IRT2：不同规模知识图谱中的归纳式链接与排序

Felix Hamann[1], Adrian Ulges[1], Maurice Falk[1]

arXiv:2301.00716v1 [cs.LG] 2 Jan 2023

摘要：我们致力于解决工业应用场景中构建领域专用知识模型的难题，在这些场景中，带标签数据和分类信息起初十分稀缺。我们重点关注归纳式链接预测模型，并以此为基础开发实用工具，帮助知识工程师探索文本集合，并发现新的（所谓开放世界）实体，将其链接到知识图谱。我们认为，尽管近年来神经网络文本挖掘方法取得了令人瞩目的成果，但当前的基准测试未能充分反映工业实际场景中常见的挑战。因此，我们的首项贡献是推出一项开放基准测试 IRT2（基于文本的归纳推理），该基准测试（1）涵盖不同规模的知识图谱（包括非常小型的图谱），（2）包含偶然出现且质量较低的文本提及，（3）不仅包括三元组补全，还包括排序；后者对于帮助专家完成发现任务至关重要。

我们研究了两种用于归纳式链接预测的神经模型：一种基于端到端学习，另一种则分步骤从知识图谱和文本数据中学习。这些模型与强大的词袋基线展开竞争。结果表明，当可用于链接的图数据减少时，神经网络方法的性能会显著提升。对于排序，结果也很有希望，神经网络方法远胜于稀疏检索器。

关键词： 文本挖掘；知识图谱；自然语言处理。

1 引言

众所周知，知识图谱（KG）是网络搜索或个人助理等各种应用中的强大工具。它们是满足复杂信息需求的关键，这类需求需要推理，无法通过简单的文本匹配来解决。通常，人们认为知识图谱的规模很大 [Ho21],例如关联开放数据云[2] 或生物医学领域的知识图谱 [HSG15]。然而，在尚无大规模知识的其他行业中，知识图谱也可能发挥重要作用，例如在技术服务领域，知识图谱可以帮助推断机器故障的根本原因；在保险领域，可以对合同细节进行建模；在金融评级领域，则可以捕捉企业的供应链。这些领域中的显式知识十分匮乏。不过，通常能够大量获取的是文本数据，例如服务工单、保险索赔或商业报告。

[1]莱茵美因应用科学大学，德国威斯巴登
{felix.hamann,adrian.ulges,maurice.falk}@hs-rm.de[2]
https://lod-cloud.net



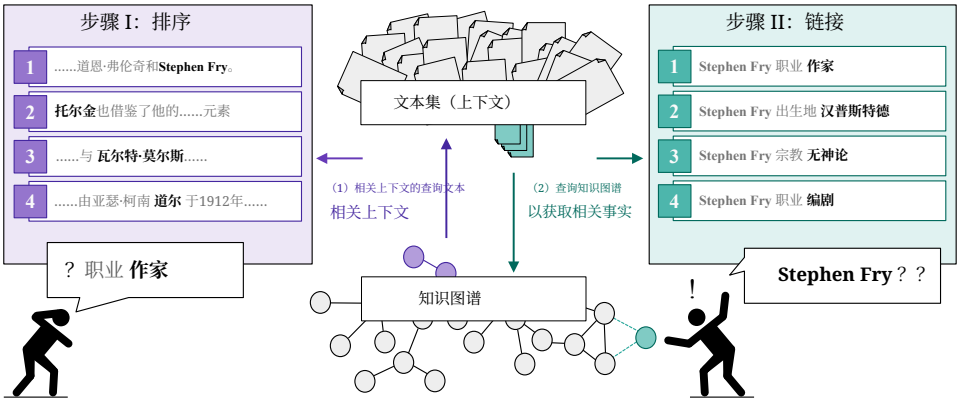


图 1：我们实现交互式知识获取的两步方法：首先，用户发现包含相关实体的句子（左侧的发现/排序）。其次，对于感兴趣的提及，系统会像处理实体一样，为该提及推荐三元组（右侧的链接）。

本文旨在支持专家通过检查文本来构建知识图谱。为此，我们假设专家以迭代方式向图中添加三元组，在此过程中，他们会在文本集中定位并识别新的实体，再将其链接到图中。我们着重探讨这一过程中的两个关键步骤，如图 1 所示：

1. **发现/排序**：专家根据知识图谱中的部分事实（即一个关系-顶点元组）浏览文本集。他/她会指定信息需求（例如“帮我找演员”，或更具体地说，“找出可能包含与已知实体具有职业关系的实体提及的句子；该已知实体是演员”）。系统会返回一个可能有用的句子排序列表，从而形成一个排序任务。

2. **链接**：专家研究这些句子。一旦发现一个有趣的实体 x（例如Frances McDormand），我们希望将 x 链接到图中，并通过添加三元组来补充信息；这些三元组不仅涉及职业关系，也涉及其他关系（例如出生地、她所使用的语言，或她出演过的电影）。为此，系统会从文本集中收集有关该实体的文本证据，形式为文本中的其他提及。系统根据这些提及推断出三元组，将 x 链接到知识图谱。

我们使用基于嵌入的半归纳式 [Al21] 开放世界 [SW17]链接预测来解决这两项任务，该方法旨在预测三元组（h，r，t）成立的可能性。与标准链接预测不同，半归纳式链接预测旨在处理需要连接到已知图中的新实体，而开放世界场景则意味着新实体仅通过自由文本进行描述。

由于特定领域的数据通常属于机密，链接预测研究会采用来自图谱的开放数据，例如 Freebase [Bo08] 或 Wikidata [VK14]。我们认为，这些基准测试所提供的见解

对于工业知识获取而言十分有限，因此我们推出了一个名为 **文本归纳推理 V2 (IRT2)** 的新基准，具有以下优势[3]：(1) 为了评估图构建不同阶段的模型，我们在不同规模的图上进行基准测试。(2) 其他数据集中的文本上下文由实体的简洁描述构成，而实际文本中往往只是偶然提及实体。我们据此对文本数据进行采样。(3) 据我们所知，我们的研究首次不仅探讨链接任务，也探讨排序任务。

我们在 IRT2 上评估了三种归纳式链接模型：一种基于关键词匹配的基线模型，以及两种将 Transformer 文本编码器与链接预测器相结合的神经模型。我们表明，端到端方法和分开训练都表现良好，而后者通常略胜一筹（即使是在小型图上）。此外，只有在较小图上的评估中，基线模型在链接任务上的表现才逊于神经模型，这表明当结构化数据变得稀疏时，神经模型更具优势。

2 相关工作

链接预测模型： 封闭世界链接预测（也称为知识图谱补全，*KGC*）在过去几年中引起了研究界的关注。RESCAL [NTK11], TransE [Bo13], 和 DistMult [Ya14] 等早期方法为后续的许多补全模型奠定了基础，例如 ComplEx [Tr16], ConvE [De18a], RotatE [Su19],和 KBGAT [Na19]。然而，我们关注的是开放世界场景：在这种场景中，给定某个实体的文本描述，可以通过对稠密的图表示和文本表示进行对齐来开展几何推理。这将知识图谱补全（旨在预测现有实体之间缺失的链接）与语言建模相结合 [Mi13, Bo16, De18b]，后者能够为自然语言文本生成稠密的向量表示。具体而言，本文研究一种半归纳式设置 [Al21]，在该设置中，需要利用实体的文本描述，将知识图谱之外的实体链接到现有图中。最早尝试结合语言推理模型与图推理模型的方法之一是 NTN [So13],，该方法训练实体描述嵌入，使相连实体的嵌入彼此相似。后续方法以此为基础：DKRL [Xi16], ConMask [SW17], MIA [Fu20], 和 KEPLER [Wa21] 提出了联合学习实体表示与图表示之间联系的模型。另一种方法则将知识图谱补全评分器与文本编码器解耦：首先独立训练评分器，然后单独学习从基于文本的实体描述到其对应的基于图的嵌入的投影 [Sh19, SVU20, ZSH20]。我们使用第 4 节所述模型研究并比较了这两种方法。与我们的工作高度相关的是 Daza 等人最近发表的方法 [DCG21]。该方法训练一个 BERT 模型（BLP），使用文本表示直接对知识图谱补全三元组的合理性进行评分，而不是训练单独的实体嵌入。然而，我们无法依赖高质量文本，因此，为实体嵌入提供来自输入文本的多种不同观察并对其进行调优，更为合适。

[3] 这些数据已公开发布，地址为 https://github.com/lavis-nlp/irt2

知识图谱补全基准：封闭世界知识图谱补全模型通常在公开可用知识图谱的子集上进行评估，例如 Freebase [Bo08], DBpedia [Au07] 或 Wikidata [VK14]。著名的基准包括 FB15K [Bo13], , 因测试集泄漏而由 FB15k237 [TC15] 取代；WN18，后由 WN18RR [De18a],取代；以及 CoDEx [SK20],等。开放世界知识图谱补全将知识图谱与其实体的文本描述相结合。相关方法包括 DKRL [Xi16], 中的 FB20K 基准、ConMask [SW17], 中的 DBpedia50k 和 DBpedia500k，以及 OWE [Sh19]中的 FB15k237-OWE。较新的研究引入了 Wikidata5M [Wa21]（Wikidata 的一个大型子集），以及 [DCG21]的 FB15k237/WN18RR 适配版本。由于工业应用场景中的数据并不公开，我们也从开放知识图谱（CoDEx）和文本来源（维基百科）中抽样构建基准数据。然而，上述所有基准都通过每个图实体的一条简短描述来处理开放世界场景。我们关注的则是将实体与多个含有噪声的文本上下文关联起来。据我们所知，唯一尝试实现这一点的基准是 IRT [Ha21]。在该基准中，知识图谱中的每个实体都关联一组 30 个包含偶然提及的文本上下文。我们扩展了这一基准，提供更多文本、多种图规模、更多样的提及，以及用于排序和链接的评估协议。

3 问题描述

我们将知识图谱（KG）定义为一个有向图 $G = (V, R, T, M, C)$，其中 V 是顶点集合 v（例如 v = Fargo），R 是关系类型集合（例如 r = 类型）。三元组 $(h, r, t) \in T \subseteq V \times R \times V$ 将顶点连接起来，例如 (Fargo, 类型, 惊悚电影)。此外，集合 M 包含文本提及，每个提及都通过函数 M 与一个顶点相关联：$V \mapsto P(M)$。例如，实体 v=惊悚电影有以下提及 $M(v)$ = {“犯罪惊悚片”，.....，“黑帮电影”}。请注意，提及可能存在歧义（例如，“这部电影”可以是对 Fargo 的提及）。此外，假设每个提及m 都出现在语料库 C 中的文本上下文句子里 c。通常，这些上下文并不表达三元组，而仅包含对实体的偶然提及。我们可以通过函数 C 来访问这些上下文：$V \times M \mapsto P(C)$。例如，对于 v = 犯罪电影和 m =“黑帮电影”：$C(v, m)$ = {“Corman 称它是最准确、最真实的黑帮电影”,.....}。请注意，通过 M 和 C，我们既可以访问实体的提及和上下文，也可以反向访问。

开放/封闭世界：文本语料库 C 包含已知（或称“封闭世界”）实体，而第二个语料库 Q 则包含尚未发现的开放世界提及 M^o。我们的目标是发现这些提及，并将它们链接到图中。我们定义一个封闭世界图 G^c（包含 G 和文本语料库 C 中相应集合的子集），以及一个开放世界图 G^o =（V^o、R、T^o、M^o、Q），其中 $V^o \subseteq V$、$T^o \subseteq T$、$M^o = M \setminus M^c$：也就是说，这是一个包含尚未发现提及的图，而这些提及在封闭世界图 G^c中并不为人所知。由此引出我们要解决的两个任务：

排序任务：第一个任务是从 Q 中检索文本上下文，其中包含我们感兴趣的提及，即对（可能尚未出现的）实体的未知提及；这些提及应能补全给定的实体-关系对。例如，对于关系 r = 类型 和尾实体 t = 犯罪电影，我们会搜索提及 任意犯罪电影的文本上下文。

链接任务： 现在，给定一个刚被发现并标记的开放世界提及，Q 的所有文本上下文均按提及打包。对于每一组这样的文本， $r \in \mathcal{R}$ 都必须找到与之链接的图顶点 V^c 。例如：*"与所有包含提及 fargo 的文本上下文相关联的类型是什么？"*。

---

4 模型

我们用三个模型来解决上述任务：**(1) JOINT**，将文本编码器与知识图谱补全（KGC）评分器联合训练；**(2) OWE**，由编码器学习将文本表示投影到预训练的图嵌入空间中；**(3)BOW**，一种采用词袋表示进行文本相似度计算的模型。我们在排序任务和链接任务上评估这三个模型。请注意，尽管为简洁起见，我们的记号通常侧重于查找尾实体，但对于给定的关系-尾实体对，我们也始终纳入头实体预测。

4.1 神经模型

这两种神经模型都结合了文本编码器 ϕ 和知识图谱补全模块 ψ （见图 2）。作为文本编码器，我们使用预训练的 BERT [De18b], , 这是一种流行的先进 Transformer [Va17]编码器。给定文本上下文 c，我们将其输入编码器，并选取 [CLS]- 标记嵌入作为上下文表示 $\phi(c)_{\text{CLS}} \in \mathbb{R}^{d'}$ （遵循 [Ha21]的观察结果）。使用学习得到的仿射投影（$\mathbf{W}$,$\mathbf{b}$），将该表示映射到复值空间，从而得到表示 $\mathbf{c}$。随后使用该表示估计三元组的合理性。我们使用 ComplEx 评分器 [Tr16], , 它根据复值的上下文、关系和尾实体嵌入 $\mathbf{c}$、$\mathbf{r}$、$\mathbf{v}$ $\in \mathbb{C}^d$计算合理性评分：$\psi(c, r, v)$ =Re($\mathbf{c} \odot \mathbf{r} \odot \mathbf{v}$)。请注意，由于投影将实数值映射到复数空间，因此我们会相应地选择矩阵和偏置向量的维度（$W \in \mathbb{R}^{d' \times 2d}$, $\mathbf{b}$ $\in \mathbb{R}^{2d}$）。总的来说，我们的神经模型可写为：

$$s(c, r, v) = \psi(\underbrace{W \cdot \phi(c)_{\text{CLS}} + \mathbf{b}}_{\mathbf{c}}, \mathbf{r}, \mathbf{v}) \tag{1}$$

多上下文扩展如上所述，同一实体可能有多个上下文 c，而非单一上下文。我们希望实体表示能够考虑整个上下文集合 Σ 。为此，我们采用一种简单的早期融合方法：先对文本表示求平均，再进行投影： Σ

$$\mathbf{c} = W \cdot \left(\frac{1}{|\Sigma|} \sum_{c \in \Sigma} \phi(c)_{\text{CLS}} \right) + \mathbf{b} \tag{2}$$

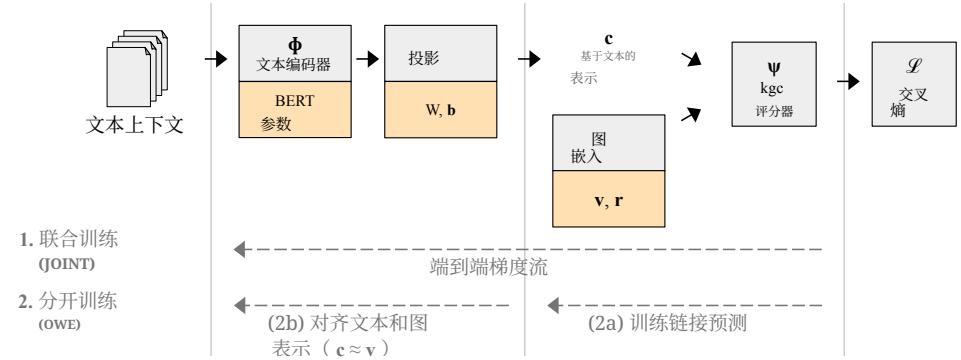


图2：我们模型（上方）和两种训练策略（下方）中的数据流：策略1（JOINT）对链接预测器和文本编码器进行端到端训练。策略2（OWE）先训练链接预测器（2a），然后将基于文本的实体表示**c** 与基于图的实体表示**v** 对齐（2b）。可学习参数以橙色突出显示。

然后，我们使用得到的表示 **c** 作为公式（1）的直接替代项。关于训练，我们采用此设置的两种变体：第一种变体（称为**单一**）在单个上下文上进行训练，如公式（1）所示，但在推理时应用公式（2）。第二种变体（称为**多重**）则在训练和推理时都应用公式（2）。

训练

如图2所示，我们研究了两种不同的训练策略，用于拟合模型参数（文本编码器 ϕ、投影 W **b** 和知识图谱嵌入 **r**，**v**）。我们的第一种方法采用联合（端到端）训练（**JOINT**）：我们迭代地从知识图谱中随机采样一个实体 v。对于 v，我们随机抽取一个三元组（v，r，v'）和上下文 c。模型的得分 s（c，r，v'）通过 softmax 映射为概率分布，并使用交叉熵损失：

$$\mathcal{L}_{\text{JOINT}} = -\sum_{v' \in \mathcal{V}^c} \left(\log \frac{\exp(s(c,r,v'))}{\sum_{u' \in \mathcal{V}^c} \exp(s(c,r,u'))} \right) \cdot \mathbf{1}_{(v,r,v') \in \mathcal{T}} \tag{3}$$

请注意，尽管上述记法是对三元组（v，r，v'）进行采样以预测尾实体 v'，我们也会对三元组（v'，r，v）进行采样，并相应地预测头实体 v'。

我们的第二种方法遵循 Shah 等人的**OWE** 模型 [Sh19]，并采用**分开训练**。训练分为两个步骤：首先，在封闭世界图上训练链接预测模型 ψ，获得一个基于图的嵌入空间。其次，训练文本编码器ϕ 和投影 **(W, b)**，使实体的基于文本的表示与（此时已固定的）基于图的表示对齐。为此，我们通过均方误差损失，减小上下文

表示 $\mathbf{c} = W \cdot \phi(c)_{CLS} + \mathbf{b}$ 与其对应的基于图的表示之间的几何距离：

$$\mathcal{L}_{\text{OWE}} = \frac{1}{2d} \cdot \sum_{0<i\leq d} \left(\mathbf{Re}(\mathbf{c}_i) - \mathbf{Re}(\mathbf{v}_i)\right)^2 + \left(\mathbf{Im}(\mathbf{c}_i) - \mathbf{Im}(\mathbf{v}_i)\right)^2 \tag{4}$$

推理：本节概述了如何在我们的两项任务中使用神经模型。首先，在**排序**任务中，给定一个封闭世界实体 v 和一个感兴趣的关系 r，任务是从查询语料库 Q 中检索一个文本上下文排序列表，这些上下文可能包含相关开放世界实体的提及，供专家识别。为此，我们计算所有上下文 $c \in Q$、$r \in \mathrm{R}$ 和 $v \in \mathrm{V}^c$ 的单一上下文分数 $s(c, r, v)$。我们按关系使用 softmax 函数对分数进行归一化，以便更好地比较独立评分的上下文 c（原始 ComplEx 分数没有界限）。随后，根据归一化分数对查询语料库中的上下文进行排序。

其次，在**链接**任务中，专家关注的是一个特定的感兴趣提及 $m \in \mathrm{M}^o$，它代表一个开放世界实体。其目标是将该实体链接到图中。为此，专家选择一个关系 $r \in \mathrm{R}$，并收集一组包含 m 的上下文 Σ，以描述该开放世界实体。利用这些上下文，所有实体候选项 $v \in \mathrm{V}^c$ 都会根据其分数 $s(\Sigma, r, v)$ 进行排序。请注意，原则上，我们可以针对每种关系重复此过程，也可以预测头实体而非尾实体。

4.2 BOW：词袋检索

为了使用一种通用的工业方法评估当前任务，我们将上述神经模型与词袋方法进行比较。给定一个或多个提及，该方法会将包含这些提及的文本上下文串联成文档，然后使用 Elasticsearch 搜索引擎实现的知名 BM25[RW94] 关键词匹配，在文档之间进行相似度匹配[4]。

排序：给定一个实体关系对，例如 v = 惊悚电影，r = 类型（“查找提及惊悚电影的文本上下文”），基线策略是寻找与已知封闭世界实体相似的开放世界实体 v'，对于这些实体，(v', r, v) 成立（即已知的惊悚电影）。我们抽取一组随机代表实体 v' 及其部分关联文本上下文。将这些上下文串联成一篇文档，然后用作查询，在包含所有开放世界上下文Q的索引中进行检索。

链接：给定一个提及 $m \in \mathrm{M}^o$ 和一个关系 r，我们采样包含 m 的上下文，并将其置于文档 $D(m)$ 中；随后使用该文档针对一个包含文档 $D(v)$ 的索引进行查询，以获取所有封闭世界顶点 $v \in \mathrm{V}^c$。前 n 个预测结果使我们能够汇编出一组

[4] https://www.elastic.co

参考顶点，我们将其用作预测的蓝图。例如：搜索 r = 文本上下文的职业，其中包含 *"s. a. corey"* ("预测包含 *s. a. corey* 的文本上下文的职业")时，结果列表可能包含其他描述作者的文本。顶点的最终预测由检索到的关系目标顶点 r 汇编而成，并根据文档在结果列表中的位置的倒数进行评分。

5 基准构建

我们的基准承接了 [Ha21] 的工作并对其进行了扩展：(1) 我们研究模型在不同知识图谱规模下的表现，尤其是小型图；(2) 我们测试模型在排序和链接任务上的表现。我们提供了四种基准变体：**极小**、**小型**、**中**和**大**。这些变体在结构信息（即图三元组）与关联文本的比例上各不相同。数据集基于 CoDEx-M [SK20]，该数据集从 Wikidata [VK14]中采样而来。具体执行以下步骤：

1. 确定概念顶点：我们假设，在现实场景中，世界知识更有可能已经在给定的知识图谱中建模。待发现的信息通常具有更强的时效性。例如，考虑机器制造商的图谱。我们知道物品可能会起火（这是世界知识），但我们的目标是找出实际会起火的具体部件。识别此类世界知识的一种启发式方法基于以下假设：这类实体的特征是其关系的头实体与尾实体数量严重不均衡。关系 r'的定义域大小（即头实体数量）$\mathrm{dom}(r) := |\{h \mid \exists t : (h, r, t) \in \mathrm{T}\}|$ 与值域大小（即尾实体数量）$\mathrm{rg}(r) := |\{t \mid \exists h : (h, r, t) \in \mathrm{T}\}|$ 之比为 $\min(\mathrm{dom}(r), \mathrm{rg}(r))/\max(\mathrm{dom}(r), \mathrm{rg}(r))$。针对数据集的每种规模变体，我们从 CoDEx 提供的关系中选取一个子集，按比例排序后再选取其中一部分，以确定将在封闭世界划分中保留的概念实体。附录第9节详细列出了所有关系及相应的选择。

2. 采样提及和文本上下文：由于我们的图基于 Wikidata，我们可以利用它与维基百科的链接。对于每个顶点 v，我们确定其对应的维基百科页面 p_v 以及所有链接回该页面的页面 $\mathcal{N}(p_v)$。利用超链接的链接文本，我们构建与顶点相关联的提及集合 $M(v)$。最后，为了构建上下文 $C(v, m)$，我们找出 $\mathcal{N}(p_v)$ 中所有提及的出现位置，并将其所在的句子保留为上下文。这是一种获取附带文本上下文的启发式方法：句子中提到了该实体，但它通常不是句子的主题。以 v = 苏格兰为例："阿伯丁是苏格兰东北部的一座城市"。

3. 开放世界/封闭世界划分：在这种情况下，开放世界/封闭世界划分不是按顶点进行，而是按提及级别进行。我们将提及集合 M 划分为 M^c（封闭世界提及）和 M^o（开放世界提及）。首先，将所有与概念实体相关的提及划入封闭世界部分。然后，对于每个剩余顶点 v，将其相关提及随机分配到开放世界和封闭世界。我们设置一个

额外的剪枝参数，用于限制封闭世界顶点允许包含的提及数量上限。根据封闭世界提及的信息，我们找出所有顶点与这些提及相关联的三元组，并将它们划入封闭世界三元组集 $\mathrm{T}^{c} \subseteq \mathrm{T}$。所谓的测试三元组（$m \in \mathrm{M}^{o}$，$r \in \mathrm{R}$，$v \in \mathrm{V}^{c}$）（见下方表格中的说明）是通过确定提及所属顶点出现在哪些三元组中，直接从三元组集 T 中推导出来的。我们从开放世界划分中划出一个子集作为验证集。上述步骤针对四种不同的数据集变体分别执行。这些变体在概念关系、保留的关系总数以及封闭世界提及剪枝方面采用了不同的参数设置（更多详情见附录）。

	Tiny	**小型**	**中**	**大**
关系 $\lvert\mathrm{R}\rvert$	5	12	45	45
实体 $\lvert\mathrm{E}\rvert$	1,174	2,887	3,592	9,952
训练三元组 $\lvert\mathrm{T}^{c}\rvert$	2,928	7,527	26,335	102,289
训练文本 $\lvert\mathrm{C}\rvert$	9,100,422	15,117,184	17,398,943	18,654,485
测试文本 $\lvert\mathrm{Q}\rvert$	6,058,557	2,390,017	1,905,151	864,598
排名查询	695	1,225	4,972	7,336
链接查询	47,857	52,654	97,921	48,801
测试三元组	102,866	102,616	174,312	90,780

表格的最后四行描述了这些挑战的规模。排序查询是一个元组 $(v \in \mathrm{V}^{c}, r \in \mathrm{R})$（例如 (writer, profession)——“查找包含作家的文本”），以及一个相关的开放世界提及真实标注集合 $\{m_1, \ldots\} \in M^{o}$，其中的提及有待发现（例如 { “托尔金”, “这位作家”, “Corey”, . . . }）。链接查询是一个元组 $(m \in M^{o}, r \in \mathrm{R})$（例如 (“托尔金”, profession)——“根据出现‘托尔金’的文本，这个实体有哪些职业？”），以及一个相关的封闭世界顶点集合 $v \in \mathrm{V}^{c}$（例如 { 作家, 语言学家, . . . }）。可以看出，排序和链接是对同一数据的不同视角：唯一的 $(m \in M^{o}, r \in \mathrm{R}, v \in \mathrm{V}^{c})$ 三元组。这些三元组作为测试三元组列于上表，可用于衡量查询及其相关真实标注的比例。例如，在最极端的情况下，极小变体上的排序平均每个查询需要发现 102866/695 ≈ 148 条提及。

6 实验

我们开展了一系列实验，以深入了解这一挑战的难度。在所有四种数据划分（极小、小型、中和大）上，我们针对两项任务（排序、链接）使用此前介绍的全部三种模型（JOINT、OWE、BOW）。我们针对每个测试三元组（即微平均）计算 hits@k 和平均倒数排名（MRR）。按照常见做法，我们对评分后的预测（排序任务中的提及和排序任务中的顶点）应用目标过滤 [Sh19] 。目标过滤是一种方法：对于给定的排序，只检查一个感兴趣的真阳性，同时从列表中移除其他所有真阳性。这样可以避免因某些排序包含大量真阳性而人为地降低结果。

我们在线提供数据集、数据集创建代码和评估协议[5]。模型实现以及所有已训练模型也另行提供[6]。我们在测试文本上下文 Q 的一个子集上评估模型（用于排序时，随机抽取 40 万个句子；用于链接时，每个提及抽取 100 个句子）。

6.1 超参数

我们通过在参数空间中结合随机搜索和网格搜索来确定最终的超参数。对于每个数据集划分，我们都会训练一个单一的封闭世界 ComplEx 模型，并通常使用 Adagrad [DHS11]来训练 **OWE** 模型。所研究的超参数包括学习率、L2 正则化权重和嵌入空间维度。我们使用 PyKEEN[Al20] 进行训练，并自行构建开放世界扩展。我们使用 PyTorch [Pa19]和 PyTorch Lightning[FT19]实现并训练 **JOINT** 和 **OWE** 模型。模型使用 Adam[KB14]进行优化。对于每种模型架构和数据集划分，我们都会进行随机参数扫描，以确定一部分参数，并对其余参数进行网格搜索。我们研究实体嵌入和关系嵌入的正则化、优化器的权重衰减和学习率、训练期间冻结编码器多少层、每次训练时每个顶点采样多少个文本上下文，以及训练期间是否对提及进行掩码。此外，我们还研究多上下文模型在每个优化器步骤中应提供多少个上下文。总体而言，所有模型的学习率、权重衰减和正则化权重都采用相似的取值范围。其他参数则取决于数据集划分。我们发现，冻结编码器的部分层有助于防止过拟合。与 [Ha21], 的观察相反，掩码并未显著提升模型性能。最终的超参数见附录（第9节）以及在线发布的模型。

6.2 链接

首先，我们将链接任务作为所提出模型的指定挑战进行评估。表 1 详细列出了我们的发现。总体而言，可以观察到以下几种趋势：(1) 神经模型在三个较小的变体上显著优于基线方法，但在大型数据集上的结果较差。这支持了我们的观点：基准应能让人们了解结构化数据稀缺程度的差异。即使数据丰富（这通常有利于神经网络方法），BOW 基线在大型数据集上的表现也能胜过神经模型，但在较小的变体上则不及神经模型。(2) 总体而言，OWE 方法优于 JOINT，不过在极小型、小型和大型数据集上的优势并不明显。两者表现差距最大的是中型数据集，OWE 模型从解耦训练中获益最多。我们认为，这表明 JOINT 在训练期间需要更好的采样／过拟合策略

[5] https://github.com/lavis-nlp/irt2[6]
https://github.com/lavis-nlp/irt2m

，因为它似乎难以应对训练期间见到的提及种类较少的情况。(3) 尽管多上下文模型的表现通常优于单上下文对应模型，但性能差距并不像 [Ha21] 报告的那样显著。与他们的发现相反，我们也没有观察到掩码提及能显著提升性能。这些观察结果表明，两种模型都难以充分利用文本线索，而是严重依赖顶点和关系嵌入中包含的结构信息。总体而言，各项指标显示，这些模型能够以较高的精确率，将由文本上下文描述的实体链接到给定图中。

		HITS@10				MRR			
		Tiny	**小型**	**Med.**	**大**	**Tiny**	**小型**	**Med.**	**大**
BOW		53.82	55.18	46.43	**71.38**	33.63	34.62	29.81	**50.61**
JOINT	单一	72.06	70.20	47.14	65.75	50.61	45.95	33.72	48.29
JOINT	多重	73.56	74.27	53.77	65.12	51.28	**52.39**	37.50	45.26
OWE	单一	74.09	**74.33**	61.98	64.27	50.25	50.57	40.60	42.69
OWE	多重	**75.39**	71.49	**64.41**	66.36	**53.06**	47.17	**43.25**	45.51

表 1：模型在**链接** 任务上的表现。词袋模型在大型数据集上的表现优于神经网络方法。在较小的数据集变体上，两种神经模型的表现都大幅优于基线。在神经模型中，OWE 通常在所有数据划分上表现更好。

6.3 排序

其次，对于排序，我们研究最初为执行链接预测而训练的模型是否也可以用于发现。我们在表 2 中报告 HITS@100 的表现。(1) 结果表明，神经模型的评分机制优于 BM25 搜索结果。乍看之下，这是一个令人意外的发现，但可以从文本样本的特性解释。由于感兴趣的提及通常不是文本上下文的主题，查询并不会直接描述其属性。这会使排序器偏离方向。例如：搜索其他喜剧演员（即 ?，职业，喜剧演员），模型会使用从维基百科上关于纽约的页面采样得到的文本样本等。因此，检索到的其他文本样本也会谈及纽约上下文中的事物，却很少涉及其他喜剧演员。神经模型更能考虑查询的这种上下文化（依据查询的关系）。(2) 总体而言，排序质量还不足以满足实际应用的要求。

我们认为有两个主要因素阻碍了有效排序：首先，各个分数是彼此独立计算的，因此难以在同一样本内对分数进行比较。其次，排序任务首先要求分数由头实体预测产生。这些分数通常质量较低，因为模型更擅长预测少数有偏差的尾实体（例如职业），而非数百或数千个多变的头实体（例如

		HITS@100			
		Tiny	**小型**	**Med.**	**大**
BOW		2.86	4.29	6.42	14.83
JOINT	单一	7.91	6.78	6.37	19.47
JOINT	多重	**13.28**	**16.17**	**14.38**	30.68
OWE	单一	6.30	8.19	6.88	10.81
OWE	多重	9.98	13.00	6.36	**31.40**

表 2：**排序** 任务的结果。所有神经模型的表现均优于基线；与链接任务相反，多上下文 JOINT 模型通常是表现最佳的方法。

作者）。前文所述的观察结果进一步加剧了这一问题：文本上下文对分数的影响相对较小。为了取得更好的性能，必须设计新的方法，一方面纳入图上下文，另一方面结合神经语义检索 [MC17, LNY21] 。

7 结论

我们提出 IRT2，这是一个用于研究如何从噪声文本扩展知识图谱（KG）的知识获取流程的基准。我们提供了不同规模的知识图谱，以及与其顶点相关联的文本。该基准包含两项任务：*排序* 和*链接*。首先，需要从未标注文本中发现实体；在第二阶段，再将它们链接到知识图谱。此外，我们还研究了三种用于这些任务的方法：两种神经开放世界三元组评分器和一种词袋模型。我们发现，这些模型在链接任务上的表现良好，但在排序任务上的表现较差。在链接任务中，BOW 基线在大型版本上优于神经模型。对于其余较小的版本，即使图数据稀缺，OWE 方法也优于其他所有方法。在排序任务中，JOINT 模型取得了最佳结果。尽管这些模型展现出用于排序的良好前景，但其表现尚不足以满足实际应用的要求。我们建议研究如何利用图数据与文本数据之间已有的链接，提升语义排序技术的效果。我们邀请社区使用该基准：（1）探索如何更好地利用数量丰富但含有噪声的文本数据，从而改进预测；（2）探索如何结合特定的图信息，为语义检索器提供上下文；（3）在具有类似约束和条件的工业环境中，发掘应用方面的见解。数据集和评估脚本可在此处获取：`https://github.com/lavis-nlp/irt2`。

致谢：本研究由 BMBF 的 FH-Kooperativ 计划项目 SCENT（13FH003KX0）资助。

8 参考文献

[Al20] Ali, Mehdi；Berrendorf, Max；Hoyt, Charles Tapley；Vermue, Laurent；Shar- ifzadeh, Sahand；Tresp, Volker；Lehmann, Jens：PyKEEN 1.0：用于训练和评估知识图谱嵌入的 Python 库。arXiv 预印本 arXiv:2007.14175，2020年。[Al21] Ali, Mehdi；Berrendorf, Max；Galkin, Mikhail；Thost, Veronika；Ma, Tengfei；Tresp, Volker；Lehmann, Jens：利用超关系事实改进归纳式链接预测。载于：国际语义网会议。Springer，第 74–92 页，2021年。[Au07] Auer, Sören；Bizer, Christian；Kobilarov, Georgi；Lehmann, Jens；Cyganiak, Richard；Ives, Zachary：DBpedia：开放数据网络的核心。载于：语义网，第 722–735 页。Springer，2007年。[Bo08] Bollacker, Kurt；Evans, Colin；Paritosh, Praveen；Sturge, Tim；Taylor, Jamie：Freebase：一个协作创建的图数据库，用于构建人类知识。载于：2008 年 ACM SIGMOD 数据管理国际会议论文集。SIGMOD '08，ACM，美国纽约州纽约市，第 1247–1250 页，2008年。[Bo13] Bordes, Antoine；Usunier, Nicolas；Garcia-Duran, Alberto；Weston, Jason；Yakhnenko, Oksana：用于建模多关系数据的平移嵌入。载于：神经信息处理系统进展。第 2787–2795 页，2013年。[Bo16] Bojanowski, Piotr；Grave, Edouard；Joulin, Armand；Mikolov, Tomas：利用子词信息丰富词向量。CoRR abs/1607.04606。网址：http://arxiv. org/abs/1607.04606，2016年。[DCG21] Daza, Daniel；Cochez, Michael；Groth, Paul：通过链接预测从文本中归纳实体表示。载于：2021 年万维网会议论文集。第 798–808 页，2021年。[De18a] Dettmers, Tim；Minervini, Pasquale；Stenetorp, Pontus；Riedel, Sebastian：卷积式二维知识图谱嵌入。载于：第三十二届 AAAI 人工智能会议。2018年。[De18b] Devlin, Jacob；Chang, Ming-Wei；Lee, Kenton；Toutanova, Kristina：BERT：用于语言理解的深度双向变换器预训练。arXiv 预印本 arXiv:1810.04805，2018年。[DHS11] Duchi, John；Hazan, Elad；Singer, Yoram：用于在线学习和随机优化的自适应次梯度方法。《机器学习研究期刊》，12(7)，2011年。[FT19] Falcon, William；PyTorch Lightning 团队：PyTorch Lightning。2019年3月。

[Fu20] Fu，Chenpeng；Li，Zhixu；Yang，Qiang；Chen，Zhigang；Fang，Junhua；Zhao，Peng- peng；Xu，Jiajie：用于开放世界知识图谱补全的多重交互注意力模型。载于：国际网络信息系统工程会议。Springer，第 630–644 页，2020年。[Ha21] Hamann，Felix；Ulges，Adrian；Krechel，Dirk；Bergmann，Ralph：用于知识发现的开放世界知识图谱补全基准测试。载于：应用智能系统工业、工程及其他应用国际会议。Springer，第 252–264 页，2021年。[Ho21] Hogan，Aidan；Blomqvist，Eva；Cochez，Michael；d'Amato，Claudia；Melo，Gerard de；Gutierrez，Claudio；Kirrane，Sabrina；Gayo，José Emilio Labra；Navigli，Roberto；Neumaier，Sebastian 等：知识图谱。《数据、语义与知识综合讲座》，12(2):1–257，2021年。[HSG15] Hoehndorf，Robert；Schofield，Paul N.；Gkoutos，Georgios V.：本体在生物学与生物医学研究中的作用：功能视角。《生物信息学简报》，16(6):1069–1080，2015年04月。[KB14] Kingma，Diederik P；Ba，Jimmy：Adam：一种随机优化方法。arXiv 预印本 arXiv:1412.6980，2014年。[LNY21] Lin，Jimmy；Nogueira，Rodrigo；Yates，Andrew：用于文本排序的预训练变换器：BERT 及更多。《人类语言技术综合讲座》，14(4):1–325，2021年。[MC17] Mitra，Bhaskar；Craswell，Nick：用于信息检索的神经模型。arXiv 预印本 arXiv:1705.01509，2017年。[Mi13] Mikolov，Tomas；Sutskever，Ilya；Chen，Kai；Corrado，Greg S；Dean，Jeff：词语和短语的分布式表示及其组合性。载于：神经信息处理系统进展，第 3111–3119 页，2013年。[Na19] Nathani，Deepak；Chauhan，Jatin；Sharma，Charu；Kaul，Manohar：学习用于知识图谱关系预测的基于注意力的嵌入。arXiv 预印本 arXiv:1906.01195，2019年。[NTK11] Nickel，Maximilian；Tresp，Volker；Kriegel，Hans-Peter：用于多关系数据集体学习的三向模型。载于：国际机器学习会议。第 11 卷，第 809–816 页，2011年。[Pa19] Paszke，Adam；Gross，Sam；Massa，Francisco；Lerer，Adam；Bradbury，James；Chanan，Gregory；Killeen，Trevor；Lin，Zeming；Gimelshein，Natalia；Antiga，Luca；Desmaison，Alban；Kopf，Andreas；Yang，Edward；DeVito，Zachary；Raison，Martin；Tejani，Alykhan；Chilamkurthy，Sasank；Steiner，Benoit；Fang，Lu；Bai，Junjie；Chintala，Soumith：PyTorch：一种命令式风格的高性能深度学习库。载于（Wallach，H.；Larochelle，H.；Beygelzimer，A.；

d'Alché-Buc，F.；Fox，E.；Garnett，R.，编）：神经信息处理系统进展32，第 8024–8035 页。Curran Associates, Inc.，2019年。[RW94] Robertson，Stephen E；Walker，Steve：对概率加权检索二泊松模型的一些简单有效近似。载于：第 17 届 ACM SIGIR 信息检索研究与发展国际年会论文集。Springer-Verlag New York, Inc.，第 232–241 页，1994年。[Sh19] Shah，Haseeb；Villmow，Johannes；Ulges，Adrian；Schwanecke，Ulrich；Shafait，Faisal：知识图谱补全模型的开放世界扩展。载于：第三十三届 AAAI 人工智能会议。2019年。[SK20] Safavi，Tara；Koutra，Danai：CoDEx：综合知识图谱补全基准。载于：2020年自然语言处理实证方法会议（EMNLP）论文集，第 8328–8350 页，2020年。[So13] Socher，Richard；Chen，Danqi；Manning，Christopher D；Ng，Andrew：利用神经张量网络进行知识库补全推理。载于：神经信息处理系统进展，第 926–934 页，2013年。[Su19] Sun，Zhiqing；Deng，Zhi-Hong；Nie，Jian-Yun；Tang，Jian：RotatE：通过复数空间中的关系旋转实现知识图谱嵌入。arXiv 预印本 arXiv:1902.10197，2019年。[SVU20] Shah，Haseeb；Villmow，Johannes；Ulges，Adrian：用于开放世界知识图谱补全的关系特定变换。载于：基于图的自然语言处理方法（TextGraphs）论文集。计算语言学协会，西班牙巴塞罗那（线上），第 79–84 页，2020年12月。[SW17] Shi，Baoxu；Weninger，Tim：开放世界知识图谱补全。arXiv 预印本 arXiv:1711.03438，2017年。[TC15] Toutanova，Kristina；Chen，Danqi：知识库与文本推理中的观测特征与潜在特征。载于：第三届连续向量空间模型及其组合性研讨会论文集。计算语言学协会，中国北京，第 57–66 页，2015年7月。[Tr16] Trouillon，Théo；Welbl，Johannes；Riedel，Sebastian；Gaussier，Éric；Bouchard，Guillaume：用于简单链接预测的复数嵌入。载于：国际机器学习大会，第 2071–2080 页，2016年。[Va17] Vaswani，Ashish；Shazeer，Noam；Parmar，Niki；Uszkoreit，Jakob；Jones，Llion；Gomez，Aidan N；Kaiser，Łukasz；Polosukhin，Illia：注意力机制就是你所需要的一切。载于：神经信息处理系统进展，第 5998–6008 页，2017年。

[VK14] Vrandečić, Denny；Krötzsch, Markus：《Wikidata：一个免费的协作式知识库》。《ACM通信》，57(10):78–85，2014年。[Wa21] Wang, Xiaozhi；Gao, Tianyu；Zhu, Zhaocheng；Zhang, Zhengyan；Liu, Zhiyuan；Li, Juanzi；Tang, Jian：《KEPLER：一种统一的知识嵌入与预训练语言表示模型》。《计算语言学协会会刊》，9:176–194，2021年。[Xi16] Xie, Ruobing；Liu, Zhiyuan；Jia, Jia；Luan, Huanbo；Sun, Maosong：《利用实体描述进行知识图谱表示学习》。载于：第三十届AAAI人工智能会议，2016年。[Ya14] Yang, Bishan；Yih, Wen-tau；He, Xiaodong；Gao, Jianfeng；Deng, Li：《为知识库中的学习与推理嵌入实体与关系》。arXiv 预印本，arXiv:1412.6575，2014年。[ZSH20] Zhou, Yueyang；Shi, Shumin；Huang, Heyan：《开放世界知识图谱补全的加权聚合器》。载于：国际计算机科学家、工程师与教育工作者先驱会议。Springer，第283–291页，2020年。

9 附录

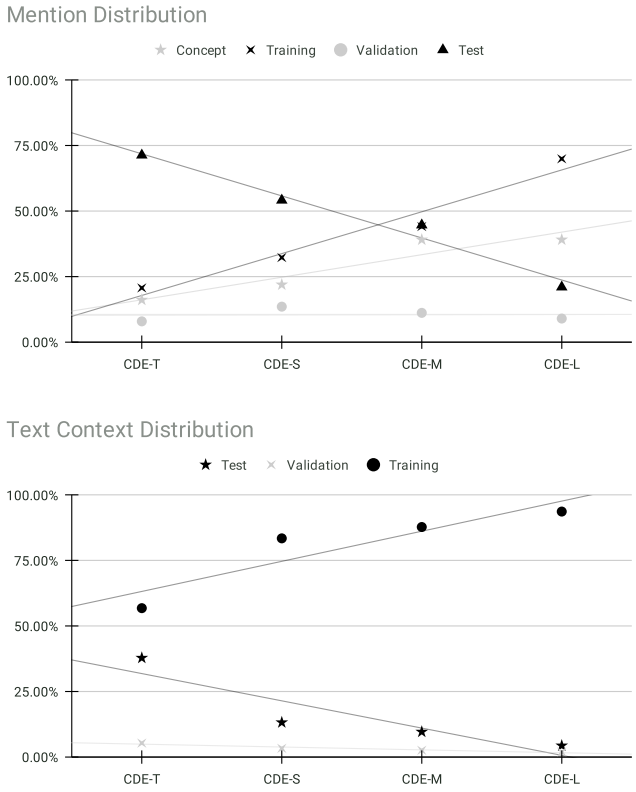
本附录详细介绍了基准构建和模型训练所用的配置选项，以便复现所报告的结果。

9.1 基准

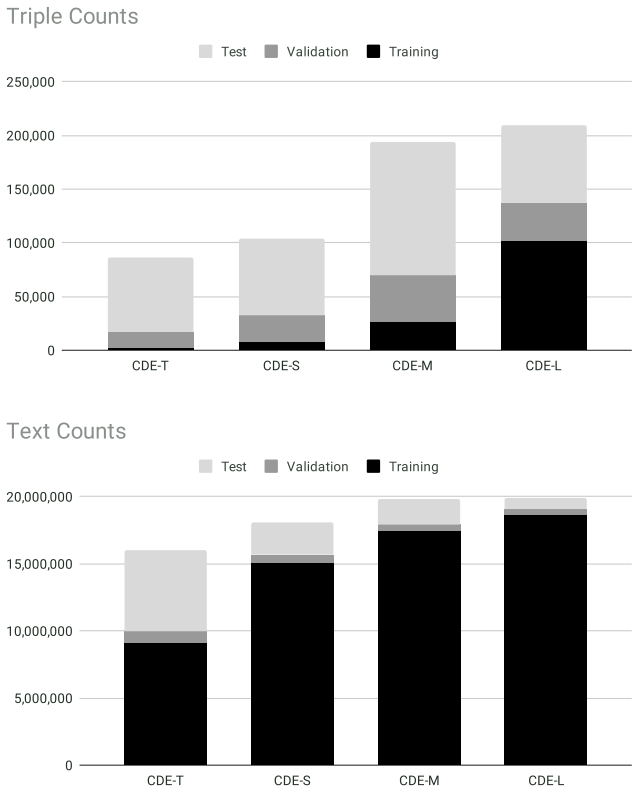
划分	Tiny	小型	中	大
概念关系	4	8	27	27
关系总数	5	12	45	45
封闭世界阈值	400	800	800	-
目标提及划分	70%	70%	70%	70%
目标验证集划分	10%	20%	20%	80%
提及阈值	5	5	5	5
概念顶点	471	962	2,200	2,200
训练顶点	1,174	2,887	3,592	9,952
训练提及	4,945	9,231	14,417	22,866
训练三元组	2,928	7,527	26,335	102,289
训练文本上下文	9,100,422	15,117,184	17,398,943	18,654,485
验证顶点	1,741	3,375	3,279	2,644
验证集提及	1,894	3,870	3,649	2,940
验证顶点三元组	13,639	24,764	43,532	35,209
验证任务三元组	11,588	25,722	43,716	38,582
验证文本上下文	850,351	597,050	505,979	383,351
测试顶点	10,909	10,527	10,221	5,607
测试提及	17,055	15,481	14,600	6,860
测试顶点三元组	70,094	71,360	124,050	71,951
测试任务三元组	102,866	102,616	174,312	90,780
测试文本上下文	6,058,577	2,390,017	11,705,151	864,598

最终基准比例在很大程度上取决于所选关系和构建时采用的剪枝选项。下表列出了用于采样的配置，以及训练集、验证集和测试集划分的关键结果。每个划分中的关系均由人工挑选，按其比例排序，并从中选取一部分作为概念关系。*封闭世界阈值* 决定了在封闭世界划分中，每种关系最多保留多少条提及。*提及阈值* 表示只保留至少关联了这么多条文本上下文的提及。*验证集-* 和测试*顶点三元组* 是最终任务三元组 的来源三元组集（顶点、关系、顶点）；最终任务三元组的形式为（提及、

关系、顶点）。通过筛除确属 开放世界顶点的项，对它们进行剪枝（所需的零样本实体链接不属于本文所述任务的范畴）——这也解释了为何极小变体的数量会减少。



上图详细展示了不同数据集变体之间比例的变化。随着训练数据增加，测试提及数量按比例减少。概念提及数量增加，且在 M 和 L 中相同。随着数据集规模增大，概念提及所占比例下降。相应地，验证文本上下文数量保持不变，而训练数据的增加会减少可用的测试数据。



上方的柱状图报告了三元组的绝对数量（在前一张表格中称为顶点三元组 ）及其关联文本。由此得到的三元组划分与提及划分的变化相对应，训练/测试数据之间呈比例变化。T 到 L 之间的三元组总数增加了一倍以上。关联文本数据随着封闭世界提及数量的增加而增加。

下表按第 5 节所述的比例列出了 CoDEx 中存在的所有关系。右侧详细说明了哪些数据集变体中存在哪些关系。最右侧显示相应关系是否被选中以保留在数据集中。左侧标记了该关系是否被选中用于选取概念实体。

ID	关系	比例	头实体	尾实体	三元组	T		S		M		L	
P1412	使用的语言 (...)	6.32e-3	9,816	62	12,584	X	X	X	X	X	X	X	X
P1303	仪器	1.05e-2	3,622	38	6,076					X	X	X	X
P140	宗教	1.59e-2	2,520	40	2,651			X	X	X	X	X	X
P27	公民身份国家	1.69e-2	13,036	220	16,828		X	X	X	X	X	X	X
P30	大陆	1.98e-2	353	7	391	X	X	X	X	X	X	X	X
P509	死因	2.02e-2	3,071	62	3,210					X	X	X	X
P172	族群	2.48e-2	2,013	50	2,293					X	X	X	X
P2348	时间段	2.63e-2	152	4	152					X	X	X	X
P102	政党成员	2.76e-2	2,175	60	2,668					X	X	X	X
P106	职业	2.85e-2	13,145	375	71,596	X	X	X	X	X	X	X	X
P495	原产国	3.85e-2	1,299	50	2,049			X	X	X	X	X	X
P136	类型	4.32e-2	5,303	229	11,761					X	X	X	X
P641	体育运动	4.43e-2	384	17	418					X	X	X	X
P19	出生地	6.11e-2	7,185	439	7,214		X	X	X	X	X	X	X
P69	就读于	6.47e-2	6,502	421	9,752					X	X	X	X
P463	成员	6.77e-2	3,705	251	11,490					X	X	X	X
P264	唱片公司	6.84e-2	2,002	137	3,456					X	X	X	X
P20	逝世地点	6.96e-2	5,417	377	5,442					X	X	X	X
P1050	医疗状况	7.59e-2	395	30	408					X	X	X	X
P101	工作领域	8.13e-2	1,967	160	2,421					X	X	X	X
P2283	uses	8.33e-2	12	1	12					X	X	X	X
P135	运动	9.20e-2	413	38	446					X	X	X	X
P119	埋葬地点	9.67e-2	1,944	188	1,972			X	X	X	X	X	X
P108	雇主	1.27e-1	3,016	382	4,795				X	X	X	X	X
P37	官方语言	1.83e-1	306	56	403					X	X	X	X
P840	叙事地点	1.97e-1	986	194	1,506					X	X	X	X
P17	国家	2.23e-1	641	143	1,323					X	X	X	X
P50	作家	2.35e-1	4	17	17								
P452	行业	2.50e-1	16	4	17				X		X		X
P551	居住地	2.52e-1	1,426	359	1,934						X		X
P749	上级组织	3.15e-1	73	23	82						X		X
P407	工作语言或名称	3.24e-1	34	11	43						X		X
P361	组成部分	3.48e-1	138	48	171						X		X
P57	导演	3.71e-1	542	201	561						X		X
P159	总部所在地	4.20e-1	157	66	169				X		X		X
P161	演员	4.80e-1	1,227	2,557	9,249						X		X
P1056	产品（.....）生产	5.00e-1	2	1	2								
P740	形成地点	5.19e-1	133	69	133						X		X
P131	位于（.....）	7.61e-1	163	124	212						X		X
P737	受.....影响	8.71e-1	514	590	1,508						X		X
P138	以.....命名	8.91e-1	49	55	57						X		X
P112	创立者	9.48e-1	55	58	67						X		X
P40	子女	9.54e-1	309	324	391				X		X		X
P451	未婚伴侣	9.62e-1	328	341	393						X		X
P530	外交关系	9.68e-1	214	221	6,225						X		X
兄弟姐妹关系	兄弟姐妹	9.95e-1	394	396	785						X		X
P26	配偶	1.00e+0	804	804	866						X		X
P3095	从事于	1.00e+0	2	2	2								
P54	运动队成员	1.00e+0	2	2	2								
P113	航空枢纽	1.00e+0	1	1	1								
P780	症状	1.00e+0	1	1	1								

9.2 超参数

所有模型均使用 PyTorch 1.11 版本，在 Nvidia GTX 2080TI 或 RTX A6000 显卡上进行训练。知识图谱补全模型使用 PyKEEN 1.8 版本的实现，文本编码器使用 Huggingface 的 Transformer 4.19 实现。每个训练周期的文本样本数量因不同模型配置而异，由“最大上下文数”（训练期间与某个实体相关联的文本上下文总数）和“每个样本的最大上下文数”（模型在每个训练步骤中使用的文本上下文数量）共同决定。如果模型在验证集划分上的目标指标（即 HITS@10）连续若干个训练周期的提升不超过 0.001，则停止模型训练。下面列出了第 6 节所介绍模型的超参数。

	排序				链接			
	Tiny	**小型**	**中**	**大**	**Tiny**	**小型**	**中**	**大**
嵌入维度	800	500	800	200	200	800	800	200
未冻结层	5	0	11	0	5	11	11	0
正则化器权重	7.16e-1	7.73e-1	7.12e-2	5.86e-3	6.56e-1	9.02e-3	7.12e-2	5.86e-3
每个样本的上下文数	1	1	1	1	1	1	1	1
最大上下文数	10	100	10	10	10	10	10	10
掩码	假	假	假	假	true	true	假	否
批次大小	4	8	40	20	4	8	40	20
学习率	3.94e-6	4.93e-6	4.39e-6	8.99e-6	8.47e-6	6.67e-6	4.39e-6	8.99e-6
权重衰减	3.48e-2	1.19e-2	6.64e-4	1.49e-4	1.22e-4	2.16e-4	6.64e-4	1.49e-4
Seed	5629275	4828059	2773483	4076657	8697782	9387603	2773483	4076657

表3：JOINT（单一上下文）——随机搜索后选出的最佳模型。

	排序				链接			
	Tiny	**小型**	**中**	**大**	**Tiny**	**小型**	**中**	**大**
嵌入维度	800	800	800	800	800	800	800	800
未冻结层	1	1	1	1	1	1	1	1
正则化器权重	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
每个样本的上下文数	40	40	40	10	40	40	40	10
最大上下文数	100	100	1000	1000	100	100	1000	1000
掩码	false	false	false	false	false	false	false	false
批次大小	1	1	1	4	1	1	1	4
子批次大小	5	5	10	10	5	5	10	10
学习率	5.00e-6	5.00e-6	5.00e-6	5.00e-6	5.00e-6	5.00e-6	5.00e-6	5.00e-6
权重衰减	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001	0.0001
Seed	4733486	5969007	7460979	6929361	4733486	5969007	7460979	6929361

表4：JOINT（多上下文）- 网格搜索后选出的最佳模型。

	排序				链接			
	Tiny	小型	中	大	Tiny	小型	中	大
每个样本的上下文数	1	1	1	1	1	1	1	1
最大上下文数	1	1	1	1	1	1	1	1
掩码	false	false	false	false	false	false	false	false
批次大小	10	10	10	10	10	10	10	10
子批次大小	10	10	10	10	10	10	10	10
学习率	5.00e-5	5.00e-5	5.00e-5	5.00e-5	5.00e-5	5.00e-5	5.00e-5	5.00e-5
Seed	1705119	2298300	3899079	8847385	1705119	2298300	3899079	8847385

表 5：OWE（单一上下文）——网格搜索后选出的最佳模型。

	排序				链接			
	Tiny	小型	中	大	Tiny	小型	中	大
每个样本的上下文数	10	10	10	10	10	10	10	10
最大上下文数	100	100	100	100	100	100	100	100
掩码	false	false	false	false	true	false	true	false
批次大小	1	1	1	1	1	1	1	1
子批次大小	10	10	10	10	10	10	10	10
学习率	1.00e-6	1.00e-6	1.00e-6	1.00e-6	1.00e-6	1.00e-6	1.00e-6	1.00e-6
Seed	3443300	3250085	4051229	6248903	9294686	9773709	9790717	6248903

表6： OWE（多上下文）- 网格 搜索。

	ComplEx			
	Tiny	小型	中	大
嵌入维度	300	300	300	500
学习率	1	0.6	0.4	0.6
正则化器权重	0.3	0.1	0.1	0.1
批次大小	64	64	64	64
Seed	1174584270	2593575093	2203942071	2649116927

表 7：OWE 参考嵌入——网格搜索后选出的最佳模型。