DistillER：大语言模型实体解析中的知识蒸馏

Alexandros Zeakis 信息学与电信系，雅典国立卡波迪斯特里亚大学，希腊雅典，alzeakis@di.uoa.gr；雅典娜研究中心，希腊雅典，azeakis@athenarc.gr

George Papadakis 信息学与电信系，雅典国立卡波迪斯特里亚大学，希腊雅典，gpapadis@di.uoa.gr

Dimitrios Skoutas 雅典娜研究中心，希腊雅典，dskoutas@athenarc.gr

Manolis Koubarakis 信息学与电信系，雅典国立卡波迪斯特里亚大学，希腊雅典，koubarak@di.uoa.gr

arXiv:2602.05452v1 [cs.DB] 5 Feb 2026

摘要

实体解析（ER）的最新进展利用了大语言模型（LLM），取得了出色的性能，但代价是需要大量计算资源或承担高昂的财务开销。现有基于 LLM 的实体解析方法，要么在无监督设置下运行并依赖规模庞大、成本高昂的模型，要么在监督设置下运行并需要真实标注，因此在时间效率与效果之间留下了关键缺口。为使 LLM 驱动的实体解析更具实用性，我们研究了知识蒸馏（KD），将大型高效模型（教师模型）的知识迁移到更小、更高效的模型（学生模型）中，而无需金标准标签。我们提出 DistillER，这是首个从三个维度系统弥合这一缺口的框架：（i）数据选择，研究如何识别信息丰富的数据子集；（ii）知识 elicitation，比较大语言模型与小语言模型（SLM）中的单教师和多教师设置；（iii）蒸馏算法，评估监督微调与强化学习方法。实验表明，在 LLM 教师模型生成的噪声标签上对学生模型进行监督微调，始终优于其他知识蒸馏策略，同时还能生成高质量的解释。最后，我们将 DistillER 与基于大语言模型和小语言模型的成熟监督及无监督实体解析方法进行基准比较，结果表明其效果和效率均有显著提升。

1 引言

实体解析（ER）旨在检测指向同一现实世界实体的不同记录 [5]。这是数据集成中的一项基本问题，应用范围从问答到分析 [6, 8]。

在实践中，实体解析解决方案通常包含两个步骤 [6]：（i）阻塞，通过生成可能匹配的候选记录对来缩小二次比较空间；（ii）匹配，这是一项更细粒度的决策任务，用于确定哪些候选对真正对应于同一实体。前一步骤已得到广泛研究 [16, 32, 34],，最先进的方法已经取得了很高的有效性。

© 2026 版权所有者/作者所有。本文可根据知识共享许可协议 CC-by-nc-nd 4.0 的条款进行分发。

关于匹配，先前的工作采用了基于预训练小语言模型（小语言模型） [9, 28], 的无监督方法，从而避免使用标注数据，同时也采用了监督方法 [1, 11, 14, 24], ，在标注样本上训练二元分类器。

随着大语言模型（LLM）的兴起，近期研究采用不同的提示策略将其用于匹配。由此形成了两种主流设置 [27], ，如图 1 所示。

（1）在 MATCH 设置中，模型以一个候选对作为输入，并回答二元问题："这对记录是否匹配？"。（2）SELECT 设置向模型提供一条查询记录和多个候选项，并要求其"选出正确匹配项"。

MATCH 已在无监督场景 [17, 18] 和监督场景 [23],中得到研究；研究表明，微调后的 GPT-3.5 表现优于预训练的 GPT-4。SELECT 仅在使用 GPT-3.5 [27], 等大型模型时才能在无监督模式下取得出色表现；集成中型模型虽能达到有竞争力的准确率，但计算和时间成本更高 [33]。在监督设置中，微调后的中型大语言模型（约 8B 个参数）甚至超越了预训练的 GPT-3.5，凸显了在 SELECT 中进行微调所带来的效率提升 [33]。尽管取得了这些进展，当前基于大语言模型的实体解析方法仍局限于两种设置：（i）依赖超大型或专有模型、计算和经济成本高昂的无监督方法，或（ii）需要人工标注数据的监督方法。目前尚无系统性方案能在不依赖真实标签的情况下，将大语言模型的有效性迁移到高效模型中。

然而，微调需要标注数据，而这些数据并不总是可用；与此同时，大型模型的计算成本高昂，运行速度也较慢。在实体解析领域的文献中，这一常见解决方案一直被忽视，即知识蒸馏（KD） [3],——一种模型压缩技术，其中较小的学生模型 学习模仿较大的教师模型的行为。早期应用主要致力于缩减语言模型的规模，DistilBERT 便是其中的典型代表 [20]。随着大语言模型的出现，知识蒸馏已超越简单的架构压缩，开始应对将知识从教师传递给学生这一更广泛的挑战，不仅捕捉预测结果，还捕捉推理模式 [30]。DeepSeek-r1 模型便是一个成功案例 [7]。然而，将知识蒸馏应用于匹配并非易事，这引发了一些关键问题：

应当蒸馏哪些数据、如何 eliciting 教师知识、哪些算法最为有效，等等。

在这项工作中，我们提出了 DistillER，这是首个将 KD 应用于 ER，尤其是基于大语言模型的匹配，且无需真实标签的框架。尽管 KD 已在自然语言处理中得到广泛探索，但其在匹配中的应用仍未得到充分研究。DistillER 通过引入采用实体解析专用蒸馏方法的框架，填补了这一空白。它包含三个主要组成部分：（i）数据选择，用于确定蒸馏过程中使用的训练数据。（ii）知识提取，取决于教师模型的选择以及所给出的指令，例如模型应只提供一个答案，还是同时给出解释。此处的主要过程是教师模型生成标注数据，以指导学生模型。（iii）蒸馏算法，包括选择学生模型并制定训练策略。训练策略可以采用监督微调（SFT）或强化学习（RL）方法。我们通过评估 DistillER 的主要实现形式，将这一框架付诸实践，并在监督和无监督设置下，将其与采用大语言模型和小语言模型的最先进实体解析方法进行比较。全面的实验研究表明，DistillER 持续取得提升，并实现了总体平均性能最佳。

简而言之，我们的贡献如下：

- 我们提出 DistillER，这是首个面向大语言模型实体解析的知识蒸馏框架。
- 我们设计并评估了数据选择策略，以支持有效的知识迁移。
- 我们定义了一个知识 elicitation 流程，该流程取决于教师模型的选择和所提供的指令，例如模型是只生成单个答案，还是同时给出解释。
- 我们考察了主要的蒸馏算法，重点关注学生模型的选择和训练策略，包括 SFT 和强化学习方法。
- 我们开展了全面的实验研究，包括组件级分析，并在真实世界数据集上与最先进的实体解析方法进行基准测试。

本文其余部分的结构如下：第2节回顾相关工作，第3节定义我们问题的关键概念。第4节概述 DistillER 框架及其关键组件。第 5 节介绍我们的评估，而

第 6 节总结了本文。

2 相关工作

我们将相关工作分为三个主要领域：实体匹配（包括基于小语言模型和大语言模型的方法）、知识蒸馏（涵盖大语言模型的一般方法及其在实体解析中的新兴应用），以及介绍实用优化策略的强化学习。

2.1 实体匹配

使用小语言模型进行匹配。在无监督设置下，ZeroER [28] 通过多种函数的相似度分数来表示候选对，并使用高斯混合模型对所得向量建模，同时采用自适应正则化和传递性来提升准确率。CollaborEM [9]采用自监督策略，将自动标签生成与协同期望最大化训练相结合，无需人工标注即可捕捉图结构和句子级信号。在有监督设置下，Su- dowoodo [26] 采用对比表征学习来



图1：MATCH 与 SELECT 提示示例。

生成具备相似度感知能力的表示，从而只需少量标签即可有效微调。HierGAT [31] 提出了一种分层图注意力 Transformer，可联合建模属性关系和匹配决策，从而丰富上下文嵌入。最后，Unicorn [24]为二元数据匹配任务提出了一种统一架构，采用通用编码器和混合专家分类器，支持跨数据集知识共享，甚至支持零样本预测。我们在第 5 节中通过实验将这些方法与 DistillER 进行比较。

基于大语言模型的匹配。基于大语言模型的匹配的一个关键方面是提示工程，其范围涵盖从零样本到少样本提示。[17], 主要针对预训练模型，对 MATCH 提示进行了广泛研究；[23]则在此基础上纳入了微调，重点关注跨领域泛化和训练数据集规模。

在 SELECT 提示中，单次查询评估多个候选项时，使用 GPT-3.5 可取得良好效果（见[27] ）；采用结合多个 8B 模型并进行投票的基于委员会的方法也能取得良好效果（见 [33]）。在后一种情况下，对 8B 模型进行微调比使用预训练的 GPT-3.5 效果更好，从而减轻了位置偏差。

其他研究包括 [29], , 该研究探讨了扩充训练数据的策略；以及 [35], , 该研究表明，采用微调的迁移学习可以胜过零样本基线。

我们在第 5 节中，将 DistillER 与小型开源模型结合，并与这些最先进的基于大语言模型的匹配算法进行实验比较。

2.2 使用大语言模型进行知识蒸馏

现有关于大语言模型知识蒸馏的研究可分为四个主要维度 [30]：(i) 知识 elicitation，即通过查询教师模型来生成或增强训练数据；(ii) 蒸馏算法，涵盖从标准监督微调到基于散度、强化学习或基于排序的方法；(iii) 学生模型增强，旨在提升自然语言理解、自然语言生成和信息检索等下游任务的表现；(iv) 垂直蒸馏，即将知识蒸馏应用于法律、医疗保健和金融等领域的特定领域应用。最近，研究重点已从通用压缩转向特定任务技能迁移，并着重关注推理、对齐和领域适应方法。

在实体解析领域，唯一相关的工作是 [25]，该工作使用 GPT-3.5 为实体对生成解释，并将这些解释与标签结合起来，对较小的模型（Flan-T5）进行微调。我们的框架则将基于指令的蒸馏扩展到成对设置之外。在实体对齐（EA）领域，[4] 提出了 LLM4EA，这是一个结合大语言模型标注器、主动学习和无监督标签精炼的框架，可大规模生成高质量标签，用于微调。在实体链接领域，[12] 提出了一种多视图增强蒸馏框架，将实体拆分为多个视图，并在保留全局结构的同时，将交叉编码器中的细粒度知识蒸馏到双编码器中。最后，[13] 提出了一个用于实体匹配主动学习的基准，研究表明，谨慎地组合分类器与选择策略，能够以更少的标签和更高的效率媲美全监督。总体而言，尽管已有一些蒸馏工作涉及实体对齐和实体链接，但对实体解析的系统性探索仍然有限，尤其缺少对监督微调或强化学习等核心范式的研究。

2.3 强化学习

强化学习已成为使大语言模型与偏好数据对齐的核心范式。一种重要方法是基于人类反馈的强化学习（RLHF），其中人工标注者提供偏好标签，以训练引导优化的奖励模型 [21]。相比之下，基于人工智能反馈的强化学习（RLAIF）通过使用另一个人工智能模型生成偏好来消除对人工标注者的依赖，再由该模型监督学生模型。近端策略优化（PPO）[21] 是 RLHF 与 RLAIF 的标准强化学习算法，它使用带正则化的裁剪替代目标，以确保策略更新稳定。近期的替代方法包括广义奖励策略优化（GRPO）[22]，它用基于组的优势估计取代价值函数，以便在推理密集型任务中实现更高效的更新；以及直接偏好优化（DPO）[19]，它完全绕过奖励建模，直接使模型与成对偏好对齐。我们在 DistillER 框架中采用这些强化学习策略，将 GRPO 和 DPO 集成为蒸馏的替代优化方法。

3 问题定义

现在，我们定义本文所解决任务的核心概念。

定义 1（实体）。我们将实体 e 定义为属性（即名值对）a_i 的集合，并记作 $e = \{a_1, a_2, .., a_n\}$。

定义 2（元组）。给定两个实体集合，$\mathcal{D}_1$ 和 $\mathcal{D}_2$，考虑一个查询实体 $q \in \mathcal{D}_1$ 和一个候选实体集合 $C \subseteq \mathcal{D}_2$，后者通常通过来自另一数据源的阻塞步骤获得。我们将元组 $t_{q,C}$ 定义为由查询实体 q 及其对应的候选集 c 组成的对，即 $t_{q,C} = (q, C)$。

当阻塞依赖于使用预训练语言模型进行近似最近邻搜索时（如 [32, 34]所示），会自然地产生元组。因此，我们可以通过计算查询实体 q 与每个候选匹配项 c 的嵌入向量之间的余弦相似度，来估计阻塞相似度 $sim(c)$。

在此背景下，我们可以将实体匹配（*EM*）形式化定义如下：判断两个可能来自不同来源的实体是否指向同一个现实世界对象。

定义 3（知识增强实体匹配）。令 $\mathcal{T}$ 表示由元组组成的集合 $t_{q,C}$，并令 K 表示针对这些元组生成的知识（例如标签、特征或模型预测）。知识增强实体匹配 问题旨在训练一个模型 M，使其基于 $(\mathcal{T}, K)$ 预测来自不同数据源的实体是否对应同一个现实世界对象。

4 框架

现在，我们介绍 DistillER 框架。该框架利用自监督学习，以端到端的方式实现知识增强实体匹配。其核心是由教师模型生成自训练学生模型所需的知识，同时，框架还包含一种有效策略，用于将数据划分为训练集和测试集。如图 2 所示，DistillER 包含三个主要阶段：（i）数据选择，用于识别适合知识蒸馏的代表性样例；（ii）知识 *elicitation*，从教师模型中提取并整理知识，同时考虑不同的模型类型和指令；（iii）蒸馏算法，将提取的知识迁移至学生模型，重点关注学生架构的选择，以及用于有效整合蒸馏知识的训练策略。下文将详细介绍这些阶段。

4.1 数据选择

本步骤接收来自阻塞的一组元组作为输入，并将其划分为两个互不相交的子集：训练集和测试集。前者用于从教师模型生成知识，后者用于评估学生模型的性能。训练集由两类元组组成：(i) 正元组，其中查询实体的实际匹配包含在候选项中；(ii) 负元组，其中候选项缺少与查询实体的实际匹配。

在数据选择中，我们应对两个主要挑战：(i) 如何表示每个元组，以及 (ii) 如何区分正元组和负元组。对于前者，我们使用阻塞生成的相似度分数，这些分数既可以聚合为每个元组的单一数值，也可以通过将分数范围划分为分箱来表示为向量。对于后者，由于我们假设没有真实标签，我们基于前述元组表示采用两种策略：(i) 排序策略按聚合分数对元组排序，将排名靠前的元组视为正例，排名靠后的视为负例。(ii) 聚类策略根据元组的向量表示将其分为两个簇——一个对应正例，另一个对应负例。在采用任一策略之后，我们随机选择 p 个正元组和 n 个负元组构成训练集，其余元组则构成测试集。

4.1.1 排序。此方法根据计算得出的分数对所有元组进行排序，然后分别将排名最高的 p 个和排名最低的 n 个指定为正例和负例。我们为元组 t 定义了两种可能的评分函数：

• **最大分数：** $r\text{max}_t = \max_{c \in t} \text{sim}(c)$，其中 $\text{sim}(c)$ 表示候选项 c 的相似度分数，该分数由阻塞得出。• **前二分数：** $r\text{top-}^2_t = \frac{\text{sim}(c_1)+\text{sim}(c_2)}{2}$，其中 c_1 ~~和 c_2 是 t 中~~相似度分数最高的两个候选项。

元组中的最大分数能有力地表明候选项中是否包含正确匹配。与至少有一个高分的元组相比，仅包含低相似度分数的元组不太可能是正例。为降低离群值的影响，我们还考虑前二分数，即两个最高相似度分数的平均值。

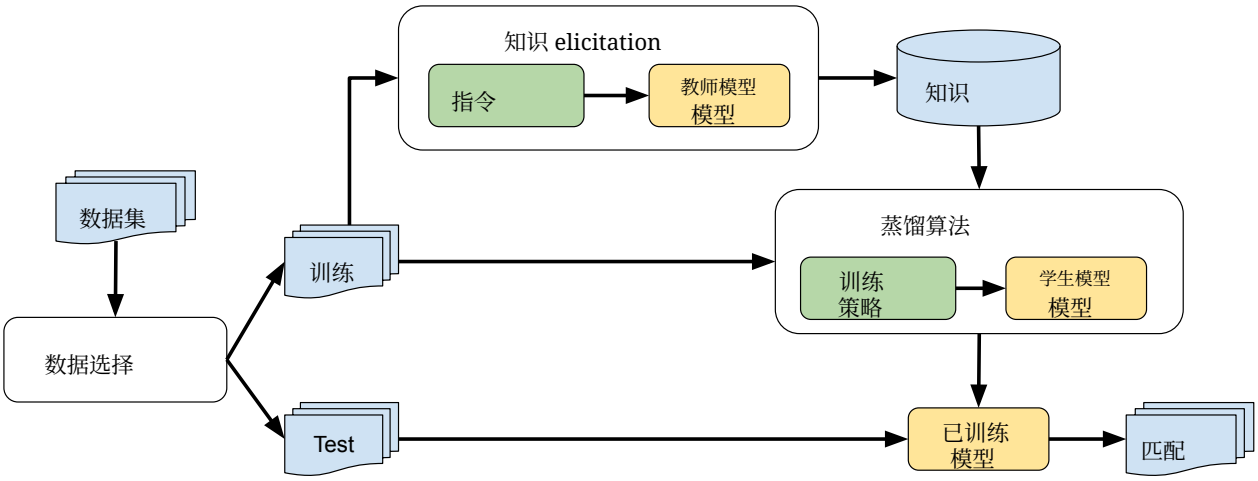


图 2：DistillER 框架，突出展示了三个关键组成部分。

示例。为说明这两个函数之间的差异，假设图*1*的 *SELECT* 提示中的元组 t_1 从阻塞获得候选实体 e_1-e_5的以下相似度分数：*{0.74, 0.95, 0.90, 0.70, 0.76}*。再假设另一个元组 t_2 有五个候选实体，其分数如下：*{ 0 42, 0 37, ...}*。最大分数分别为 *0 24, 0 39, 0 99* 和，而前二分数分别为 $r_{t_1}^{max} = 0.95$ $r_{t_2}^{max} = 0.99$ 和 $r_{t_1}^{top\text{-}2} = 0.925$ 和 $r_{t_2}^{top\text{-}2} = 0.705$。

4.1.2 聚类。 第二种策略将数据选择视为一个包含两个簇的聚类问题。与类似的主动学习方法（例如 [13]）不同，后者将此任务建模为一个分类问题；由于缺少已标注实例，我们以无监督的方式解决该问题。我们将每个元组的相似度分数转换为具有固定数量 b 个分箱的直方图向量。每个分箱统计落在其范围内的分数，从而为所有元组提供均衡的表示。随后，可以使用标准算法对生成的向量进行聚类，并将簇的数量设为两个。接着，我们将包含相似度分数最高的元组的簇标记为正例，而另一个簇则视为负例。最后，我们分别随机选择 p 和 n 个元组。

示例。 使用上一个示例中 t_1 和 t_2 的相似度分数，我们将每个元组映射为一个 *10* 维直方图向量，在 $[0, 1]$ 上按宽度为 *0.1* 的区间分箱。对于 t_1，有三个分数落入 $[0.7, 0.8)$ 区间，两个落入 $[0.9, 1.0]$，得到向量 $[0, 0, 0, 0, 0, 0, 3, 0, 2]$。对于 t_2，一个分数落入 $[0.2, 0.3)$，两个落入 $[0.3, 0.4)$，一个落入 $[0.4, 0.5)$，还有一个落入 $[0.9, 1.0]$，得到向量 $[0, 0, 1, 2, 1, 0, 0, 0, 0, 1]$。

4.2 知识 elicitation

此步骤侧重于从教师模型中提取知识 K。它接收一组元组 T 作为输入，并以标签、特征或模型预测的形式，输出关于这些元组生成的知识 K。DistillER 根据两个标准区分可能的方法：(i) 用作教师的模型类型；(ii) 提供给教师模型的明确指令。下面将介绍 DistillER 针对每项标准提供的方法。

4.2.1 教师模型。 教师模型负责为给定的元组 $t_{q,\ c}$标注，即预测 C 中的哪个候选项 c_i 最适合与查询实体 q 匹配。基于所使用的语言模型类型来进行标注，我们定义了以下策略：

- **大语言模型标注**：每个元组 $t_{q,\ c}$ 都会被序列化为一个 SELECT 提示，由大语言模型从候选实体中选出一个，作为与查询实体实际匹配的实体。为确保标注质量，DistillER 使用中型或大型大语言模型，即参数量至少为 8B 的模型。
- **小语言模型标注**：与大语言模型相比，小语言模型效率更高，但缺少交互式接口。它们无法处理基于提示的表达形式，只能处理固定的成对输入（e_a，e_b），这种输入近似于 MATCH 提示，但仍需进行特定任务训练（即微调）才能达到较高的有效性。为此，DistillER 使用预训练大语言模型按以下方式标注部分训练数据：首先，将训练集划分为两个互不相交的部分，分别占 X% 和 Y%（其中 $Y\% = 100\% - X\%$）。接着，大语言模型对前一部分进行标注，并使用生成的噪声标签微调选定的小语言模型。最后，微调后的小语言模型对训练集中剩余的 Y% 进行标注。请注意，这可能导致每个元组对应多个匹配项，但这符合此步骤关于训练数据含有噪声的假设。总的来说，小语言模型的标注时间可表示为：

$$\text{小语言模型标注时间} = \text{X\%} \cdot \text{大语言模型标注时间} + \text{小语言模型微调（X\%）} + \text{小语言模型测试（Y\%）}。$$

由于第一项远大于最后两项，因此我们可以放心地将上述公式简化如下：

$$\text{小语言模型标注时间} \approx \text{X\%} \cdot \text{大语言模型标注时间}$$

- **多教师标注**：该策略不依赖单个模型进行标注，而是汇总模型委员会生成的标签，并通过简单的投票机制为每个元组进行标注，返回出现频率最高的标签；若出现平票，则从得票最多的标签中随机选择。由于大语言模型与小语言模型标注之间缺乏兼容性，我们会组合使用同一模型系列中的模型，即所有模型均为大语言模型或小语言模型。

示例。 在图*1*的示例中，使用大语言模型标注器可能会选择候选项*2*，它也是一台产品编号相同的索尼唱盘机。使用小语言模型标注器，则可能会建议候选项*2*和*3*，它们都是索尼唱盘机。最后，假设我们使用由三个大语言模型组成的委员会，它们

分别选择候选项*2*、*2*和*3*为正确项。多教师模型方法会建议将候选项*2*标记为真实匹配项，因为它获得的票数最多。

4.2.2 指令。 DistillER 框架的此步骤仅在使用大语言模型作为教师模型时适用。其目标是确定 SELECT 提示中任务描述的形式（即图1中的粗体部分）。系统支持两种不同形式的指令，二者在 SELECT 提示输出的格式和范围上有所不同。DistillER 支持以下两类指令：

• **仅答案：** 系统要求大语言模型从元组 t_q中c识别最有可能匹配的候选项 c_i。这种简洁的格式无需额外推理即可实现直接监督。

• **答案 + 解释：** 系统要求大语言模型不仅返回匹配的候选项 c_i，还要说明其选择的理由。这种扩展格式使 DistillER 能够同时捕获决策及其背后的推理过程，从而在蒸馏过程中提供更丰富的知识。

4.3 蒸馏算法

该组件专注于将提取的知识K与原始元组集合 C一同传递给学生模型。它接收由数据选择生成的训练集以及知识 elicitation 步骤积累的知识作为输入，输出则为经过训练的学生模型。此过程涉及两个关键挑战：（i）选择合适的学生模型；（ii）定义有效的训练策略。对于前者，DistillER 支持使用大语言模型和小语言模型作为学生模型。对于后者，下面将详细介绍 DistillER 支持的策略。

4.3.1 监督微调。 此步骤使用从知识 K 中得到的标注数据对预训练模型进行微调，以提升其在特定任务上的表现。DistillER 支持的两类学生模型采用不同的微调流程：

• **大语言模型**：这种方法以对话形式，使用教师模型生成的标注元组对大语言模型进行微调。SELECT 提示确保只有一个正例答案，从而实现直接的端到端预测，无需进一步消歧。

• **小语言模型**：该方法被建模为二元分类任务，输入由固定的成对输入（e_a, e_b）组成，用以近似 MATCH 提示，预测结果为 0（否）或 1（是）。由于成对输入可能产生多个正类预测（即，元组$t_{q,\ c}$ 中的多个候选项都被判定为匹配项），该方法还会采用两种受支持的消歧策略之一：（i）仅根据已分类为匹配项的结果，在大语言模型上使用 SELECT 提示，以获得唯一的最终答案；（ii）应用利用输入数据源之间一对一约束的聚类算法。例如，唯一映射聚类（UMC） [10] 按相似度分数从高到低对冲突匹配项排序，并为每个查询实体保留权重最高的匹配项作为真实匹配。需要注意的是，SELECT 提示的结构与前文介绍的相同，但所涉及的候选项集合有所不同；该集合是阻塞阶段产生的一个较小子集。

示例。 图*1*示例中的元组可用于训练大语言模型学生模型，只要附上检测到的匹配项的 *ID*（最好是正确的，即 *[2]*）。在这种情况下，图*1*中的元组在小语言模型微调时会被转换为五个训练实例（即 (e_a, e_b) 对），每个候选实体对应一个。每个提示都配有一个是或否答案，具体取决于该候选项是否被标记为答案。如果该元组用于测试，我们假设候选项 *2* 和 *3* 都被标记为可能的答案。消歧步骤应检测到候选项 *3* 的产品编号不同，因此不是有效答案。

4.3.2 强化学习。 强化学习通过利用从提取的知识中得到的偏好信号，提供了一种进一步优化学生模型的机制 K。训练模型的目标是最大程度地与这些偏好对齐，使其能够动态适应，达到仅靠监督标签无法实现的效果。在本研究中，我们聚焦于仅适用于大语言模型的强化学习策略，因为所研究的方法依赖于小语言模型学生模型不具备的能力（例如指令遵循和丰富的偏好建模）。优化过程取决于奖励或偏好信号的构造方式（即强化学习策略），以及学生模型的初始化状态（即模型状态）。

策略： 我们考虑两种直接利用偏好信号的强化学习方法：

• **广义奖励策略优化（GRPO）** [22]：这是一种策略优化变体，利用基于分组的优势估计，根据每个提示对应的多个回答高效地更新学生模型。这种方法特别适用于大语言模型的复杂推理任务。我们使用以下奖励函数：

– **数字计数奖励：** 根据方括号内数字的数量为生成结果评分。形式化表示如下：

$$R_{\text{digit}}(c) = \begin{cases} 0 & \text{if } br(c) = 0, \\ \frac{1}{br(c)} & \text{if } br(c) > 1, \end{cases} \tag{1}$$

其中， $br(c)$表示方括号 [] 内的数字数量，即生成结 c果（即模型生成的输出）。

– **基于长度的奖励：** 去除空格和换行符后，根据生成结果的长度评分。形式化表示如下：

$$R_{\text{length}}(c) = \begin{cases} 0 & \text{if } |c| = 0, \\ 1 & \text{if } |c| \in \{1, 2\}, \\ \frac{3}{|c|} & \text{if } |c| > 2, \end{cases} \tag{2}$$

其中， c 表示去除首尾空格和换行符后的生成结果。

– **正确答案奖励：** 根据预测答案是否与真实答案匹配为生成结果评分。形式化表示如下：

$$R_{\text{answer}}(c) = \begin{cases} 0 & \text{if } pr(c) \neq t, \\ 1 & \text{if } pr(c) = t, \end{cases} \tag{3}$$

其中， $pr(c)$表示根据生成结果 c得出的预测答案， t表示正确答案。

最后，**总奖励**计算为三个单独奖励的加权总和：

$$R_{\text{total}}(c) = w_1 \cdot R_{\text{digit}}(c) + w_2 \cdot R_{\text{length}}(c) + w_3 \cdot R_{\text{answer}}(c) \tag{4}$$

示例。 以等权重扩展我们的示例：– “正确答案是 *[2]*”：此回答具有 $R_{total}(c) = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot \frac{3}{25} + \frac{1}{3} \cdot 1 = 0.706$

– “答案: [2] R_{total} c $= \frac{1}{3}$ $1 + \frac{1}{3}$ $\frac{3}{11} \pm \frac{1}{3}$ $1 =$ ”: 此回答具有() ·· 0.757 .

– “正确答案是 *[2]* , 因为 *[1]* 和 *[4]* 属于不同的品牌。”: 此回答 R_{total}（c） $= \frac{1}{3} \cdot \frac{1}{3} + \frac{1}{3}$- $\frac{3}{70} + \frac{1}{3}$-1 $= 0.458$这体现了对简洁、准确回答的偏好，避免不必要地提及其他选项。

• **直接偏好优化（DPO）** [19]: 这是一种基于偏好的方法，无需显式进行奖励建模，便可直接利用成对偏好数据来对齐学生模型。

示例。按照我们的示例，模型将通过对话进行训练，其中提示为 *SELECT* 提示，并同时提供两种偏好。“选中样本”是被接受的答案（*[2]* 在图*1*的示例中），而“拒绝样本”则可以是其他任何答案，例如 *[1]*。启用解释功能后，该方法会同时考虑答案及其理由。

模型状态：学生模型的初始化对强化学习性能起着关键作用：

• **预训练：** 模型基于通用预训练进行初始化，依靠涌现能力来适应偏好信号。

• **监督微调：** 模型已针对目标任务进行微调，因此能够更快速、更有效地纳入偏好。

与监督微调相比，强化学习能够实现持续适应，使学生模型根据从知识 K 中得出的不断变化的偏好信号，动态改进其预测。这种动态指导既可以来自带有噪声的人类标注，也可以来自模型生成偏好，帮助学生模型更好地与预期行为对齐。

5 评估

本节将全面评估 DistillER，考察其在流程多个阶段的表现。

5.1 实验设置

数据集。 我们在以往研究中广泛使用的八个真实世界实体解析数据集上评估了我们的方法 [15, 27]。每个实体都表示为一组 <属性、值> 对，并将其序列化为单条文本描述。这种表示方式灵活，既适用于表格数据，也适用于半结构化数据源。表1总结了数据集的统计信息。训练时，我们从每个数据集中抽取 10% 的实体，构建一个全局训练集。

模型。 我们使用了四种开源大语言模型：Llama-3.1:8b、Llama-3.1:70b、Qwen-2.5:14b 和 Qwen-2.5:32b[1]，以及两种小语言模型：S-MiniLM[2] 和 RoBERTa[3]。

评估指标。 我们报告所有实验中的 F 值（F1分数）。请注意，对于使用 SELECT 提示的模型，精确率=召回率；而对于 MATCH 提示，精确率 ≠召回率。

设置。 对于预训练大语言模型，我们使用 Ollama[4], 提供的模型，而微调则使用 Unsloth[5]进行。所有实验均在一台服务器上运行，该服务器搭载 Ubuntu 20.04、AMD Ryzen Threadripper 3960X 24 核处理器、256 GB RAM 和 RTX 4090 GPU。

阻塞。 对于阻塞，我们采用文献 [32, 34]中的标准方法：使用 S-GTR-T5 对实体进行序列化，候选对通过互惠剪枝进行精炼，并使用 FAISS 进行索引，为每个实体检索前N 个候选项。

[1]https://ollama.com/library/{X} llama3.1:8b、llama3.1:70b、qwen2.5:14b qwen2.5:32b [2]https://huggingface.co/sentence-transformers/all-MiniLM-L6 2 3 -v https://huggingface.co/FacebookAI/roberta-base [4]https://ollama.com/ [5] https://unsloth.ai/

5.2 数据选择

Q1：哪种数据选择策略最有效?

在本实验中，我们评估用于选择高质量训练元组的无监督策略，无需依赖标签。我们从每个数据集中采样 10% 的元组，得到包含 2,181 个元组的全局训练集。为近似 [27, 33], 中建议的 3:1 正负样本比例，我们将 $p = 7.5\%$ 和 $n = 2.5\%$ 分别设为基于排序的方法（最大值和前二）以及基于聚类的方法（K 均值和凝聚式聚类）的比例。作为纯无监督基线，我们纳入随机选择 10% 的元组。此外，利用可用标签，我们构建第二个基线（采样），明确选择 7.5% 的正元组和 2.5% 的负元组。为评估每种策略，我们采用两个大语言模型 Llama-3.1:8b 和 Qwen-2.5:14b，以其预训练形式在各选择方法构建的不同数据划分上生成预测。我们比较它们在这些数据划分上的表现，评估其对数据集偏斜的敏感性，以及构建均衡训练子集的重要性。

为此，我们将正例比例定义为“正例实例”/“总实例数”，其中大于 0.5 的值表示正例对多于负例对。表3报告了正例比例，凸显了生成的数据集中严重的类别不平衡。随机选择会过度偏向负类，因为查询元组中真正为正例的数量极少（另见表1）。在一些数据集（如 D5）中，排序在最大值和前二两种策略下都近似于 3:1 的比例；但在另一些数据集（如 D6 和 D8）中，则会降至各占 50%。尽管如此，与随机选择或聚类相比，排序通常能更充分地保留正例。相比之下，聚类往往偏向负类（如 D6、D8），从而降低下游性能。D5–D8 数据集凸显了这一现象：这些数据集以负例为主，且实体包含多个属性和词元，因此元组中的所有候选项更容易出现假阳性词元重叠和错误的相似度分数。因此，最终得到的相似度向量可能会被错误地分配到负类簇中，使大多数方法无法实现有效的类别分离。

表4中的结果证实了这些观察。对于 Llama-3.1:8b，高度不平衡的样本（随机、聚类）会导致较差的性能，而在最大值和前二策略下，排序的表现均与带标签的采样基线相当。然而，规模更大的 Qwen-2.5:14b 对选择策略的敏感度较低，在所有方法下都能稳定获得更高的分数。

总结：排序是最可靠的无监督选择方法，能够保持均衡的类别分布，并且在最大值和前二策略下，取得与带标签的采样基线相当的性能。

5.3 教师比较

Q2：哪些教师模型最有效、最高效?

在本实验中，我们比较不同教师模型在标注元组时生成单一答案的效果和效率。对于效果，我们评估训练数据上的F1分数；对于效率，我们测量所需时间。我们考虑两类教师模型：大语言模型和小语言模型，并分别进行研究，最后在末段中进行比较。

	产品						论文				电影					
	D2		D3		D8		D4		D9		D5		D6		D7	
Name	Buy	Abt	Amz	GPr	Wmt	Amz	ACM	DBLP	DBLP	学术搜索	IMDb	TMDb	IMDb	TVDb	TVDb	TMDb
|E|	1,064	1,076	1,354	2,935	2,554	22,068	2,272	2,576	2,476	61,350	5,117	6,042	5,117	5,178	5,178	6,042
|A|	3	3	4	4	6	6	4	4	4	4	13	30	13	9	9	30
|D|	1,064		1,102		853		2,214		2,304		1,965		966		1,035	

表1：数据集u 用于实验评估。 $|E|$ 表示实体数量， $|A|$ 表示属性数量以及 $|D|$ 表示数量的重复项。

		D2		D3		D8		D4		D9		D5		D6		D7	
		Buy	Abt	Amz	GPr	Wmt	Amz	ACM	DBLP	DBLP	学术搜索	IMDb	TMDb	IMDb	TVDb	TVDb	TMDb
$\|e_i\|$	50%	145	370	579	368	233	235	239	213	213	193	154	56	154	80	80	56
	95%	257	528	6,894	432	284	301	315	284	281	266	217	682	217	715	715	682
$\|C_i\|$	50%	6		7		10		4		7		3		2		5	
	95%	10		10		10		9		10		10		10		10	
$\|EP_i\|$	50%	2,362		3,155		2,583		1,091		1,564		322		314		360	
$\|OP_i\|$	50%	3,408		3,413		3,191		2,143		2,345		1,633		1,440		3,619	
	95%	5,012		5,563		3,853		3,186		3,123		2,501		2,154		12,413	
|P|		0.95		0.71		0.36		0.97		0.83		0.31		0.14		0.19	

表2：各数据集的提示相关统计数据。 $|e_i|$ 表示以字符数计的序列化实体长度， $|C_i|$ 表示阻塞候选列表大小， $|EP_i|$ 和 $|OP_i|$ 分别表示以字符数计的提示长度估计值和观测值， $|P|$ 表示正例率；展示了所有实体对中各项统计数据的50%（中位数）和90%百分位数。

		排序			聚类		
	随机	Max	前二		Agg	K均值采样	
D2	0.96	0.94	0.95		0.99	0.94	0.75
D3	0.75	0.58	0.61		0.72	0.72	0.75
D4	0.98	0.90	0.93		0.98	0.99	0.75
D5	0.42	0.74	0.67		0.44	0.44	0.75
D6	0.17	0.46	0.44		0.17	0.14	0.75
D7	0.20	0.43	0.42		0.23	0.24	0.75
D8	0.34	0.50	0.46		0.38	0.35	0.75
D9	0.94	0.82	0.83		0.91	0.88	0.75
Mean	0.59	0.67	0.67		0.60	0.59	0.75

表3：不同数据选择策略下的正例比例。

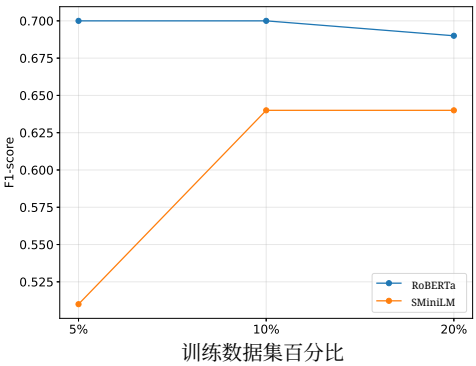


图3：使用 Qwen-2.5:32b 标签训练时，不同训练集大小下各小语言模型教师的训练数据平均 F1分数。

大语言模型教师：在本实验中，我们使用全部四个大语言模型以及多教师设置，并按效果对它们进行比较。结果如表5所示。

在 Llama-3.1 系列中，我们观察到使用较大模型时效果明显提升，F1分数从 0.72 提高到 0.82。尽管如此，两个模型在数据集 D3、D7 和 D8 上的表现似乎都不佳，这可能是因为这些数据集的实体档案过于庞大。查看表2中的统计数据以及实体以字符数计，我们确认 D3 的实体数量较多，平均字符数从 579 到 7K 不等。结合阻塞候选列表大小 $|C_i|$ ——大多数数据集中阻塞后每个实体有五个候选项，而 D8 中有十个——我们预计产品数据集中的提示会更长 $|EP_i|$。除产品数据集外，我们在实践中还发现，D7 的平均提示长度也较长（$|OP_i|$）。类似地，在 Qwen-2.5 系列中，较大的模型将性能从 0.81 略微提升至 0.83。总体而言，前二的表现均由较大的模型取得。为衡量模型间的互补性，我们比较了 Llama 和 Qwen 单个模型的平均召回率（0.82）与汇总所有模型预测后得到的召回率（0.88）。这表明这些模型在一定程度上具有互补性。

因此，采用简单的多数投票方案，如 [33], Multi-LLM，可实现 0.83 的平均 F1分数；但这需要并行运行四个教师模型。因此，这种方法无法证明其在效果与效率之间的权衡是合理的，可以适当地省略。

总结：使用大语言模型作为教师模型可以生成接近真实标注数据的高质量标签，而多教师模型设置并不能证明其在效果与效率之间的权衡方面具有优势。

小语言模型教师：首先，我们比较由最佳大语言模型教师生成的不同噪声标签训练出的两个小语言模型教师。我们使用 20% 的训练数据（占总数据的 8%）由大语言模型进行标注，然后用这些数据微调一个小语言模型，再由该模型标注剩余的 80%（占总数据的 8%）。结果如表6所示。

S-MiniLM的平均表现最差（0.65），而 RoBERTa 表现更好，使用来自 Qwen-2.5:32b 的噪声标签时，平均达到 0.69。这应归因于 RoBERTa 是一个规模更大、鲁棒性更强的模型（参数量为 1.25 亿，而 S-MiniLM 为 2200 万），而 S-MiniLM 更注重时间效率。尽管如此，

	随机		排序 Max		前二		聚类 Agg		K 均值		采样	
	L:8b	问题：14b	L:8b	问题：14b	L:8b	问题：14b	L:8b	问题：14b	L:8b	问题：14b	L:8b	问题：14b
D2	0.89	0.94	0.92	0.92	0.92	0.91	0.88	0.90	0.87	0.90	0.68	0.86
D3	0.58	0.69	0.57	0.64	0.54	0.61	0.46	0.59	0.51	0.58	0.61	0.69
D4	0.96	0.97	0.86	0.93	0.88	0.95	0.96	0.97	0.96	0.99	0.74	0.81
D5	0.61	0.78	0.87	0.98	0.81	0.94	0.69	0.86	0.67	0.83	0.85	0.94
D6	0.43	0.81	0.61	0.72	0.57	0.71	0.44	0.83	0.39	0.80	0.77	0.82
D7	0.34	0.66	0.50	0.63	0.48	0.63	0.36	0.70	0.40	0.70	0.74	0.88
D8	0.36	0.67	0.59	0.77	0.55	0.75	0.44	0.65	0.38	0.65	0.64	0.82
D9	0.90	0.96	0.83	0.91	0.82	0.88	0.85	0.93	0.83	0.92	0.73	0.87
Mean	0.63	0.81	0.72	0.81	0.70	0.80	0.63	0.80	0.63	0.80	0.72	0.84

表4：不同数据选择策略在训练数据上的F1分数。L:8b代表Llama-3.1:8b，Q:14b代表Qwen-2.5:14b。

	Llama-3.1		Qwen-2.5		Multi-LLM
	8b	70b	14b	32b	
D2	0.92	**0.97**	0.92	0.92	0.95
D3	0.57	0.66	0.64	0.65	**0.67**
D4	0.86	0.94	0.93	**0.96**	0.93
D5	0.87	0.98	0.98	**0.99**	0.98
D6	0.61	0.71	0.72	**0.73**	**0.73**
D7	0.50	**0.66**	0.63	0.63	0.65
D8	0.59	0.74	0.77	**0.84**	0.82
D9	0.83	0.89	0.91	**0.96**	0.93
Mean	0.72	0.82	0.81	**0.83**	**0.83**

表5：各LLM教师模型对应的标注训练数据F1分数。

	S-MiniLM			RoBERTa		
	GT	L：70B	Q：32B	GT	L：70B	Q：32B
D2	0.48	0.48	0.41	0.58	0.51	**0.61**
D3	0.38	0.37	0.43	0.48	**0.49**	0.46
D4	0.78	0.77	0.75	**0.79**	0.76	0.78
D5	0.84	0.85	0.84	**0.86**	0.85	0.85
D6	0.71	0.65	0.66	**0.73**	0.65	0.66
D7	0.68	0.66	0.62	**0.75**	0.66	0.64
D8	0.69	0.63	0.64	**0.72**	0.68	**0.72**
D9	0.76	0.76	0.76	**0.80**	**0.80**	0.79
Mean	0.67	0.65	0.64	**0.71**	0.68	0.69

表6：不同小语言模型教师在使用来自不同大语言模型教师的噪声数据和真实标签（20%）进行训练时，在标注训练数据上的F1分数。L代表Llama-3.1，Q代表Qwen-2.5

所有教师模型在产品数据集上的表现都更差，这表明产品表示中包含的共同词元可能会干扰教师模型识别真实匹配项。例如，一条以太网线的描述可能是“Cat6 以太网线 5m”，而另一条记录则是“Cat6 以太网线 15m”；共同词元会掩盖长度这一细微但至关重要的差异。此外，使用真实标注数据训练时，两种小语言模型教师

略好一些，但差异很小（不到 5%）。

接下来，我们评估训练数据规模变化时小语言模型的稳健性。我们将原来的 20% 缩减至 10%

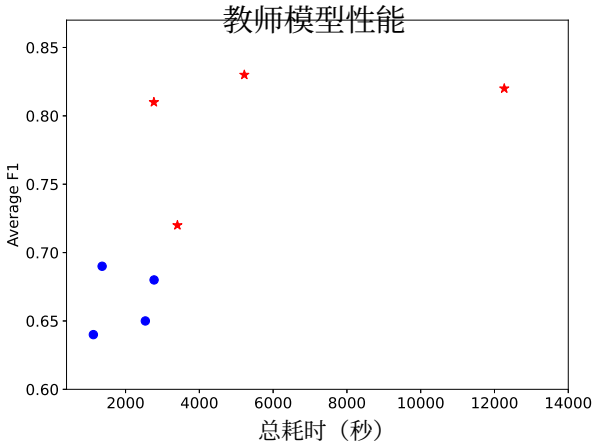


图4：不同教师模型在有效性与时间效率之间的权衡。

和 5%，仅使用来自 Qwen-2.5:32b 的噪声标签。结果如图3所示。

正如预期，S-MiniLM 受影响最大，得分从 0.64 降至 0.50，下降了 20%。RoBERTa 最初是在更多、更大的数据集上训练的，且旨在提升稳健性，因此在所有数据集占比下都表现得更加稳定。

总结：使用小语言模型教师可以生成高质量标签，尤其是使用 *RoBERTa* 时；即使仅使用很少的训练数据进行微调，它也相当有效且稳健。

比较：我们针对各类别中的所有教师模型，比较其有效性（即平均 F1分数）和时间效率（即总耗时）。结果见图4。请注意，LLM教师模型的总耗时包括标注所有训练数据所需的时间；而小语言模型教师的总耗时计算方式请参见第4.2节中的相应公式。

大语言模型教师在效果方面明显优于小语言模型，取得了平均高于 0.80 的 F1分数，而小语言模型教师的分数则未超过 0.70。在时间效率方面，小语言模型教师所需的推理时间仅为相应大语言模型教师的 ≈ 20%。这是因为大语言模型推理占据了小语言模型教师运行时间的主要部分，但仅应用于 20% 的训练数据。总体而言，更大的模型（Qwen:32b、Llama:70b）证实了这样的预期：更高的效果伴随着更长的推理时间；而较小的模型推理时间更短，但效果也有所降低。

	UMC	学生模型	教师模型	微调 真实标签	大语言模型	小语言模型
D2	0.79	0.88	**0.92**	0.75	0.88	0.48
D3	0.49	0.56	**0.65**	0.51	**0.65**	0.41
D4	0.98	0.96	**0.99**	**0.99**	**0.99**	0.85
D5	0.40	0.63	0.84	0.85	**0.91**	0.89
D6	0.17	0.49	0.89	0.89	**0.90**	**0.90**
D7	0.19	0.34	0.77	**0.81**	0.77	0.79
D8	0.20	0.33	**0.83**	**0.83**	0.82	0.79
D9	0.89	0.86	**0.92**	0.90	0.88	0.65
Mean	0.51	0.63	**0.85**	0.82	**0.85**	0.72

表 7：在测试数据上，使用不同教师模型微调的 Llama-3.1:8b 的 F1分数，与主要相关基线进行比较。大语言模型为 Qwen-2.5:32b，小语言模型是基于 Qwen-2.5:32b 标签训练的 RoBERTa。

	UMC	S-MiniLM 真实标签	大语言模型	小语言模型	RoBERTa 真实标签	大语言模型	小语言模型
D2	**0.79**	0.65	0.65	0.47	0.78	0.75	0.51
D3	**0.49**	0.31	0.31	0.15	0.40	0.41	0.15
D4	**0.98**	0.96	0.96	0.92	0.97	**0.98**	0.88
D5	0.40	0.73	0.71	0.73	0.79	**0.81**	0.73
D6	0.17	0.37	0.40	0.48	0.53	**0.55**	0.42
D7	0.19	0.60	0.57	**0.62**	0.49	0.61	0.56
D8	0.20	0.42	0.45	0.29	0.44	**0.57**	0.28
D9	**0.89**	0.86	0.86	0.76	**0.89**	0.86	0.72
Mean	0.51	0.61	0.61	0.55	0.66	**0.69**	0.53

表8：不同教师模型下，微调小语言模型在测试数据上的F1分数与基线的比较。大语言模型为Qwen-2.5:32b，小语言模型为基于Qwen-2.5:32b标签训练的RoBERTa。

这证明了我们的目标是将较小模型的效率与较大模型的效果结合起来。

以下实验仅考虑将 Qwen:32b 和基于 Qwen:32b 标签训练的 RoBERTa 分别用作大语言模型教师模型和小语言模型教师模型。

总结：在效果方面，大语言模型教师明显更优，但耗时更长；小语言模型教师则能很好地兼顾效果与效率，运行速度快得多。

5.4 监督微调

Q3：基于噪声标签微调的模型表现如何?

在本实验中，我们研究基于噪声标签微调的模型的性能。根据底层模型，我们将其分为两类：大语言模型和小语言模型。

对于大语言模型训练，我们使用秩为 16 的 LoRA $\alpha = 16$，且不使用丢弃，以实现高效适配；与完整微调相比，这种方法能减少内存占用。训练进行三个 epoch，学习率为 $2 \cdot 10^{-4}$，梯度累积步数为 4，并使用 AdamW 8 位优化器。我们还采用 4 位量化来减少内存占用。对于小语言模型训练，我们遵循 EMTransformer 方法 [2]。

微调后的大语言模型：我们采用上一节中各类别表现最佳的教师模型，并使用这些噪声标签对 Llama-3.1:8b 进行微调，以便将其与以下模型进行比较：(i) Unique Mapping Clustering（UMC），一种用于无监督实体匹配的基线算法；(ii) 其预训练版本（学生模型）；(iii) 教师模型（Qwen:32b）的预训练版本；(iv) 其基于真实标注数据微调后的版本（真实标签）。结果见表 7。

由LLM教师模型训练的学生模型始终优于由相应的SLM训练的学生模型，尤其是在D2和D3中；如表6所示，这两个产品数据集的标签存在问题。两种学生模型都超过了UMC基线和预训练模型，这证实了即使标签带有噪声，微调也能提升性能。有趣的是，由LLM教师模型训练的学生模型甚至超过了使用真实标签训练的模型，平均F1分数达到0.85。这可以解释为，涉及更难或处于边界情况的标注提供了更有力的训练机会。在微调过程中接触这些情况，有助于模型更好地区分模式，从而提升整体性能。这一点也适用于教师模型，因为在某些情况下，尤其是在D5中，由LLM教师模型训练的学生模型超过了教师模型。总体而言，由LLM教师模型训练的学生模型

	学生	教师	微调大语言模型 LLM	SLM	微调小语言模型 RoBERTa LLM	RoBERTa SLM	SMiniLM LLM	SMiniLM SLM
Ann.	0	0	5,219	1,364	5,219	1,364	5,219	1,364
训练	0	0	4,827	4,793	1,465	1,514	378	393
D2	1,495	2,801	2,145	2,057	12	12	3	3
D3	2,151	3,329	2,771	2,700	25	25	7	6
D4	2,118	4,663	4,560	4,414	19	19	5	5
D5	7,099	7,135	7,685	7,438	42	42	11	11
D6	8,446	6,628	7,123	6,869	31	31	8	8
D7	7,562	8,264	9,873	9,582	70	70	18	18
D8	10,306	5,898	5,148	4,911	55	55	15	15
D9	3,111	5,473	5,288	4,795	26	26	7	7
Sum	42,291	44,195	54,644	48,926	6,969	3,163	5,675	1,835

表9：不同教师模型在测试数据上使用 Llama-3.1:8b 进行监督微调所需的时间（秒），并与基线进行比较。大语言模型为 Qwen-2.5:32b，小语言模型则使用基于 Qwen-2.5:32b 标签训练的 RoBERTa。

保持了与教师模型同样高的有效性，这正是知识蒸馏的目标之一。

总结：使用大语言模型生成的噪声标签对大语言模型进行微调，效果优于基线，也优于使用小型语言模型生成的标签进行微调。

微调小语言模型：我们现在使用 S-MiniLM 和 RoBERTa 评估微调小语言模型，这些模型使用来自 Qwen-2.5:32b 和 RoBERTa 的噪声标签进行训练。基线包括使用真实标签微调的小语言模型以及 UMC算法。结果见表8。

对于每种小语言模型，基于小语言模型的标注表现都逊于基于大语言模型的标注；对于 S-MiniLM 和 RoBERTa，平均分别低 10% 和 23%。两者之中，RoBERTa 效果更好，使用基于大语言模型的标注时，平均 F1分数达到 0.69，而 S-MiniLM 则降至 0.61。这可能是因为 RoBERTa 是一个规模更大、鲁棒性更强的模型。两种模型的表现都与使用真实标注数据训练的版本相当，其中 RoBERTa 高出 3%。所有模型的表现均优于 UMC 基线，后者表现不佳。

总结：在大语言模型生成的噪声标签上微调小语言模型，其表现优于基线，也优于在小型语言模型生成的标签上进行微调。

比较：现在，我们从效果（平均 F1分数）和效率（推理时间）两方面比较各类别中的所有微调模型。效果结果见表7和表8，效率结果见表9。

微调后的大语言模型和小语言模型在推理时间上有明显差异，后者快了两个数量级

	UMC 预训练		GRPO			DPO		
			真实标签	大语言模型	小语言模型	真实标签	大语言模型	小语言模型
D2	0.79	**0.88**	0.75	0.81	0.59	0.56	0.72	0.50
D3	0.49	0.56	0.52	0.55	0.48	0.41	**0.58**	0.41
D4	**0.98**	0.96	0.96	0.97	0.93	0.86	0.94	0.63
D5	0.40	0.63	0.77	0.76	0.76	**0.86**	0.85	0.69
D6	0.17	0.49	0.78	0.75	0.80	0.84	**0.85**	0.72
D7	0.19	0.34	0.66	0.61	0.73	**0.76**	0.72	0.63
D8	0.20	0.33	0.48	0.36	0.60	0.61	**0.73**	0.57
D9	**0.89**	0.86	0.88	**0.89**	0.84	0.76	0.83	0.62
Mean	0.51	0.63	0.73	0.71	0.71	0.71	**0.78**	0.60

表10：在测试数据上，使用不同教师模型相较于基线模型，对 Llama-3.1:8b 进行强化学习的 F1分数。大语言模型为 Qwen-2.5:32b，小语言模型是在 Qwen-2.5:32b 标签上训练的 RoBERTa。PT 代表预训练。

			强化学习大语言模型			
			GRPO		DPO	
	学生	教师	LLM	SLM	LLM	SLM
Ann.	0	0	5,219	1,364	5,219	1,364
训练	0	0	18,071	17,562	5,453	5,644
D2	1,495	2,801	375	347	1,209	3,129
D3	2,151	3,329	504	470	2,450	4,392
D4	2,118	4,663	724	669	1,124	3,344
D5	7,099	7,135	1,168	1,074	3,571	8,443
D6	8,446	6,628	1,069	993	4,396	9,746
D7	7,562	8,264	1,571	1,450	7,729	10,928
D8	10,306	5,898	925	856	4,145	7,925
D9	3,111	5,473	792	726	2,127	5,044
Sum	42,291	44,195	30,423	25,515	37,427	59,962

表11：在测试数据上，不同教师模型在 Llama-3.1:8b 上进行强化学习所需的时间（秒），并与基线进行比较。LLM 为 Qwen-2.5:32b，SLM 为基于 Qwen-2.5:32b 标签训练的 RoBERTa。

。这应归因于小语言模型的规模远小于大语言模型。出于同样的原因，小语言模型的训练时间比大语言模型最多快一个数量级。在小语言模型类别中，SMiniLM 的规模小于 RoBERTa，因此训练时间缩短了 3/4。

与教师模型相比，从整体表现来看，小语言模型的效果较差，这意味着即使训练速度很快，它们也无法适应并达到教师模型的峰值表现。另一方面，大语言模型能够有所提升：我们清楚地看到，均值从 0.63 提升至使用小语言模型教师模型时的 0.72，以及使用大语言模型教师模型时的 0.85。值得注意的是，最后一个结果与教师模型的平均表现相同。比较推理时间后，我们发现了类似的情况，这可以归因于这两个模型的规模相差不大（学生模型 $\approx$ 8b，教师模型 $\approx$ 32b）；尽管如此，它们对 GPU 资源的需求却不同（学生模型 $\approx$ 5GB，教师模型 $\approx$ 20 GB）。

总结：微调后的大语言模型比对应的小语言模型需要更多时间，但在有效性方面与教师模型表现相近，同时降低了内存需求。

		Llama		GRPO		DPO	
	UMC	PT	SFT	PT	SFT	PT	SFT
D2	0.79	0.88	0.88	0.81	0.90	0.72	0.89
D3	0.49	0.56	0.65	0.55	0.65	0.58	0.58
D4	0.98	0.96	0.99	0.97	0.98	0.94	0.93
D5	0.40	0.63	0.91	0.76	0.90	0.85	0.86
D6	0.17	0.49	0.90	0.75	0.87	0.85	0.86
D7	0.19	0.34	0.77	0.61	0.76	0.72	0.72
D8	0.20	0.33	0.82	0.36	0.75	0.73	0.71
D9	0.89	0.86	0.88	0.89	0.92	0.83	0.87
Mean	0.51	0.63	0.85	0.71	0.84	0.78	0.80

表12：在测试数据上，不同模型版本在 Llama-3.1:8b 上进行强化学习的 F1分数，并与基线进行比较。PT 指预训练，SFT 指监督微调。

5.5 强化学习

Q4：基于噪声标签进行强化学习会如何影响模型性能?

在本实验中，我们从两个维度研究强化学习对噪声标签的影响：教师模型和标签的选择，以及底层模型的版本（预训练或微调）。

训练时，我们使用秩为 8 的 LoRA，$\alpha = 16$，且不使用丢弃，以实现高效适配，并相比全量微调减少内存占用。训练默认进行一个轮次，学习率为 $2 \cdot 10^{-4}$，梯度累积步数为 4，并使用 AdamW 8 位优化器。我们还采用 4 位量化来减少内存占用。

教师：我们评估不同教师模型对有效性和效率的影响。我们从前文各类别中选出表现最佳的教师模型，以单轮训练 Llama-3.1:8b。真实标签作为参照（基于人类反馈的强化学习）。我们采用 GRPO 和直接偏好优化两种优化算法。结果见表 10 和表 11。

就效果而言，使用噪声标签的 GRPO 的 F1分数（0.71）略低于使用真实标签的情况（0.73），这符合噪声带来的预期影响。令人意外的是，使用 DPO 和大语言模型生成的标签进行训练，F1分数达到 0.78，高于使用人工真实标签的情况（0.71），尽管前者的噪声更大。这一出乎意料的结果可能源于 LLM式偏好与 DPO目标的对齐程度更高，但仍需进一步研究。DPO 与 SLM标注器结合时，F1分数仅为 0.60，证实了其能力有限。

关于表11中的时间效率，GRPO 需要更长的训练时间，因为它会为每个样例生成多个候选项。DPO 训练速度更快，但推理成本更高：由 LLM教师模型训练时的推理速度约为由 SLM教师模型训练时的两倍，因为 LLM 能提供质量更高的偏好信号，从而使决策边界更加贴合，并减少推理过程中需要处理的不确定案例数量。

总结：采用大语言模型标注器的直接偏好优化是最有效的策略，且比 *GRPO* 训练更快，但由于基于参考模型进行评分，推理成本更高。

模型版本：我们评估底层模型版本是否会影响强化学习性能。我们使用预训练模型和微调模型（来自第5.4节，采用 Qwen-2.5:32b 噪声标签），训练一个周期并使用大语言模型标签。结果见表12。

GRPO 和直接偏好优化（DPO）均优于预训练模型（F1分数分别提升了 18% 和 2%）。GRPO 的提升幅度更大，可能是因为奖励函数鼓励生成简洁、准确的答案；而 DPO 则依赖信息量更丰富的

			SFT		DPO		
	UMC	PT	An.	Ex.	An.	示例预训练	示例监督微调
D2	0.79	0.88	0.88	0.83	0.72	0.11	0.08
D3	0.49	0.56	0.65	0.63	0.58	0.27	0.28
D4	0.98	0.96	0.99	0.98	0.94	0.14	0.04
D5	0.40	0.63	0.91	0.87	0.85	0.52	0.64
D6	0.17	0.49	0.90	0.89	0.85	0.61	0.82
D7	0.19	0.34	0.77	0.77	0.72	0.64	0.82
D8	0.20	0.33	0.82	0.83	0.73	0.64	0.71
D9	0.89	0.86	0.88	0.88	0.83	0.13	0.07
Mean	0.51	0.63	0.85	0.84	0.78	0.38	0.43

表13：将标签的解释（示例）整合到监督微调和使用直接偏好优化的强化学习中时的F1分数，与仅答案版本（分析）以及两个主要基线：UMC和预训练（PT）进行比较

	DistillER		SotA					
	基于大语言模型的监督微调		大语言模型相关	小语言模型无监督		小语言模型监督		
	LLM	SLM	CEM	ZER	CBM	SWO	HGT	UCN
D2	**0.88**	0.48	0.47	0.45	0.00	0.72	0.59	0.52
D3	**0.65**	0.41	0.41	0.26	0.33	0.50	0.29	0.50
D4	**0.99**	0.85	0.83	0.94	0.86	0.28	0.98	0.60
D5	**0.91**	0.89	0.50	0.22	0.43	0.36	0.58	0.68
D6	**0.90**	**0.90**	0.19	0.81	0.48	0.08	0.41	0.77
D7	0.77	0.79	0.25	0.55	0.40	0.20	0.52	**0.96**
D8	0.82	0.79	0.08	0.57	0.09	0.39	0.53	**0.86**
D9	0.88	0.65	0.68	0.86	0.68	0.24	0.84	**0.92**
Mean	**0.85**	0.72	0.43	0.58	0.41	0.35	0.59	0.73

表14：DistillER与最先进算法的F1分数。大语言模型为Qwen-2.5:32b，小语言模型为基于Qwen-2.5:32b标签训练的RoBERTa。

示例，所需投入也更少。尽管如此，所有基于强化学习（RL）的方法表现仍不及未经强化学习微调的原始模型（Llama-SFT），后者的平均表现依然更优。

讨论：尽管本实验分析中考察的所有方法都会受到教师模型生成的噪声标签影响，GRPO 还取决于奖励函数的质量。我们可以确信，所设计的奖励函数仍有改进空间，可供未来工作进一步探索。关于直接偏好优化，我们注意到，在我们的场景中使用的反例全都是阻塞选项。另一种做法是选择差异更大的反例，例如从集合中随机选取一个实体，或选择排序远低于原始 k 候选项的阻塞选项。由于我们的场景采用端到端流程，我们希望模型仅依据阻塞选项学会区分这两个类别；然而，从基于强化学习的模型的表现来看，这似乎尤其具有挑战性，特别是考虑到标注答案本身可能一开始就不正确。

总结：强化学习可以提升预训练模型的性能，但对于已经微调的模型，带来的收益有限。

5.6 解释

Q5：如何通过微调和强化学习生成解释，而不是单个答案?

在本实验中，我们探究使用教师模型生成的解释，而不是仅依赖标注过程中生成的单个答案。我们使用表现最佳的教师模型 Qwen-2.5:32b，为每个查询实体的所有候选选项生成解释，其中也包括无选项，因为正确答案可能不在候选项之中。随后，我们将这些解释用于监督微调和结合直接偏好优化的强化学习。在这项评估中，我们衡量模型使用解释时能否正确预测，并据此评估推理模式是否能提升性能。不过，对解释内容进行评估仍是一项尚待开展但前景广阔的任务。通过对生成的解释进行初步检查，我们发现，当教师模型为错误选项（选项X）提供理由，而正确选项是选项Y时，它往往也会同时解释选项Y为何可能正确。这表明模型能够抵御干扰。

SFT 和 DPO 都是依赖带标签样例进行训练的策略：SFT 提供正确回答，而 DPO 同时提供偏好回答和可拒绝的备选项。然而，GRPO 不适用于此处，因为高效评估自由文本解释颇具挑战。研究将这些模型与基于单个答案训练的对应模型，以及 UMC 和预训练模型的基线进行比较。结果见表 13。

我们观察到，无论使用解释还是单个答案进行训练，监督微调的性能都保持相近，生成的回复依然连贯且有意义。相比之下，直接偏好优化的性能明显下降：预训练模型的效果损失约50%，而微调几乎没有带来提升。这可能是因为直接偏好优化过拟合了解释的写作风格或长度，而没有关注其正确性；尤其是当解释包含对错误答案的推理时。电影数据集（D6–D8）则是例外：较长且描述更详尽的解释似乎更符合DPO目标，因此其结果与使用单个答案训练时相近。

总结：在监督微调中，利用生成的解释引导模型得出正确答案是一种有效的替代方案，而 *DPO* 在这种设置下不够稳健。

5.7 基线

Q6：在监督微调中，DistillER 与现有的无监督和有监督方法相比表现如何?

在本实验中，我们将 DistillER 与不同类别中最先进的实体解析方法进行比较。对于基于大语言模型的方法，我们考察 ComEM **(CEM)** [27],，这是一种无监督方法，它先通过 MATCH 提示筛选候选项，再对预训练的 Llama-3.1:8b 应用 SELECT 提示。第二类基线方法涉及小语言模型。对于无监督方法，我们考察：ZeroER **(ZER)** [28], , 它在高斯混合模型下使用基于相似度的特征向量对候选对进行建模；以及 CollaborEM **(CBM)** [9], , 它通过自动生成标签和协同训练实现多特征协作。对于有监督方法，我们将 Qwen-2.5:32b 生成的噪声标签应用于以下方法：Hier-GAT **(HGT)** [31], , 它利用分层图注意力变换器，捕捉实体解析决策与属性关系之间的相互依赖；SudoWoodo **(SWO)** [26],，它采用对比表征学习来生成具有相似性感知能力的数据表征；以及 Unicorn **(UCN)** [24] 的有监督变体，该变体在特定训练数据集上进行微调。按照文献中的常见做法，我们依据有效性（F1分数）比较所有方法。结果见表14。

从大语言模型基线开始，CEM 的表现很差，平均 F1分数仅为 0.43。

原因在于其激进的候选过滤，导致部分查询没有任何候选项，因此被排除在结果之外。 .

在大多数情况下，基于小型语言模型的方法也表现不佳；其核心局限在于，阻塞对往往高度相似，难以提供足够的区分度来实现可靠的分离。对于无监督方法，尽管 CBM 在大多数数据集上都取得了较高的召回率，但其精确率较低，导致 F1 分数不佳。这种失衡可能是由训练数据分布偏斜造成的：数据偏向负类，使模型倾向于预测负例。在 D2 中，这一影响尤为明显，模型没有做出任何正类预测。大多数有监督模型的表现同样不佳。例如，采用对比学习的 SWO 无法区分这两个类别。只有 UCN 能够适应这些条件，这得益于多任务训练和丰富的训练数据；尤其是在 D7-D9 中，其表现优于 DistillER。

DistillER 优于基于小型语言模型的方法并不令人意外，因为后者依赖较小的模型，而前者利用了表示能力更强的大语言模型。使用大语言模型生成的标签进行微调时，DistillER 的表现最佳；但即使使用小型语言模型生成的标签进行训练，其表现仍可与其他最先进的方法相媲美。

总结：使用大语言模型生成的标签进行微调时，*DistillER* 的性能达到最先进水平；即使使用小型语言模型生成的标签，其竞争力依然很强，而基于较小型小语言模型的方法则难免受到局限。

6 结论

在本研究中，我们探讨了使用大语言模型进行实体解析的知识蒸馏，并提出了一个涵盖三个维度的框架：数据选择、知识 elicitation 和蒸馏算法。在数据选择方面，我们提出并评估了多种方法，最终发现，按阻塞相似度最大值对元组进行排序，能在效果和效率之间取得最佳平衡。在知识 elicitation 方面，我们比较了大语言模型、小语言模型和多大语言模型教师，结果表明，即使是中型模型（例如 32B），大语言模型教师也能生成高质量标签，而小语言模型教师则是快速且高效的替代方案。在蒸馏方面，我们评估了两种策略：监督微调和强化学习。对于监督微调，使用大语言模型教师生成的噪声标签对大语言模型进行微调可取得最佳结果，而小语言模型的竞争力较弱。对于强化学习，我们采用了成熟的优化算法，发现对微调后的模型进行直接偏好优化效果最佳，不过仍不及使用噪声标签直接进行监督微调的效果。我们还探索了生成解释而非单标签答案，结果表明，使用大语言模型生成的解释进行监督微调是最有效的策略。最后，与最先进的实体解析方法相比，采用监督微调的 DistillER 表现突出，超越了基于大语言模型和小语言模型的方法，无论这些方法是否需要标注数据。

致谢

本研究部分由欧盟项目 STELAR (101070122)、WiseFood (101181895) 和 AI-DAPT (101135826) 资助。

研究成果

我们的所有代码、数据集和微调模型均可在 https://github.com/alexZeakis/DistillER 上公开获取。

参考文献

[1] Ursin Brunner 和 Kurt Stockinger。2020。使用 Transformer架构进行实体匹配——数据集成迈出的重要一步。载于 EDBT。463–473。[2] Ursin Brunner 和 Kurt Stockinger。2020。使用 Transformer架构进行实体匹配——数据集成迈出的重要一步。载于第23届数据库技术拓展国际会议，哥本哈根，2020年3月30日至4月2日。OpenProceedings，463–473。[3] Cristian Buciluˇa、Rich Caruana 和 Alexandru Niculescu-Mizil。2006。模型压缩。载于第12届 ACM SIGKDD 知识发现与数据挖掘国际会议论文集。535–541。[4] Shengyuan Chen、Qinggang Zhang、Junnan Dong、Wen Hua、Qing Li 和 Xiao Huang。2024。利用大语言模型的噪声标注进行实体对齐。arXiv 预印本 arXiv:2405.16806（2024）。[5] Vassilis Christophides、Vasilis Efthymiou、Themis Palpanas、George Papadakis 和 Kostas Stefanidis。2021。大数据端到端实体解析概述。ACM CSUR 53，6（2021），127:1–127:42。[6] Vassilis Christophides、Vasilis Efthymiou 和 Kostas Stefanidis。2015。数据网络中的实体解析。Morgan & Claypool。[7] DeepSeek-AI、Daya Guo、Dejian Yang、Haowei Zhang 等。2025。DeepSeek-R1：通过强化学习激发大语言模型的推理能力。CoRR abs/2501.12948（2025）。[8] Xin Luna Dong 和 Divesh Srivastava。2013 6，11 2013 1188 1189 。大数据集成。PVLDB（），–。[9] Congcong Ge、Pengfei Wang、Lu Chen、Xiaoze Liu、Baihua Zheng 和 Yunjun Gao。2021。CollaborEM：一种利用多特征协作的自监督实体匹配框架。TKDE 35，12（2021），12139–12152。[10] Simon Lacoste-Julien、Konstantina Palla、Alex Davies、Gjergji Kasneci、Thore Graepel 和 Zoubin Ghahramani。2013。SIGMa：用于对齐大型知识库的简单贪心匹配。载于 KDD。572–580。[11] Yuliang Li、Jinfeng Li、Yoshihiko Suhara、AnHai Doan 和 Wang-Chiew Tan。2020。使用预训练语言模型进行深度实体匹配。Proc. VLDB Endow. 14，1（2020），50–60。[12] Yi Liu、Yuan Tian、Jianxun Lian、Xinlong Wang、Yanan Cao、Fang Fang、Wen Zhang、Haizhen Huang、Denvy Deng 和 Qi Zhang。2023。通过多视图增强蒸馏提升实体链接。arXiv 预印本 arXiv:2305.17371（2023）。[13] Venkata Vamsikrishna Meduri、Lucian Popa、Prithviraj Sen 和 Mohamed Sarwat。2020。用于实体匹配主动学习方法的综合基准测试框架。载于2020年 ACM SIGMOD 数据管理国际会议论文集。1133–1147。[14] Sidharth Mudgal、Han Li、Theodoros Rekatsinas、AnHai Doan、Youngchoon Park、Ganesh Krishnan、Rohit Deep、Esteban Arcaute 和 Vijay Raghavendra。2018。用于实体匹配的深度学习：设计空间探索。载于 SIGMOD。19–34。[15] Sidharth Mudgal、Han Li、Theodoros Rekatsinas、AnHai Doan、Youngchoon Park、Ganesh Krishnan、Rohit Deep、Esteban Arcaute 和 Vijay Raghavendra。2018。用于实体匹配的深度学习：设计空间探索。载于 SIGMOD Conference。ACM，19–34。[16] Derek Paulsen、Yash Govind 和 AnHai Doan。2023。Sparkly：一种简单却出人意料地强大的实体匹配 TF/IDF阻断器。Proc. VLDB Endow. 16，6（2023），1507–1519。[17] Ralph Peeters 和 Christian Bizer。2023。使用大语言模型进行实体匹配。CoRR abs/2310.11244（2023）。[18] Ralph Peeters 和 Christian Bizer。2023。使用 ChatGPT 进行实体匹配。载于欧洲数据库与信息系统进展会议。221–230。[19] Rafael Rafailov、Archit Sharma、Eric Mitchell、Christopher D Manning、Stefano Ermon 和 Chelsea Finn。2023。直接偏好优化：你的语言模型实际上是一个奖励模型。神经信息处理系统进展 36（2023），53728–53741。[20] Victor Sanh、Lysandre Debut、Julien Chaumond 和 Thomas Wolf。2019。DistilBERT，一种蒸馏版 BERT：更小、更快、更经济、更轻量。arXiv 预印本 arXiv:1910.01108（2019）。[21] John Schulman、Filip Wolski、Prafulla Dhariwal、Alec Radford 和 Oleg Klimov。2017。近端策略优化算法。arXiv 预印本 arXiv:1707.06347（2017）。[22] Zhihong Shao、Peiyi Wang、Qihao Zhu、Runxin Xu、Junxiao Song、Xiao Bi、Haowei Zhang、Mingchuan Zhang、YK Li、Yang Wu 等。2024。Deepseekmath：突破开放语言模型数学推理能力的极限。arXiv 预印本 arXiv:2402.03300（2024）。[23] Aaron Steiner、Ralph Peeters 和 Christian Bizer。2024。面向实体匹配的大语言模型微调。CoRR abs/2409.08185（2024）。[24] Jianhong Tu、Ju Fan、Nan Tang、Peng Wang、Guoliang Li、Xiaoyong Du、Xi-aofeng Jia 和 Song Gao。2023。Unicorn：一种支持数据集成中匹配任务的统一多任务模型。Proc. ACM Manag. Data 1，1（2023），84:1–84:26。[25] Somin Wadhwa、Adit Krishnan、Runhui Wang、Byron C. Wallace 和 Luyang Kong。2024。从自然语言解释中学习可泛化实体匹配。载于 EMNLP。计算语言学协会，6114–6129。[26] Runhui Wang、Yuliang Li 和 Jin Wang。2023。Sudowoodo：用于多用途数据集成与准备的对比式自监督学习。载于 ICDE。1502–1515。

[27] Tianshu Wang、Hongyu Lin、Xiaoyang Chen、Xianpei Han、Hao Wang、Zhenyu Zeng 和 Le Sun。2024。匹配、比较还是选择？对大语言模型用于实体匹配的研究。*CoRR* abs/2405.16884（2024）。[28] Renzhi Wu、Sanya Chaba、Saurabh Sawlani、Xu Chu 和 Saravanan Thirumu-ruganathan。2020。Zeroer：利用零个标注样例进行实体解析。载于*SIGMOD*。1149–1164。[29] Yikuan Xia、Jiazun Chen、Xinchi Li 和 Jun Gao。2024。APrompt4EM：用于通用实体匹配的增强提示调优。*CoRR* abs/2405.04820（2024）。[30] Xiaohan Xu、Ming Li、Chongyang Tao、Tao Shen、Reynold Cheng、Jinyang Li、Can Xu、Dacheng Tao 和 Tianyi Zhou。2024。大语言模型知识蒸馏综述。*arXiv* 预印本 *arXiv:2402.13116*（2024）。[31] Dezhong Yao、Yuhong Gu、Gao Cong、Hai Jin 和 Xinqiao Lv。2022。利用分层图注意力网络进行实体解析。载于*SIGMOD*。429–442。[32] Alexandros Zeakis、George Papadakis、Dimitrios Skoutas 和 Manolis Koubarakis。2023。用于实体解析的预训练嵌入：实验分析。*Proc. VLDB Endow.* 16，9（2023），2225–2238。[33] Alexandros Zeakis、George Papadakis、Dimitrios Skoutas 和 Manolis Koubarakis。2025。AvengER：针对实体解析中的 SELECT 提示集成并微调大语言模型。载于欧洲语义网会议。Springer，301–320。[34] Alexandros Zeakis、George Papadakis、Dimitrios Skoutas 和 Manolis Koubarakis。2025。预训练嵌入用于实体解析的深入分析。《*VLDB*期刊》34，1（2025），1–27。[35] Zeyu Zhang、Paul Groth、Iacer Calixto 和 Sebastian Schelter。2024。AnyMatch——利用小型语言模型实现高效零样本实体匹配。*CoRR* abs/2409.04073（2024）。