 5e-5。我们使用预热步数为 1000 的余弦调度器，在训练期间自动调整学习率。所有排序模型均在单张配备 16GB 内存的 V100 GPU 上训练。

7 结果

7.1 主要结果

我们的主要结果总结于表 2。所有神经重排序模型在两项评估指标上均优于基线专家系统，这表明重排序确实提高了更相关的病史采集问题的排名。在神经基线模型中，基于 Transformer 的交叉编码器优于双编码器，这与以往的研究结果一致 [29]。令人意外的是，尽管 LongT5 自回归重排序器拥有更多参数（2.2 亿个参数），其表现仍逊于交叉编码器（∼110M 个参数）。

无论选择哪种损失函数，Transformer 和 S4 架构下表现最佳的都是全局重排序器。在各种损失函数中，逐点二元交叉熵（BCE）的表现最好。我们推测，由于真实相关性分数是二元值而非连续值，当前任务未能充分利用列表式损失函数。

全局重排序器的有效性在于，它不仅关注对话与候选问题之间的语义交互，也关注候选问题彼此之间的语义交互。这使模型能够利用病史采集问题之间的依赖关系（例如共现统计），从而改善排序结果。

模型	Dev		Test	
	nDCG	平均精度均值	nDCG	平均精度均值
专家系统 (基线)	0.592	0.383	0.570	0.349
双编码器 (Transformer)	0.690	0.548	0.677	0.531
交叉编码器				
Transformer	0.718	0.584	0.706	0.566
Nystromformer	0.653	0.496	0.654	0.497
双向 S4（维基百科预训练）	0.648	0.490	0.641	0.481
双向 S4（PubMed 预训练）	0.643	0.483	0.630	0.464
自回归重排序器				
LongT5-base	0.690	0.546	0.678	0.529
自回归 S4（维基百科预训练）	0.658	0.502	0.648	0.490
自回归 S4（PubMed 预训练）	0.654	0.498	0.642	0.484
全局重排序器				
Transformer				
+逐点损失：二元交叉熵	**0.744**	**0.618**	**0.743**	**0.618**
+成对损失：RankNet	0.739	0.612	0.735	0.603
+成对损失：LambdaLoss	0.739	0 616	0.739	0.612
+列表损失：ListNet	0.737	0.609	0.740	0.610
+列表损失：ListMLE	0.727	0.597	0.721	0.587
+列表损失：ApproxNDCG	0.701	0.555	0.697	0.550
+列表损失：NeuralNDCG	0.742	0.617	0.741	0.612
Nystromformer				
+逐点损失：二元交叉熵	0.684	0.537	0.678	0.530
双向 S4（维基百科预训练）				
+逐点损失：二元交叉熵	0.667	0.516	0.663	0.510
双向 S4（PubMed 预训练）				
+逐点损失：二元交叉熵	0.697	0.556	0.670	0.518

表 2：重排序实验结果。

还值得注意的是，尽管 S4 在某些长序列处理基准测试 [12], 中表现出色，但在当前任务中仍落后于 Transformer模型。一个可能的原因是，这里的 S4 模型仅在相对少量的文本上进行过预训练，而 Transformer模型则在海量文本上进行过预训练。此外，我们任务中的文本序列长度从几百词到约三千词不等，对于 S4 来说，这或许还不足以充分发挥其潜力。

7.2 消融分析

我们对全局重排序器进行了消融分析，以评估对话上下文长度、类型嵌入的影响，以及打乱候选问题顺序的影响。结果见表 3。

上下文长度消融。 在进行上下文长度消融时，仅考虑对话中最后的 N 个词元（完整模型使用 4096 个词元，消融设置分别使用 3072、2048 和 1024 个词元）。尽管大多数文本序列都短于 1000 个词元，但截断文本仍会降低某些特别长（超过 1024 个词元）的文本序列的测试性能，因为这可能会移除一些重要信息。总体而言，全局重排序器受益于更多对话上下文，但当上下文扩展到超过 2048 个词元后，这种收益似乎会逐渐减弱。

位置嵌入和类型嵌入的作用。 我们发现，移除类型嵌入（这类嵌入经过学习，用于区分词元来自对话还是候选问题）对测试性能几乎没有影响。我们为每个候选问题重置了位置嵌入

在输入序列中的位置，因为这可能有助于模型学会不受问题顺序的影响。我们训练了一个对输入序列使用连续位置嵌入的模型。结果表明，位置嵌入在训练全局重排序器时作用有限。

打乱顺序的影响。我们测试了输入候选问题的顺序对排列不变性的重要性。候选问题列表 $[q_1, q_2, \ldots, q_n]$ 与先前对话拼接后，作为模型的输入。我们发现，尽管专家系统应按照相关性顺序提出问题，但使用打乱顺序的数据训练模型时，性能显著更高。我们认为，这会促使模型学会对问题重新排序，而不依赖候选问题的原始顺序。

消融实验	Dev		Test	
	nDCG	平均精度均值	nDCG	平均精度均值
- 最大长度：4096（完整）	0.744	0.618	0.743	0.618
- 最大长度：3072	0.741	0.614	0.739	0.611
- 最大长度：2048	0.746	0.622	0.747	0.622
- 最大长度：1024	0.747	0.623	0.737	0.609
- 无类型嵌入	0.733	0.610	0.732	0.607
- 顺序位置嵌入	0.749	0.625	0.741	0.613
- 不进行问题随机打乱	0.515	0.313	0.523	0.319

表 3：全局重排序器消融研究结果

8 讨论

在这项工作中，我们致力于解决自动化病史采集中的一个重要问题：缩小训练-推理差距。我们的方案受到现代神经信息检索系统的启发，分为两个阶段：(1) 使用专家系统推荐一系列候选问题（从数千个可能的问题中筛选），(2) 训练一个机器学习重排序模型，根据医患对话的自由文本对专家系统推荐的问题重新排序。

为执行重排序（第二阶段），我们提出了一种新方法，称为“全局重排序器”，并将其与现有的神经基线模型进行比较。我们还探索了多种语言模型骨干架构，包括各种 Transformer 模型和结构化状态空间（S4）模型[5]。我们发现，尽管所有神经重排序模型的表现都优于原始专家系统，但采用全注意力变换器作为骨干架构的全局重排序器表现最佳，其 nDCG 比第一阶段推荐结果提高了 30%，平均精度均值（mAP）提高了 77%。

虽然我们的结果直接表明，在病史采集专家系统之上训练重排序模型是有效的，但我们认为，这种方法也适用于其他决策支持系统。该方法在以下条件下会发挥作用：(1) 存在训练与推理之间存在差距的评分系统；(2) 可能预测的范围非常大，因此需要大量数据才能从头开始进行机器学习。除了病史采集之外，我们认为医疗诊断预测也满足这些条件。现有许多源自专家系统的诊断模型，而由于可能的诊断范围非常大，从头开始训练诊断模型可能很困难。重排序可用于缩小现成诊断专家系统与实际患者群体结局之间的差距。

局限性 这项工作仍存在若干局限。尽管我们提出的全局重排序器整体表现优于其他排序模型，但其计算效率仍然不高，因为在处理长序列时，完整注意力机制的 Transformer 模型具有二次计算复杂度。随着对话变长或候选问题数量增加，这将成为更加严重的瓶颈。其次，全局重排序器只能从语言中学习病史采集问题与对话上下文之间的关联，却不具备内在的医学

[5]作为本次发表的一部分，我们发布了在英文维基百科和 Pubmed PMC开放获取子集上预训练的双向和自回归 S4 检查点。

知识。为此类模型补充真实的医学知识至关重要，这样模型才能做出更有依据的决策，也不会对低频长尾病史采集问题产生偏见。未来，我们计划探索更有效的方法来编码长文本上下文，并将知识注入语言模型。

8.1 伦理

本项工作是作为一项质量改进活动开展的，该活动符合 45 CFR §46.104 (d)(4)(iii) 的定义——即无需为“医疗保健运营”目的取得同意的二次研究。

参考文献

[1] Qingyao Ai、Xuanhui Wang、Sebastian Bruch、Nadav Golbandi、Michael Bendersky 和 Marc Najork。使用深度神经网络学习组级多变量评分函数。载于*2019*年美国计算机协会 *SIGIR* 信息检索理论国际会议论文集，第85–92页，2019年。[2] Iz Beltagy、Matthew E Peters 和 Arman Cohan。Longformer：长文档 Transformer。*arXiv*预印本 *arXiv:2004.05150*，2020年。[3] Chris Burges、Tal Shaked、Erin Renshaw、Ari Lazier、Matt Deeds、Nicole Hamilton 和 Greg Hullender。使用梯度下降学习排序。载于第*22*届国际机器学习会议论文集，第89–96页，2005年。[4] Christopher Burges、Robert Ragno 和 Quoc Le。使用非光滑代价函数学习排序。神经信息处理系统进展，第19卷，2006年。[5] Zhe Cao、Tao Qin、Tie-Yan Liu、Ming-Feng Tsai 和 Hang Li。学习排序：从成对方法到列表式方法。载于第*24*届国际机器学习会议论文集，第129–136页，2007年。[6] Shu Chen、Zeqian Ju、Xiangyu Dong、Hongchao Fang、Sicheng Wang、Yue Yang、Jiaqi Zeng、Ruisi Zhang、Ruoyu Zhang、Meng Zhou、Penghui Zhu 和 Pengtao Xie。Meddialog：大规模医疗对话数据集。*CoRR*，2020年。[7] Xiaoyang Chen、Kai Hui、Ben He、Xianpei Han、Le Sun 和 Zheng Ye。Co-BERT：一种结合局部上下文和查询特定上下文的上下文感知 BERT 检索模型。*arXiv*预印本 *arXiv:2104.08523*，2021年。[8] Krzysztof Marcin Choromanski、Valerii Likhosherstov、David Dohan、Xingyou Song、Andreea Gane、Tamas Sarlos、Peter Hawkins、Jared Quincy Davis、Afroz Mohiuddin、Lukasz Kaiser 等。重新思考 Performer 中的注意力机制。载于国际学习表征会议，2020年。[9] Rhys Compton、Ilya Valmianski、Li Deng、Costa Huang、Namit Katariya、Xavier Amatriain 和 Anitha Kannan。Medcod：一种医学准确、富有情感、多样且可控的对话系统。载于健康机器学习会议，第110–129页。PMLR，2021年。[10] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。BERT：用于语言理解的深度双向 Transformer 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（长篇与短篇论文），第4171–4186页，2019年6月。[11] Tianyu Gao、Xingcheng Yao 和 Danqi Chen。SimCSE：句子嵌入的简单对比学习。载于*2021*年自然语言处理实证方法会议论文集，第6894–6910页，2021年11月。[12] Albert Gu、Karan Goel 和 Christopher Re。利用结构化状态空间高效建模长序列。载于国际学习表征会议，2021年。[13] Albert Gu、Ankit Gupta、Karan Goel 和 Christopher Ré。关于对角状态空间模型的参数化与初始化。*arXiv*预印本 *arXiv:2206.11893*，2022年。

[14] Mandy Guo、Joshua Ainslie、David Uthus、Santiago Ontanon、Jianmo Ni、Yun-Hsuan Sung 和 Yinfei Yang。LongT5：面向长序列的高效文本到文本 Transformer。*arXiv*预印本 *arXiv:2112.07916*，2021年。[15] John R Hampton、MJ Harrison、John R Mitchell、Jane S Prichard 和 Carol Seymour。病史采集、体格检查和实验室检查对门诊患者诊断与管理的相对贡献。英国医学杂志，2(5969):486–489，1975年。[16] Sepp Hochreiter 和 Jürgen Schmidhuber。长短期记忆。神经计算，9(8): 1735–1780，1997年。[17] Samuel Humeau、Kurt Shuster、Marie-Anne Lachaux 和 Jason Weston。多重编码器：用于快速、准确多句评分的架构与预训练策略。载于国际学习表征会议，2019年。[18] Kalervo Järvelin 和 Jaana Kekäläinen。用于检索高度相关文档的信息检索评估方法。SIGIR '00。美国计算机协会，2000年。ISBN 1581132263。[19] Hao-Cheng Kao、Kai-Fu Tang 和 Edward Chang。利用分层强化学习进行上下文感知症状检查与疾病诊断。载于*AAAI*人工智能大会论文集，第32卷，2018年。[20] Nikita Kitaev、Lukasz Kaiser 和 Anselm Levskaya。Reformer：高效的 Transformer。载于国际学习表征会议，2019年。[21] Y. LeCun、B. Boser、J. S. Denker、D. Henderson、R. E. Howard、W. Hubbard 和 L. D. Jackel。反向传播在手写邮政编码识别中的应用。神经计算，1(4): 541–551，1989年。doi: 10.1162/neco.1989.1.4.541。[22] Jimmy Lin、Rodrigo Nogueira 和 Andrew Yates。用于文本排序的预训练 Transformer：从 BERT 到更多模型。人类语言技术综合讲座，14(4):1–325，2021年。[23] Tie-Yan Liu 等。面向信息检索的学习排序。信息检索基础与趋势，3(3):225–331，2009年。[24] Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Mandar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。RoBERTa：一种经过稳健优化的 BERT 预训练方法。*arXiv*预印本 *arXiv:1907.11692*，2019年。[25] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。载于国际学习表征会议，2018年。[26] Harsh Mehta、Ankit Gupta、Ashok Cutkosky 和 Behnam Neyshabur。通过门控状态空间实现长程语言建模。*arXiv*预印本 *arXiv:2206.13947*，2022年。[27] Sewon Min、Kenton Lee、Ming-Wei Chang、Kristina Toutanova 和 Hannaneh Hajishirzi。面向多样化多答案检索的联合段落排序。载于*2021*年自然语言处理实证方法会议论文集，第6997–7008页，2021年11月。[28] Rodrigo Nogueira 和 Kyunghyun Cho。使用 BERT 进行段落重排序。*arXiv*预印本 *arXiv:1901.04085*，2019年。[29] Rodrigo Nogueira、Wei Yang、Kyunghyun Cho 和 Jimmy Lin。使用 BERT 进行多阶段文档排序。*arXiv*预印本 *arXiv:1910.14424*，2019年。[30] Rodrigo Nogueira、Zhiying Jiang、Ronak Pradeep 和 Jimmy Lin。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论文集：*EMNLP 2020*，第708–718页，2020年11月。[31] Aaron van den Oord、Yazhe Li 和 Oriol Vinyals。利用对比预测编码进行表征学习。*arXiv*预印本 *arXiv:1807.03748*，2018年。[32] Przemyslaw Pobrotyn 和 Radoslaw Bialobrzeski。NeuralNDCG：通过排序的可微松弛直接优化排序指标。*arXiv*，abs/2102.07831，2021年。

[33] Ronak Pradeep、Rodrigo Nogueira 和 Jimmy Lin。利用预训练序列到序列模型进行文本排序的 expando-mono-duo 设计模式。*arXiv* 预印本 *arXiv:2101.05667*，2021年。[34] Tao Qin、Tie-Yan Liu 和 Hang Li。直接优化信息检索指标的通用近似框架。信息检索，13(4):375–397，2010年。[35] Colin Raffel、Noam Shazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li、Peter J Liu 等。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):1–67，2020年。[36] Nils Reimers 和 Iryna Gurevych。Sentence-BERT：使用孪生 BERT 网络的句子嵌入。载于*2019*年自然语言处理实证方法会议与第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集，第3982–3992页，香港，中国，2019年11月。计算语言学协会。doi: 10.18653/v1/D19-1410。[37] Hagen Soltau、Mingqiu Wang、Izhak Shafran 和 Laurent El Shafey。理解医疗对话：丰富的转录、置信度分数与信息抽取。*CoRR*，2021年。[38] Kai-Fu Tang、Hao-Cheng Kao、Chun-Nan Chou 和 Edward Y Chang。问询与诊断：采用深度强化学习的神经症状检查集成系统。载于 *NIPS* 深度强化学习研讨会，2016年。[39] Yi Tay、Mostafa Dehghani、Samira Abnar、Yikang Shen、Dara Bahri、Philip Pham、Jinfeng Rao、Liu Yang、Sebastian Ruder 和 Donald Metzler。长程竞技场：高效 Transformer 的基准测试。载于国际学习表征会议，2020年。[40] Nandan Thakur、Nils Reimers、Johannes Daxenberger 和 Iryna Gurevych。增强型 SBERT：用于改进成对句子评分任务中双编码器的数据增强方法。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，第296–310页，2021年。[41] Ilya Valmianski、Nave Frost、Navdeep Sood、Yang Wang、Baodong Liu、James J. Zhu、Sunil Karumuri、Ian M. Finn 和 Daniel S. Zisook。SmartTriage：用于个性化患者数据采集、文档生成和决策支持的系统。载于健康机器学习会议论文集，第158卷机器学习研究论文集，第75–96页。PMLR，2021年12月4日。[42] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。注意力机制就是你所需的一切。神经信息处理系统进展，30，2017年。[43] Mengqian Wang、Ilya Valmianski、Xavier Amatriain 和 Anitha Kannan。学习医疗对话中的功能区段：迭代式伪标签与人在回路方法，2022年。[44] Sinong Wang、Belinda Z Li、Madian Khabsa、Han Fang 和 Hao Ma。Linformer：具有线性复杂度的自注意力机制。*arXiv* 预印本 *arXiv:2006.04768*，2020年。[45] DL Watson。健康访谈回答与医疗记录的比较。生命与健康统计。第*1*辑，项目与数据收集程序，(3):1–74，1965年。[46] Thomas Wolf、Lysandre Debut、Victor Sanh、Julien Chaumond、Clement Delangue、Anthony Moi、Pierric Cistac、Tim Rault、Rémi Louf、Morgan Funtowicz 等。Transformer：自然语言处理的先进技术。载于*2020*年自然语言处理实证方法会议论文集：系统演示，第38–45页，2020年。[47] Fen Xia、Tie-Yan Liu、Jue Wang、Wensheng Zhang 和 Hang Li。学习排序的列表式方法：理论与算法。载于第*25*届国际机器学习会议论文集，第1192–1199页，2008年。

[48] Yunyang Xiong、Zhanpeng Zeng、Rudrasis Chakraborty、Mingxing Tan、Glenn Fung、Yin Li 和 Vikas Singh。Nyströmformer：一种近似自注意力机制的基于奈奎斯特的算法。载于*AAAI*人工智能大会论文集，第35卷，第14138–14148页，2021年。
[49] Manzil Zaheer、Guru Guruganesh、Kumar Avinava Dubey、Joshua Ainslie、Chris Alberti、Santiago Ontanon、Philip Pham、Anirudh Ravula、Qifan Wang、Li Yang 等。Big Bird：面向更长序列的 Transformer模型。神经信息处理系统进展，33: 17283–17297，2020年。[50] George Zerveas、Navid Rekabsaz、Daniel Cohen 和 Carsten Eickhoff。Coder：通过上下文化文档嵌入重排序来改进检索的高效框架。*arXiv* 预印本 *arXiv:2112.08766*，2021年。