arXiv:2302.06761v3 [cs.CL] 8 May 2023

用于本体包含推理的语言模型分析

Yuan He[1], **Jiaoyan Chen**[2],**ErnestoJiménez-Ruiz**[3,4], **HangDong**[1],**Ian Horrocks**[1] 牛津大学, [2] 曼彻斯特大学, [3] 伦敦城市大学, [4] 奥斯陆大学
{yuan.he,hang.dong,ian.horrocks}@cs.ox.ac.uk
jiaoyan.chen@manchester.ac.uk
ernesto.jimenez-ruiz@city.ac.uk

摘要

近期，关于预训练语言模型（语言模型）能否充当知识库（KB）的研究引起了广泛关注。然而，现有研究聚焦于简单的、基于三元组的关系型知识库，却忽略了更复杂的、基于逻辑的概念化知识库，例如 OWL 本体。为探究语言模型对本体的知识，我们提出 OntoLAMA，这是一组基于推理的探测任务及数据集，涉及本体包含公理，涵盖原子概念和复杂概念[1]。我们在不同领域和规模的本体上进行了广泛实验，结果表明，与传统的自然语言推理（NLI）相比，语言模型编码的包含推理（SI）背景知识相对较少；但只需提供少量样本，就能显著提升其包含推理能力。我们将开源代码和数据集。[2]

1 引言

大型预训练语言模型（语言模型）的进展激发了研究者的兴趣，使其开始探究语言模型能够从知识库（知识库）中学习或推断出多少显式语义（AlKhamissi 等人，2022）。LAMA（语言模型分析）探针（Petroni 等人，2019）是最早采用基于提示的方法来模拟从各种知识库中查询关系知识的工作之一，例如 ConceptNet（Speer 和 Havasi，2012）和 GoogleRE[3]。随后的一些研究侧重于从常识知识库（Da 等人，2021）、生物医学知识库（Sung 等人，2021）、时序知识库（Dhingra 等人，2022）以及跨语言知识库（Liu 等人，2021a）中探测特定类型的知识。

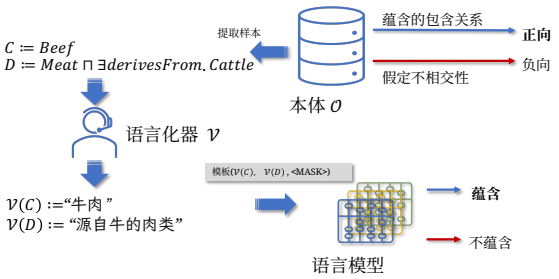

图1：ONTOLAMA 框架。

然而，现有的"语言模型即知识库"研究聚焦于简单的、基于三元组的关系型知识库，却忽视了更加形式化、基于逻辑且概念化的知识库。例如，像 "伦敦是英国的首都"这样的陈述可以表示为三元组 (伦敦, 首都, 英国)；但像 "关节炎是一种具有炎症形态的关节病"这样的句子描述的是概念 "关节炎"，却难以仅用三元组表示。此类概念知识需要一种形式化且具有表达能力的表示方式，才能得到精确定义。概念知识的一种著名模型是 OWL[4] 本体（Bech- hofer 等人，2004；Grau 等人，2008），它可以被视为一种描述逻辑（DL）知识库，具有丰富的内置知识表示词汇，并支持多种推理工具。以 "关节炎"为例，在描述逻辑中，该概念可以描述为 $Arthritis \sqsubseteq Arthropathy \sqcap \exists hasMorphology.Inflammatory$。

在这项工作中，我们沿着"语言模型即知识库"研究路线进一步推进，迈向形式化程度更高的语义，目标是描述逻辑知识库，尤其是 OWL 本体。当前有关语言模型与本体的研究大多由特定应用驱动。Liu 等人 (2020）、He 等人(2022）、

[1]本体 概念 也称为类。为避免与机器学习分类中的类 混淆，我们始终使用术语概念。[2]代码和说明：https://krr-oxford.github.io/DeepOnto/ontolama；HuggingFace 上的数据集：https://huggingface.co/datasets/krr-oxford/OntoLAMA/或 Zenodo 上的数据集：https://doi.org/10.5281/zenodo.6480540[3]https://code.google.com/archive/p/relation-extraction-corpus/

[4]为简便起见，我们将第二版 OWL 2 简称为 OWL：https://www.w3.org/TR/owl2-overview/

以及 Chen 等人 (2022) 通过对语言模型进行微调来处理本体整理任务，例如概念插入和匹配；而Ye 等人(2022）则将本体转换为图，以用于少样本学习中的数据增强。与这些由应用驱动的方法不同，我们探讨一个更为基础的问题：**语言模型在多大程度上能够推断由本体建模的概念知识？**具体而言，我们关注本体概念之间的包含关系。如图1所示，我们首先从本体中提取被视为正向（C， D）和负向（C 且 D 存在包含关系）样本的概念对，以及被视为负向（C 且D 假定彼此不相交）的概念对。请注意，采样过程完全自动化，并仔细考虑了 OWL 本体的语法和语义。为了将概念，尤其是包含复杂逻辑表达式的概念，转换为自然语言文本，我们开发了一种递归概念语言化器。我们以类似于自然语言推理（NLI）任务的方式来定义包含推理（SI）任务，并将概念对视为前提-假设对（Padó 和 Dagan，2022），随后将其套入模板，以生成语言模型的输入。

我们基于不同领域、不同规模的本体构建了 SI 数据集，并进行了大量实验。结果表明，在零样本设置下，语言模型在典型的 NLI 任务上的表现优于所构建的 SI 任务，这说明语言模型编码的本体包含关系背景知识相对较少。然而，只需提供少量样本（K-样本设置），SI 任务上的表现便会显著提升。这一观察结果与本研究考察的三个语言模型一致。

2 背景

2.1 OWL本体

OWL本体是一种描述逻辑（DL）知识库，由 TBox（术语）、ABox（断言）和 RBox（关系）公理（Krötzsch 等人，2012）组成。在这项工作中，我们重点关注 TBox 公理，它们规定了论域中概念之间的包含关系。包含公理的形式为$C \sqsubseteq D$ 其中 C 和 D 是概念表达式，涉及原子概念、否定（$\neg$）、合取（$\sqcap$）、析取（$\sqcup$）、存在限制（$\exists r.C$）、全称限制（$\forall r.C$）等（完整定义见附录 A）。原子概念**概念是一个命名概念、顶层概念（以每个个体为实例的概念）或底层概念（空概念）；而复杂概念** 由至少一种可用的逻辑运算符构成。等价公理 $C \equiv D$ 等价于 $C \sqsubseteq D$ 和 $D \sqsubseteq C$。

关于语义，在描述逻辑中，我们定义一个解释 $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$，它由一个非空集合 $\Delta^{\mathcal{I}}$ 和一个函数 $\cdot^{\mathcal{I}}$ 组成；该函数将每个概念 C 映射到 $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$，并将每个属性 r 映射到$r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$。如果 $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ 成立，我们称 $\mathcal{I}$ 是 $C \sqsubseteq D$ 的一个模型；如果 $\mathcal{I}$ 是本体$\mathcal{O}$ 中所有公理的模型，则称 $\mathcal{I}$ 是该本体的一个模型。如果对于 $\mathcal{O}$的每个模型 $\mathcal{I}$，$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$都成立，那么我们可以说$\mathcal{O} \models C \sqsubseteq D$。这定义了相对于本体的逻辑蕴涵，其定义比基于人类信念的文本蕴涵更严格。

个体 a 在$\mathcal{O}$ 中是概念 C 的实例，当且仅当 $\mathcal{O} \models C(a)$（$a^{\mathcal{I}} \in C^{\mathcal{I}}$ 对于 $\mathcal{O}$的每个模型 $\mathcal{I}$）。C 和 D 在 $\mathcal{O}$ 中是不相交的，当且仅当 $O \models C \sqcap D \sqsubseteq \bot$（或等价地 $O \models C \sqsubseteq \neg D$），这意味着 C 和 D没有共同实例 a。

开放世界假设（OWA）是 OWL 本体的基础。根据该假设，我们不能断定本体未蕴含的内容必然为假。例如，如果我们有一个仅包含一条公理的本体 $Paella \sqsubseteq \exists hasIngredient.Chicken$，那么在开放世界假设下，我们无法确定西班牙海鲜饭是否可以加入西班牙辣香肠作为配料。为了便于复用和扩展，本体通常会（有意地）保持欠规格化（Cimiano和Reyle，2003）；这一特性影响了我们在第3.1节中对负样本的定义。

2.2 相关工作

近年来，提示学习范式的兴起为如何在无需监督或仅需少量监督的情况下更好地使用预训练语言模型提供了启示（Liu 等人，2022）。然而，语言模型通常以随机方式进行预训练，因此难以研究语言模型隐式编码了哪些知识（Petroni 等人，2019），以及如何以最优或可控的方式访问语言模型（Gao 等人，2021；Li 等人，2022）。

我们的工作受到“语言模型即知识库”相关研究的启发（AlKhamissi 等人，2022），该领域设计了不同的探测方法，用于检验语言模型对关系数据的掌握情况。在Petroni 等人（2019）中，世界知识探测任务被设定为完形填空式问答任务，要求语言模型填入给定的 <MASK>标记，其中的

文本被包裹在人工设计的模板中。Sung 等人（2021）开展了类似研究，但将重点转向特定领域语言模型的（生物医学）领域知识。Liu 等人（2021a）使用多语言知识图谱（KG）对语言模型进行预训练，并在跨语言任务上进行测试。Dhingra 等人（2022）提出了带有时间信号的数据集，并使用文本到文本转换器 T5（Raffel 等人，2022）生成的模板对语言模型进行探测。

然而，现有的“语言模型即知识库”研究大多聚焦于关系事实，却忽略了逻辑语义和概念知识。相比之下，我们的工作聚焦于 OWL 本体，它们通过底层逻辑形式体系来表示概念知识。尽管近期有一些研究同时涉及语言模型和本体，但它们并未在语义层面对二者进行比较，而是更强调下游应用。例如，He 等人(2022) 将语言模型用作同义词分类器，以预测本体之间的映射；而Ye 等人(2022) 则使用本体为语言模型提供额外上下文，以帮助其进行预测。

3 包含推理

3.1 任务定义

回顾第 2.1 节中的定义，包含公理 $C \sqsubseteq D$ 可以解释为：“ C 的每个实例都是 D 的实例 ”。据此，我们可以构造一个前提 — 假设对，其中前提为“x 是一个 C”，假设为“x 是一个 D”，其中 x 是某个个体。请注意，前提和假设有多种表达方式，我们采用一种简单而有效的方式（见第 5.1 节）。接下来，需要一个本体语言化器，将概念表达式 C 和 D 转换为自然语言文本。与自然语言推理（NLI）或文本蕴涵识别（RTE）（Poliak, 2020; Padó 和 Dagan, 2022）类似，包含推理（SI）任务由此定义为判断前提是否蕴含假设。请注意，SI 类似于双向 RTE 任务 [5]，但我们不考虑中立类 [6]。

给定本体 $\mathcal{O}$，我们提取正向和负向包含关系，以探测语言模型。正向样本是概念对 $(C,\ D)$，其中 $\mathcal{O} \models C \sqsubseteq D$。由于开放世界假设，我们无法确定 $(C,\ D)$，其中 $\mathcal{O} \not\models C \sqsubseteq D$ 是否确实构成负向包-含关系（更多解释见附录F）；为生成合理的负样本，我们提出假定不相交性[7]，定义如下：

定义（假定不相交性）。如果两个概念 C和 D在 $\mathcal{O} \cup \{C \sqcap D \sqsubseteq \bot\}$中可满足，并且 $\mathcal{O}$ 中不存在命名原子概念 A 满足 $\mathcal{O} \models A \sqsubseteq C$ 和 $\mathcal{O} \models A \sqsubseteq D$，那么 C 和D 就被假定为不相交。

第一个条件确保将它们的不相交公理添加到 $\mathcal{O}$ 后，C 和 D 仍然**可满足**；而第二个条件确保 C 和 D **没有共同后代**，否则不相交公理会使任何共同后代都不可满足。如果两个概念 C 和 D 满足这两个条件，我们就将（$C,\ D$）视为有效的负向包含关系。

然而，在实际应用中，验证每一对概念的可满足性 $(C,\ D)$ 效率不高，尤其是在本体规模庞大且复杂时。因此，我们在附录 E 中提出了一种可行的替代方案，以取代可满足性检查。

为进行推理以提取蕴含的正向包含关系，并验证采样得到的负向包含关系，我们需要采用经过验证、可靠且完备的 OWL 推理器，例如 HermiT（Glimm 等人，2014）。

在以下小节中，我们提出两种具体的 SI 任务及其各自的包含关系采样方法。

3.2 原子包含关系推断

第一项任务针对仅涉及命名原子概念的包含公理。这类公理通常是本体中最常见的，并且可以通过使用概念名称轻松地转述。在本研究中，我们使用内置注释属性 rdfs:label 定义的标签（英文）作为概念名称。

正样本取自目标本体中所有被蕴含的包含公理。我们考虑两类负样本：（*i*）**软负样本**由两个随机概念组成，以及（*ii*）**硬负样本**由两个随机兄弟概念组成。两个兄弟概念会构成“硬”负样本，因为它们共享同一个父概念（因此具有更接近的语义），但

[5] RTE 指南：https://tac.nist.gov/2008/rte/rte.08.guidelines.html。[6] 中性 本质上意味着两个术语彼此无关。本体通常都存在欠规格化的问题，因此，即使两个概念之间尚未蕴涵出包含关系或非包含关系，它们在现实世界中仍可能存在隐含关联。

[7] Schlobach (2005) 和 Solimando 等人 (2017) 在不同语境下提出了类似的假设。

模式	文本化（$\mathcal{V}$）
A （原子）	A 的名称（rdfs:label）
r （属性）	r 的名称（rdfs:label），受限于附录中的规则 C
$\neg C$	“非 $\mathcal{V}(C)$”
$\exists r.C$	“某种满足 $\mathcal{V}(r)$ 某个 $\mathcal{V}(C)$的事物”
$\forall r.C$	“某种满足 $\mathcal{V}(r)$ 仅 $\mathcal{V}(C)$的事物”
$C_1 \sqcap ... \sqcap C_n$	如果 $C_i = \exists/\forall r.D_i$ 且 $C_j = \exists/\forall r.D_j$，它们将被重写为 $\exists/\forall r.(D_i \sqcap D_j)$ 文本化之前；假设重新书写后，新表达式为 $C_1 \sqcap ... \sqcap C_{n'}$ **(a)** 如果**所有** C_is（对于 $i = 1,..., n'$）都是限制，形式为 $\exists/\forall r_i.D_i$: “某物，$\mathcal{V}(r_1)$ 某些/仅有 $V(D_1)$ 以及.....以及 $\mathcal{V}(r_{n'})$ 某些/仅有 $V(D_{n'})$” **(b)** 如果**某些** C_is（对于 $i = m+1, ..., n'$）是以下形式的限制 $\exists/\forall r_i.D_i$: “$\mathcal{V}(C_1)$ 且.....且 $\mathcal{V}(C_m)$ 使得 $\mathcal{V}(r_{m+1})$ 某些/仅有 $V(D_{m+1})$ 且.....且 $\mathcal{V}(r_{n'})$ 某些/仅有 $V(D_{n'})$” **(c)**如果**没有** C_i 是一个限制: “$\mathcal{V}(C_1)$与.....与 $\mathcal{V}(C_{n'})$”
$C_1 \sqcup ... \sqcup C_n$	与将 $C_1 \sqcap ... \sqcap C_n$ 语言化相似，区别在于“与”替换为“或”，且情形 **(b)** 采用与情形**(c)**相同的文本化方式

表 1：在 OWL 本体中将复杂概念表达式 C 语言化的递归规则。请注意，合取/析取模式中的 C_i 也是任意的复杂概念。

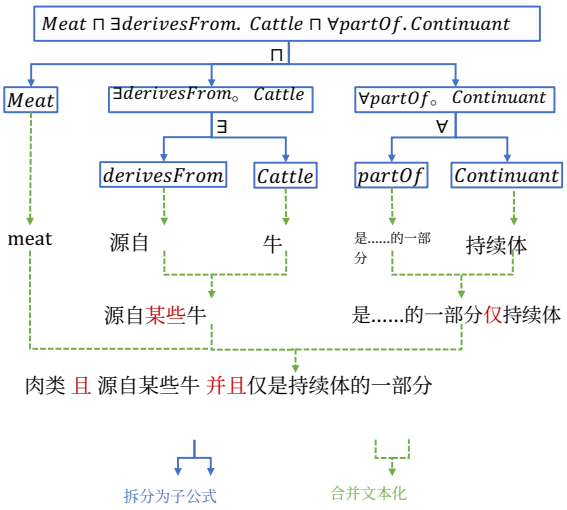

图2：示例说明如何将递归概念语言化器应用于一个复杂概念表达式。该算法首先 将复杂概念拆分为子公式树，对叶节点进行语言化，然后递归地合并语言化后的子公式。每一步合并中，与逻辑运算符相关的关键词都以红色标出。更多示例请参阅附录D。

通常是不相交的。采样得到的概念对必须满足第3.1节中定义的假定不相交性，才能被接受为有效负样本。我们首先采样数量相等的软负样本和硬负样本，然后随机截取结果集，使其规模与正样本集一致，以保持类别平衡。

3.3 复杂包含推理

在第二个包含推理任务中，我们考虑涉及复杂概念的包含公理。具体而言，我们选择形式为 $A \equiv C$[8] 的等价公理作为锚点（其中 A 和 C 分别是原子概念和复杂概念），并将其等价地转换为形式为 $A \sqsubseteq C$ 和 $C \sqsubseteq A$ 的包含公理，由此复杂概念可以同时出现在前提侧和假设侧。

递归概念语言化器 为了将复杂的 C 转换为自然语言文本，我们开发了递归概念语言化器，它由语法树解析器和一组递归规则组成（见表1）。具体示例如

图2展示了一个示例：复杂概念 $Meat \sqcap \exists derivesFrom.Cattle \sqcap \forall partOf.Continuant$ 首先由语法解析器拆分为子公式树，然后根据表1中的递归规则进行语言化。叶节点要么是原子概念，要么是属性，并以其名称进行语言化。在每个递归步骤中，语言化后的子节点会根据其父节点中的逻辑模式合并。请注意，我们对合取（⊓）和析取（⊔）模式进行了适度的冗余消除。以图2为例，语言化后的原子概念“肉类”置于“that”之前，作为先行项；两个限制的语言化合取则置于“that”之后，作为关系从句。如果不涉及概念$Meat$，“肉类”可以替换为“某物”。此外，如果两个具有相同量词和属性的限制通过 ⊓ 或 ⊔连接，它们将合并为一个限制。例如，$\exists derivesFrom.Cattle \sqcap \exists derivesFrom.Sheep$ 将转换为$\exists derivesFrom.(Cattle \sqcap Sheep)$。

我们从目标本体中提取形式为$A \equiv C$ 的等价公理。以每条此类公理为锚点，可以得到形式为 $A_{sub} \sqsubseteq C$或 $C \sqsubseteq A_{super}$ 的正向复杂包含公理，其中 A_{sub} 和 A_{super} 分别是 A的子类和超类。为推导出具有挑战性的负样本，我们首先随机替换

[8]这种形式的等价公理称为命名概念的定义，在 OWL 中很常见。

来源	#概念	#等价公理	#数据集（训练/开发/测试）
Schema.org	894	-	Atomic SI: 808/404/2, 830
DOID	11, 157	-	Atomic SI: 90, 500/11, 312/11, 314
FoodOn	30,995	2,383	Atomic SI: 768, 486/96, 060/96, 062 Complex SI: 3, 754/1, 850/13, 080
GO	43,303	11,456	Atomic SI: 772, 870/96, 608/96, 610 Complex SI: 72,318/9,040/9,040
MNLI	-	-	biMNLI: 235,622/26,180/12,906

表 2：本体、SI 数据集和 biMNLI 数据集的统计信息。

将 $A \equiv C$中的命名概念或属性替换为另一个概念或属性，以生成以下情况之一：(i) $A' \equiv C$ （如果将 A 替换为A'），或(ii) $A \equiv C'$ （如果 C 被扰动）。不失一般性，我们假定随机替换导致情况(ii)。然后，我们检查 A 和C' 是否满足第 3.1 节所述的假定不相交性。若满足，则最终的负向包含关系可以是 $A \sqsubseteq C'$ 或 $C' \sqsubseteq A$；否则，我们跳过该样本。例如，给定 $SunflowerSeed \equiv Seed \sqcap \exists DerivesFrom.HelianthusAnnuus$，如果将 C 中的 $Seed$ 替换为 $Fruit$，从而生成 C'，则一种可能的负向包含关系是 $SunflowerSeed \sqsubseteq Fruit \sqcap \exists DerivesFrom.HelianthusAnnuus$。

4 数据集

在这项工作中，我们考虑了不同领域和规模的本体，包括：

- Schema.org[9] （发布于 2022-03-17）：一种通用本体，为 Web 上的结构化数据维护基本架构；
- DOID[10] （发布于 2022-09-29）：一个关于人类疾病的本体（Schriml 等，2012）；
- FoodOn[11] （发布于 2022-08-12）：一个专注于食品相关知识的本体，包括食品产品、食物来源、食品营养等（Dooley 等，2018）。
- GO[12] （发布于 2022-11-03）：一个粒度极细、应用广泛的生物医学本体，专注于基因和基因功能（Ash- burner 等，2000）。

我们使用了实验时最新的版本。附录中展示了本体预处理的详细过程B。

[9]https://schema.org/ [10]https://disease-ontology.org/ [11]https://foodon.org/ [12]http://geneontology.org/

我们为每个本体构建一个原子 SI 数据集，但由于 FoodOn 和基因本体包含大量等价公理，仅为它们创建复杂 SI 数据集。为避免特定等价公理产生过多重复的概念表达式，在 Complex SI 设置中，我们针对每条等价公理最多采样 4 个正样本和 4 个负样本。为实现类别平衡，我们特意在每个数据划分中保持负样本数量与正样本数量相同。对于大多数生成的数据集，我们按 8 : 1 : 1 的比例将其划分为训练集、开发集和测试集；对于规模相对较小的 Schema.org 原子 SI 数据集和 FoodOn 复杂 SI 数据集，则采用 2 : 1 : 7 的划分比例。请注意，我们在探测研究中主要关注 K-shot 设置，因此所需的训练集和开发集样本较少。

为了比较语言模型在传统 NLI 任务上的表现，我们还创建了 biMNLI，即多体裁自然语言推理（MNLI）语料库的一个子集（Williams 等人，2018），其中 *(i)* 移除了中立类别及其样本，*(ii)* 将匹配测试集和不匹配测试集合并为一个测试集，*(iii)* 将 10% 的训练数据用作开发集，并且*(iv)* 将蕴含与矛盾的比例设为 1 : 1 （通过舍弃多数类别中的额外样本），以保持先验平衡。本体中的命名概念和等价公理数量，以及 SI 数据集和 biMNLI 数据集（各划分中）的样本数量，均报告于表2。

5 实验

5.1 基于提示的推理

为在基于提示的设置下执行推理任务，我们将口语化表达的包含公理和 <MASK> 标记包装成一个模板，以

作为语言模型的输入。我们选择使用不同组合的人工设计模板[13] (T_1 和T_2)以及标签词(L_1 至 L_3)，这些组合在自然语言推理任务中已取得了令人期待的结果(Schick 和 Schütze，2021；Gao 等人，2021)，具体如下：

$$T_1 := \underbrace{\text{它是 <A> } \mathcal{V}(C)}_{\text{前提}}\text{? <MASK>, } \underbrace{\text{它是 <A> } \mathcal{V}(D)}_{\text{假设}}\text{.}$$

$$T_2 := \text{“}\underbrace{\text{它是 <A> } \mathcal{V}(C)}_{\text{前提}}\text{”? <MASK>, “}\underbrace{\text{它是 <A> } \mathcal{V}(D)}_{\text{假设}}\text{”.}$$

L_1 :={“positive”: [“是”], “negative”: [“否”]}L_2 :={“positive”: [“正确”], “negative”: [“错误”]}
L_3 :={“positive”: [“是”， “正确”],“negative”: [“否”， “错误”]}

其中 <A> 是 *“a”*、*“an”*，或直接使用 空白 ，具体取决于下一个词[14]， $\mathcal{V}(\cdot)$ 是第3节中定义的概念 verbalisation 函数，而 <MASK> 是语言模型需要预测的词元。对于输入样本 y ，预测类 $x = (C, D)$ （“正向”或“负向”）的概率定义如下：

$$\begin{aligned} P(y \mid x) &= P(\texttt{<MASK>} \in L_j[y]) \mid T_i(C, D)) \\ &= \frac{\sum_{v \in L_j[y]} \exp(\mathbf{w}_v \cdot \mathbf{h}_{\texttt{<MASK>}})}{\sum_{w \in L_j[\cdot]} \exp(\mathbf{w}_w \cdot \mathbf{h}_{\texttt{<MASK>}})} \end{aligned}$$

其中， $L_j[\cdot]$ 和 $L_j[y]$ 分别表示 L_j 中定义的所有标签词以及L_j中定义的类 y 的标签词； $T_i(C, D)$ 表示通过模板 T_i对概念 C 和 D 进行文本化后得到的变换后的文本； $\mathbf{w}_v$ 和$\mathbf{w}_w$ 分别是标签词 v 和 w对应的向量；而$\mathbf{h}_{\texttt{<MASK>}}$ 表示掩码词元的隐藏向量。通过最小化交叉熵损失来训练模型进行预测。

对于 biMNLI 数据集，前提和假设替换为数据集中原本提供的内容，但我们删除了末尾的标点符号。

在有关语言模型的主要实验中，对于 K-shot设置，我们考虑 T_i 和L_j 的所有组合，并另外考虑3个随机种子（因此每种设置进行18次实验），其中$K > 0$。 K 的值指的是我们分别从训练集和开发集中随机抽取的每个分类标签（正向或负向）的样本数。对于 $K = 0$ （零样本），不同的随机种子不会影响结果。对于全监督设置，我们只考虑一个随机种子和一种组合（ T_1 和L_1），因为我们的初步实验表明，在大样本上进行微调时，不同随机种子以及不同模板和标签词组合所带来的方差较小。

我们的代码实现主要依赖The OWL API[15] 进行本体处理和推理，并依赖OpenPrompt[16] 进行提示学习（Ding 等人，2022）。每项 K-shot（其中$K > 0$）实验训练 10 个训练轮次；而对于涉及大量训练样本的全监督设置，我们只训练 1 个训练轮次。[17] 在开发集上表现最佳的模型（每个训练轮次各选一个）将用于测试集推理。我们使用 AdamW 优化器（Loshchilov 和 Hutter，2019），初始学习率、权重衰减和预热步数分别设为 10^{-5}、 10^{-2} 和 50。所有实验均在两块 Quadro RTX 8000 GPU 上进行。

5.2 结果与分析

语言模型与设置 我们选择 RoBERTa 系列中的语言模型（Liu 等人，2019），因为它们常用于完形填空式探测任务（Liu 等人，2021b；Sung 等人，2021；Kavumba 等人，2022）。表 3展示了 roberta-large 和roberta-base的关键实验结果；此外，我们还在后文对 roberta-large的一个生物医学变体进行了消融研究。

对于表 3中的两个语言模型，我们报告了 K-样本设置下的结果， $K \in \{0, 4, 32, 128\}$。此外，我们还报告了 roberta-large 在全监督设置下作为预言机的结果。对于每种设置，我们报告平均准确率和标准差（如适用）。为清楚观察性能如何随 K 增加而变化，我们给出了图 3 ，展示了 K-样本设置下 roberta-large 的结果，并补充了 K($\{8, 16, 64\}$) 的数值。两个语言模型的完整结果表，以及展示 roberta-base性能的图表，见附录 G。

基线 如前所述，我们特意保持每个数据划分中的类别平衡，因此多数投票的准确率分数均为50.0%。此外，

[13]我们稍作修改，在前提句和假设句前添加前缀“It/it is <A>”，使句子完整。[14]如果下一个词以元音开头，则使用“an”；如果下一个词是“*something*”，则留空。

[15]https://owlapi.sourceforge.net/[16] https://thunlp.github.io/OpenPrompt/[17]由于 Schema.org的 Atomic SI 和 FoodOn的复杂 SI 数据集训练集规模较小，因此其全监督设置仍需 10 个训练轮次。

		Atomic SI				Complex SI	
设置	biMNLI	Schema.org	DOID	FoodOn	GO	FoodOn	GO
多数投票	50.0	50.0	50.0	50.0	50.0	50.0	50.0
word2vec							
K=4	51.5 (0.2)	54.9 (2.9)	64.6 (2.6)	63.5 (1.0)	60.1 (4.1)	56.8 (4.2)	56.9 (5.4)
K=128	52.1 (0.4)	73.0 (0.4)	70.8 (1.7)	71.4 (1.0)	66.3 (0.9)	63.8 (0.6)	66.4 (1.3)
roberta-base							
K=0	62.5 (6.5)	56.4 (3.6)	53.3 (4.0)	54.6 (4.4)	49.0 (2.4)	55.9 (3.6)	48.7 (3.1)
K=4	67.6 (5.2)	62.9 (5.2)	61.8 (6.7)	62.1 (4.2)	65.2 (5.0)	62.4 (3.2)	52.2 (7.1)
K=32	78.8 (1.1)	84.3 (2.0)	89.0 (1.4)	85.0 (1.1)	84.6 (2.5)	77.0 (1.5)	76.4 (2.5)
K=128	85.1 (1.0)	91.1 (0.7)	92.4 (0.7)	90.0 (0.7)	89.0 (0.8)	85.5 (1.3)	86.9 (1.5)
roberta-large							
K=0	68.7 (6.2)	61.7 (7.2)	59.8 (5.4)	60.1 (8.8)	54.6 (1.9)	56.1 (1.9)	50.4 (0.6)
K=4	78.1 (6.6)	69.4 (5.4)	74.0 (5.5)	71.6 (4.4)	67.6 (3.4)	64.1 (5.1)	56.9 (5.7)
K=32	89.9 (1.2)	87.3 (1.9)	92.3 (0.7)	88.9 (1.6)	87.7 (1.6)	80.8 (3.8)	81.6 (2.2)
K=128	93.0 (0.8)	92.9 (0.8)	93.4 (0.5)	92.2 (0.5)	91.0 (0.7)	88.4 (1.1)	90.2 (1.0)
full	97.5	95.4	97.8	98.7	98.1	95.8	98.8

表3：biMNLI、Atomic SI 和 Complex SI任务的结果，每个单元格均注明“平均准确率（标准差）”，但多数投票和全监督设置除外，因为这两种设置没有标准差数据。

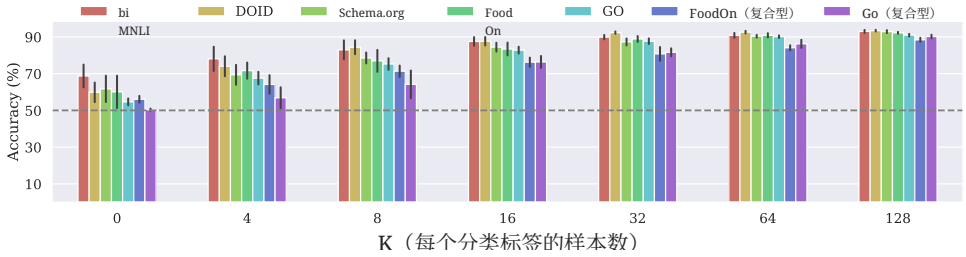

图3：biMNLI、Atomic SI 和 Complex SI 任务的 K-shot 结果（针对 roberta-large）的可视化，其中点状水平线表示多数投票。柱形的排列顺序与图例相同。

我们考虑使用 word2vec (Mikolov等人, 2013年）在GoogleNews上预训练的[18]，并搭配逻辑回归分类器作为基线模型，以展示经典的非上下文词嵌入模型在SI任务上的表现。对于这一基线，我们只报告 $K \in \{4, 128\}$ 的结果，因为增加 K 并不会带来显著变化，并且 $K = 128$ 的结果与 $K = 4$的结果大致相当，适用于 roberta-large。这表明，SI样本模式并不容易通过 word2vec捕获。

SI 与 biMNLI 的比较 从结果中，我们首先观察到，roberta-large 和 roberta-base 在 biMNLI 上的零样本结果均优于在所有 SI 数据集上的结果，分别至少高出 7.0% 和 6.1%。这表明，在我们的提示设置下，这两种语言模型在 biMNLI 上编码了更多背景知识而非 SI。不过，随着 K 增加，biMNLI 和 SI 上的表现均持续显著提升（而标准差通常会降低）；我们可以看到，在 $K = 32$时，roberta-base在原子 SI 任务上的平均准确率已超过 biMNLI。在 $K = 64$ 时（见图3），biMNLI 和所有原子 SI 任务的平均准确率均收敛至约90。0%；两种语言模型在两个 Complex SI任务上的得分也均超过80。0%。此外，roberta-large 在每种设置下的得分始终高于roberta-base 。

SI任务之间的比较 我们观察到，Complex SI 通常比原子SI更难。例如，在 $K = 0$，roberta-large 在 GO 的 Complex SI 数据集上达到 50.4%，几乎与多数投票相当；在 $K = 128$，roberta-large 在 FoodOn 的 Complex SI 数据集上达到 88.4%，而在其他数据集上的得分均超过 90%。我们还可以观察到

[18] https://code.google.com/archive/p/word2vec/

K	DOID	GO	Go（复合型）
0	49.7 (0.4)	50.1 (0.2)	50.0 (0.0)
4	64.8 (7.9)	66.2 (6.5)	50.0 (0.7)
32	94.7 (1.3)	93.5 (1.1)	73.5 (3.6)
128	96.3 (0.4)	95.2 (0.5)	90.5 (1.8)

表4：roberta-large-pm-m3-voc在生物医学本体DOID和基因本体的SI任务上的结果。

从图3 可以看出，Complex SI任务的得分通常低于原子SI任务。在原子SI任务中，我们发现基因本体 最具挑战性，这在意料之中，因为基因本体 是细粒度专家级本体。然而，令我们惊讶的是，在 $K = 32$ ，92.3%的得分DOID 高于所有其他任务；考虑到DOID是领域特定本体，这一点令人意外。

领域特定SI 我们进一步测试了领域特定语言模型在领域特定SI任务上的表现。具体而言，我们考察了变体 roberta-large-pm-m3-voc。该模型在生物医学语料库PubMed摘要、PMC全文和MIMIC-III临床笔记上进行了预训练，并采用了基于PubMed数据重新学习的子词词汇表（Lewis 等人，2020）。在表4中，我们展示了 K 样本结果，呈现了roberta-large-pm-m3-voc 在与生物医学本体DOID 和基因本体相关的三项SI任务上的表现。零样本得分与多数投票几乎相当，但在roberta-large 的表现基础上，随着DOID 和基因本体 的原子SI任务中， K逐渐提升，性能提升也更加显著。令人意外的是，基因本体的复杂SI设置 似乎对RoBERTa的这一生物医学变体颇具挑战。例如，在 $K = 4$时，得分相较于 $K = 0$并未提升。

模板和标签词 语言模型的访问方式是影响性能的重要因素，尤其是在训练样本很少或没有训练样本时。例如，roberta-large 在 $K = 0$ FoodOn 的原子 SI 任务上取得了 8.8% 的标准差，这表明不同的模板和标签词组合会带来显著的性能波动。尽管基因本体的 Complex SI 上标准差仅为 0.6%，但相应的准确率（50.4%）表明这些组合均不起作用。此外，某个语言模型中的有效模板或标签词无法迁移到另一个语言模型，这一点可从 roberta-large-pm-m3-voc 在生物医学本体的 $K = 0$ SI任务上表现不佳看出。这些观察结果表明，我们要么尚未找到通用的模板和标签词组合，要么语言模型需要针对不同类型的知识定制访问方式

组合，或者语言模型需要针对不同类型的知识定制访问方式。

6 结论与讨论

作为一项将本体引入“LMs-as-KBs”系列的工作，本文强调如何遵循其形式语义，建立从逻辑表达式到自然语言表达式的有意义转换。为此，我们借助自然语言推理（NLI）设置定义了包含推理（SI）任务，并谨慎考虑如何处理文本蕴含与逻辑蕴含之间的差异。我们还为 OWL 本体开发了递归概念 verbaliser，作为辅助工具。结果表明，在我们的 SI 设置下，只需提供少量标注样本，语言模型便能成功学习推断原子和复杂包含关系。这为利用语言模型研究更复杂的推理任务，或在训练集有限的情况下借助本体语义引导语言模型，铺平了道路。

事实上，当前的 SI设置并非探测本体包含知识的唯一方法；例如，我们可以直接将 $C \sqsubseteq D$语言化为“$\mathcal{V}(C)$ 是 $\mathcal{V}(D)$”，并构造类似事实核查的探测任务，或者等价地，构造一个前提为空的推理任务。然而，我们的初步实验表明，这种设置不如当前的 SI设置有效。

本文提出的工作为未来研究带来了机遇：*(i)* 所提出的本体语言化方法尚未涵盖复杂概念的所有可能模式（例如，带有基数限制和名词的模式）；*(ii)* 我们尚未充分考虑本体中可能包含的同义词、定义和注释等文本信息；*(iii)* 我们只考虑了 TBox（术语）公理，而 ABox（断言）公理也可用于成员关系预测任务等场景，其目标是对个体所属的概念进行分类。因此，开发一种能够语言化逻辑表达式的稳健工具，并扩展本体推理设置，是接下来可以开展的工作。另一个值得近期探索的方向是，在考虑本体逻辑语义的情况下，使用本体训练语言模型。由此得到的语言模型有望应用于不同的下游本体整理任务，例如本体匹配和实体链接，并且微调所需的样本更少。

局限性

由于我们主要关注所谓 TBox（术语）公理中所蕴含的概念知识，因此不考虑 ABox（断言）公理。ABox 公理可以描述特定个体的情况（例如，某人的健康状况），这可能引发隐私问题，而且我们也不希望语言模型获取此类知识。因此，处理 ABox 公理可能需要在数据预处理方面进行额外的工程工作。

伦理考量

在本研究中，我们利用公开可用的本体构建了用于所提出的包含推理（SI）任务的新数据集：`Schema.org`、`DOID`、`FoodOn`和基因本体，其下载链接见第4节。biMNLI 数据集由现有的开源 MNLI 数据集构建而成。我们已确认，所有这些数据集均不存在隐私或许可方面的问题。

致谢

本研究得到了 Samsung Research UK (SRUK) 以及 EPSRC 项目 OASIS (EP/S032347/1)、UKFIRES (EP/S019111/1) 和 ConCur (EP/V050869/1) 的资助。

参考文献

Badr AlKhamissi, Millicent Li, Asli Celikyilmaz, Mona T. Diab 和 Marjan Ghazvininejad。2022。《作为知识库的语言模型综述》。*ArXiv*，abs/2204.06031。

Michael Ashburner、Catherine A. Ball、Judith A. Blake、David Botstein、Heather L. Butler、J. Michael Cherry、Allan Peter Davis、Kara Dolinski、Selina S. Dwight、Janan T. Eppig、Midori A. Harris、David P. Hill、Laurie Issel-Tarver、Andrew Kasarskis、Suzanna E. Lewis、John C. Matese、Joel E. Richard- son、Martin Ringwald、Gerald M. Rubin 和 Gavin Sherlock。2000。《基因本体：生物学统一化工具》。*NatureGenetics*，25:25–29。

Sean Bechhofer、Frank Van Harmelen、Jim Hendler、Ian Horrocks、DeborahL McGuinness、PeterF Patel- Schneider、Lynn Andrea Stein 等。2004。《OWL 网络本体语言参考》。*W3C* 建议。

Jiaoyan Chen、Yuan He、Ernesto Jiménez-Ruiz、Hang Dong 和 Ian Horrocks。2022。本体蕴涵预测的上下文语义嵌入。*ArXiv*，abs/2202.09791。

Philipp Cimiano 和 Uwe Reyle。2003。基于本体的语义构建中的欠规格化与消歧。在句法-语义接口前景与最新进展洛林/萨尔兰研讨会论文集，*2003*年*10*月*20*日至*21*日，法国南锡，第33–38页。

Jeff Da、Ronan Le Bras、Ximing Lu、Yejin Choi 和 Antoine Bosselut。2021。《少样本知识模型中的常识涌现分析》。载于自动化知识库构建会议。

Bhuwan Dhingra、Jeremy R. Cole、Julian Martin Eisenschlos、Daniel Gillick、Jacob Eisenstein 和 William W. Cohen。2022。时间感知语言模型即时间知识库。计算语言学协会会刊，10：257–273。

Ning Ding、Shengding Hu、Weilin Zhao、Yulin Chen、Zhiyuan Liu、Haitao Zheng 和 Maosong Sun。2022。OpenPrompt：用于提示学习的开源框架。载于计算语言学协会第*60*届年会论文集：系统演示，第105–113页，都柏林，爱尔兰。计算语言学协会。

Damion M. Dooley、Emma J. Griffiths、Gurinder Pal Singh Gosal、Pier Luigi Buttigieg、R. Hoehn- dorf、Matthew Lange、Lynn M. Schriml、Fiona S. L. Brinkman 和 William W. L. Hsiao。2018。FoodOn：一个统一的食品本体，旨在提高全球食品可追溯性、质量控制和数据整合。《style id='1'>NPJ《食品科学》，2。

Tianyu Gao、Adam Fisch 和 Danqi Chen。2021。让预训练语言模型成为更好的少样本学习器。载于第*59*届计算语言学协会年会暨第*11*届国际自然语言处理联合会议论文集（第*1*卷：长论文），第3816–3830页，线上。计算语言学协会。

BirteGlimm、Ian Horrocks、Boris Motik、Giorgos Stoi- los 和 Zhe Wang。2014。HermiT：一个 OWL 2 推理器。《自动推理杂志》，53(3)：245–269。

Bernardo Cuenca Grau、Ian Horrocks、Boris Motik、Bi- jan Parsia、Peter Patel-Schneider 和 Ulrike Sattler。2008。OWL 2：OWL 的下一步。*Web* 语义，6(4):309–322。

Yuan He、Jiaoyan Chen、Denvar Antonyrajah 和 Ian Horrocks。2022。BERTMap：基于 BERT 的本体对齐系统。*AAAI*人工智能会议论文集，36(5):5684–5691。

PrideKavumba、Ryo Takahashi 和 YusukeOda。2022。基于提示的模型是否一无所知？载于计算语言学协会第*60* 届年会论文集

计算语言学（第 *1* 卷：长篇论文），第 2333–2352 页，爱尔兰都柏林。计算语言学协会。

Markus Krötzsch、František Simanˇcík 和 Ian Horrocks。2012。《描述逻辑入门》。*ArXiv*，abs/1201.4089。

Patrick Lewis、Myle Ott、Jingfei Du 和 Veselin Stoyanov。2020。面向生物医学和临床任务的预训练语言模型：理解并拓展最先进技术。收录于第三届临床自然语言处理研讨会论文集，第146–157 页，线上。计算语言学协会。

Xiang Lisa Li、John Thickstun、Ishaan Gulrajani、Percy Liang 和 Tatsunori Hashimoto。2022。Diffusion-LM 改进可控文本生成。*ArXiv*，abs/2205.14217。

Hao Liu、Yehoshua Perl 和 James Geller。2020。通过转换和总结生物医学本体结构，使用 BERT 训练模型进行概念放置。《生物医学信息学杂志》，112:103607。

Linlin Liu、Xin Li、Ruidan He、Lidong Bing、Shafiq Joty 和 Luo Si。2021a。《基于知识的多语言语言模型》。*ArXiv*预印本 *arXiv:2111.10962*。

Pengfei Liu、Weizhe Yuan、Jinlan Fu、Zhengbao Jiang、Hiroaki Hayashi 和 Graham Neubig。2022。《预训练、提示与预测：自然语言处理中的提示方法系统综述》。《*ACM*计算综述》（*CSUR*）。

Yinhan Liu、Myle Ott、Naman Goyal、Jingfei Du、Man- dar Joshi、Danqi Chen、Omer Levy、Mike Lewis、Luke Zettlemoyer 和 Veselin Stoyanov。2019。《RoBERTa：一种经过稳健优化的 BERT预训练方法》。*ArXiv*预印本 *arXiv:1907.11692*。

Zeyu Liu、Yizhong Wang、Jungo Kasai、Hannaneh Hajishirzi 和 Noah A. Smith。2021b。跨时间探测：RoBERTa 知道什么，以及何时知道？载于计算语言学协会论文集：*EMNLP 2021*，第 820–842 页，多米尼加共和国蓬塔卡纳。计算语言学协会。

Ilya Loshchilov 和 Frank Hutter。2019。解耦权重衰减正则化。发表于国际学习表征会议。

Tomas Mikolov、Kai Chen、Greg Corrado 和 Jeffrey Dean。2013年。向量空间中的词表示高效估计。*arXiv*预印本 *arXiv:1301.3781*。

Sebastian Padó 和 Ido Dagan。2022。679文本蕴含。发表于牛津计算语言学手册。牛津大学出版社。

Fabio Petroni、Tim Rocktäschel、Sebastian Riedel、Patrick Lewis、Anton Bakhtin、Yuxiang Wu 和 Alexander Miller。2019。语言模型能否充当知识库？载于《*2019*年自然语言处理经验方法会议暨第九届自然语言处理国际联合会议（*EMNLP-IJCNLP*）论文集》，第2463–2473页，香港，中国。计算语言学协会。Adam Poliak。2020。关于将文本蕴涵识别作为自然语言处理评估的一项调查。载于首届自然语言处理系统评估与比较研讨会论文集，第92–109页，线上。计算语言学协会。Colin Raffel、Noam S hazeer、Adam Roberts、Katherine Lee、Sharan Narang、Michael Matena、Yanqi Zhou、Wei Li 和 Peter J. Liu。2022。探索统一文本到文本转换器迁移学习的极限。《机器学习研究杂志》，21(1)。Timo Schick 和 Hinrich Schütze。2021。利用完形填空题进行少样本文本分类和自然语言推理。载于《欧洲计算语言学协会第*16*届会议论文集：主卷》，第255–269页，线上。计算语言学协会。Stefan Schlobach。2005。通过定位问题进行调试与语义澄清。载于《语义网：研究与应用》，第二届欧洲语义网会议（*ESWC*），第3532卷，收录于《计算机科学讲义》，第226–240页。施普林格。Lynn M. Schriml、Cesar Arze、Suvarna Nadendla、Yu-Wei Wayne Chang、Mark Mazaitis、Victor Felix、Gang Feng 和 W. Kibbe。2012。疾病本体：疾病语义集成的基础。《核酸研究》，40:D940–D946。Alessandro Solimando、Ernesto Jiménez-Ruiz 和 Giovanna Guerrini。2017。最大限度减少本体对齐中的保守性违规：算法与评估。《知识与信息系统》，51(3):775–819。Robyn Speer 和 Catherine Havasi。2012。在 ConceptNet 5 中表示一般关系知识。载于*LREC*。Mujeen Sung、Jinhyuk Lee、Sean Yi、Minji Jeon、Sung-dong Kim 和 Jaewoo Kang。2021。语言模型能否充当生物医学知识库？载于《*2021*年自然语言处理经验方法会议论文集》，第4723–4734页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。Adina Williams、Nikita Nangia 和 Samuel Bowman。2018。用于通过推理理解句子的广覆盖挑战语料库。载于《*2018*年北美计算语言学协会分会会议论文集：人类语言技术》，第

1（长篇论文），第1112–1122页，新奥尔良，路易斯安那州。计算语言学协会。

Hongbin Ye、Ningyu Zhang、Shumin Deng、Xiang Chen、Hui Chen、Feiyu Xiong、Xi Chen 和 Huajun Chen。2022。面向少样本学习的本体增强提示调优。*2022*年*ACM Web*大会论文集。

一个 OWL 本体概念表达式

描述逻辑 $\mathcal{SROIQ}$ 是 OWL 2本体语义的基础。给定顶层概念 $\top$、底层概念 $\bot$、命名概念 A、个体 a、角色（或属性） r 和非负整数 n，$\mathcal{SROIQ}$概念表达式的构造方式如下：

$$C, D ::= \top | \bot | A | (C \sqcap D) | (C \sqcup D) | \neg C | \exists r.C | \forall r.C | \geq n\ r.C | \leq n\ r.C | \exists r.Self | \{a\}$$

回顾解释 $I = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$的定义，其中 $\Delta^{\mathcal{I}}$ 是非空集合（论域），且 $\cdot^{\mathcal{I}}$将每个概念 C 映射到 $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$，将每个属性$r$ 映射到 $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$，并将每个个体 a 映射到元素 $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$。我们在表 5中给出了概念构造子的语义。

构造函数	语义
A	$A^{\mathcal{I}}$
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$\top$	$\Delta^{\mathcal{I}}$
$\bot$	$\emptyset$
$\exists r.C$	$\{x \mid$ 有一个 $r^{\mathcal{I}}$-后继属于 $x\ C^{\mathcal{I}}\}$
$\forall r.C$	$\{x \mid$ 所有 $r^{\mathcal{I}}$-后继都属于 $x\ C^{\mathcal{I}}\}$
$\geq n\ r.C$	$\{x \mid$至少有 $n\ r^{\mathcal{I}}$ 个后继属于 $x\ C^{\mathcal{I}}\}$
$\leq n\ r.C$	$\{x \mid$ 至多有 $n\ r^{\mathcal{I}}$ 个后继属于 $x\ C^{\mathcal{I}}\}$
$\exists r.Self$	$\{x \mid \langle x, x\rangle \in r^{\mathcal{I}}\}$
$\{a\}$	$\{a^{\mathcal{I}}\}$

表 5：OWL本体概念构造器的语义。

B 本体预处理

如果本研究使用的某些本体包含与包含关系采样无关的无意义概念（例如过时概念），和/或包含明显不自然的概念名称（或别名），我们会对所有本体进行**通用**预处理，然后对每个本体进行**单独**预处理。

通用预处理

•移除过时概念（由内置注释属性 owl:deprecated 标记）以及明显冗余的概念，例如 foodOn:stupidType。

• 使用 rdfs:label 作为提取概念名称的主要注释属性，但其字面值不可用时除外。提取出的概念名称均转换为小写，并删除其中的下划线“ ”。_

单独预处理

• Schema.org：概念名称（在此本体中定义）采用 Java 标识符风格；因此会被解析为自然表达式，例如将“*APIReference*”转换为“*API Reference*”。

• DOID：移除概念 doid:Disease，因为它是紧邻根概念 owl:Thing 之下的通用概念，会导致产生过多形如$C \sqsubseteq Disease$的简单包含关系。

• FoodOn：重构包含非自然语言文本的标签字符串，这些文本符合三种正则表达式模式（注意，(.*) 捕获需要保留的内容）：

(a) `[0-9]+ - (.*) \(.+\)`

(b) \('(.*)\(gs1', 'gpc\)'\)

(c)\('(.*)\(efsa', 'foodex2\)'\)

随后移除开头和结尾的空白字符。请注意，该本体中的概念有时会由 rdf:label 给出空字面值；在这些情况下，则改用注释属性 obo:hasSynonym 和 obo:hasExactSynonym。

• 基因本体：不进行个体单独处理。

C 对象属性口头表述

与语言化原子概念时只需使用其名称（或别名）不同，为了进行基础语法修正，我们采用一些简单规则将对象属性语言化。如果属性名称以被动动词、形容词或名词开头，我们便在句首追加“*is*”。例如，“*characteristic of*” 会改为“*is characteristic of*”；“*realised in*” 会改为“*is realised in*”。请注意，单词的词性标签由 Python 库 Spacy[19] 自动确定。

[19] https://spacy.io/

复杂概念 C	文本化 $\mathcal{V}(C)$
$BioRegulation \sqcap \exists negRegulate.ProlineBiosynProc$	“负向调控某种脯氨酸的生物调控 生物合成过程”
$ApoptoticProc \sqcap \exists partOf.Luteolysis$	“属于某种溶黄体过程的细胞凋亡过程”
$ConcnOf \sqcap \exists charOf.(fucose \sqcap \exists partOf.MaterialEnt)$	“某种岩藻糖特征的某物质的浓度 属于某个物质实体的一部分”
$\exists derivesFrom.(TimothyPlant \sqcup TrifoliumPratense) \sqcap$ $PlantFoodProd \sqcap Silage$	“由某种猫尾草制成的青贮饲料和植物食品 植物或红三叶草”
$Apple \sqcap \neg\exists hasPart.ApplePeel$	“苹果（整体或部分），且不是某种具有部分 *ap*-的事物。 *ple peel*

表 6：GO 和 FoodOn 的复杂 SI 数据集中的复杂概念文本化示例。请注意，在实际数据集中，命名概念和对象属性以其国际化资源标识符（唯一标识符）表示，而非表中所示的缩写名称。

D 复杂概念文本化示例

为了更清楚地了解我们的语言化方法如何运作，我们在表6中展示了从构建的复杂 SI 数据集中选取的一些典型概念语言化示例。

E 假定不相交性的实现选择

如第 3.1 节所述，对于我们采样为潜在负向包含关系的每个概念对 $(C,\ D)$，验证其不相交公理会耗费大量时间，因为我们需要反复将该不相交公理添加到本体中$\mathcal{O}$，进行推理，然后再移除该公理。因此，实践中可以使用以下条件替代可满足性检查：

(i) **不存在包含关系**：$\mathcal{O} \not\models C \sqsubseteq D$ 并且 $\mathcal{O} \not\models D \sqsubseteq C$；
(ii) **不存在共同实例**：不存在命名实例 a 属于 $\mathcal{O}$，使得 $\mathcal{O} \models C(a)$ 且$\mathcal{O} \models D(a)$。

如果 C 和 D 满足这两个条件，那么在将不相交公理 $C \sqcap D \sqsubseteq \bot$ 加入 $\mathcal{O}$ 后，它们**很可能是可满足的**。由于这两个条件无需对新公理进行**额外推理**，因此比对候选项反复进行可满足性检查高效得多。

需要注意的是，我们仍然需要进行**无共同后代**检查，以避免可预见的不可满足性。

(iii) **无共同后代**：$\mathcal{O}$ 中不存在命名的原子概念 A，使得 $\mathcal{O} \models A \sqsubseteq C$ 和 $\mathcal{O} \models A \sqsubseteq D$。

这是因为，如果存在一个名为A 的原子概念，它是推断出的子类（即后代）

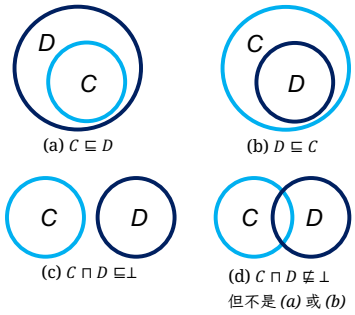

图 4：两个本体概念之间关系的基于集合的语义。

属于 C 和 D，那么 C 和 D 在 $\mathcal{O} \cup \{C \sqcap D \sqsubseteq \bot\}$ 中可能是可满足的，但 A 肯定是不可满足的（等价于 $\bot$）。

F 基于集合的包含样本解释

本节将进一步说明我们如何定义包含推理（SI）任务中的正向样本和负向样本。

回顾第2.1节中的定义，本体$\mathcal{O}$ 蕴含包含公理 $C \sqsubseteq D$，当且仅当它对**的每一种解释** $\mathcal{I}$ 都成立 $\mathcal{O}$。用基于集合的语义来说，这对应于图*4*中的情况（a）。在（*b*）、（*c*）或（*d*）这些情况下，至少存在一种解释 $\mathcal{I}$，使得我们可以找到一个个体x，它 $x^{\mathcal{I}} \in C^{\mathcal{I}}$ 且 $x^{\mathcal{I}} \notin D^{\mathcal{I}}$；因此 $\mathcal{O}$ **不蕴含**包含公理 $C \sqsubseteq D$。只有当（*a*）**并非对**的每一种解释 $\mathcal{O}$都成立时，才蕴含不包含。

不相交性对应于*(c)* 图 4 中，其中每种解释下，C 与 D 这两个集合均无重叠。非包含关系通常源于本体中的不相交公理（但不相交性**通常**蕴含来自不相交公理（但不相交性 $\forall x.C(x) \rightarrow \neg D(x)$是

		Atomic SI				Complex SI	
设置	biMNLI	Schema.org	DOID	FoodOn	GO	FoodOn	GO
roberta-base							
K=0	62.5 (6.5)	56.4 (3.6)	53.3 (4.0)	54.6 (4.4)	49.0 (2.4)	55.9 (3.6)	48.7 (3.1)
K=4	67.6 (5.2)	62.9 (5.2)	61.8 (6.7)	62.1 (4.2)	65.2 (5.0)	62.4 (3.2)	52.2 (7.1)
K=8	70.7 (4.5)	71.2 (4.5)	72.9 (5.7)	69.0 (5.2)	70.4 (5.1)	66.0 (4.4)	63.0 (5.0)
K=16	74.3 (3.3)	79.7（4.2）	83.4（2.5）	79.8（3.0）	78.3（3.0）	70.2（5.5）	73.8（4.0）
K=32	78.8 (1.1)	84.3 (2.0)	89.0 (1.4)	85.0 (1.1)	84.6 (2.5)	77.0 (1.5)	76.4 (2.5)
K=64	80.9 (1.5)	88.3 (1.5)	91.2 (0.7)	88.2 (0.7)	87.3 (0.8)	80.0 (2.0)	81.7 (1.4)
K=128	85.1 (1.0)	91.1 (0.7)	92.4 (0.7)	90.0 (0.7)	89.0 (0.8)	85.5 (1.3)	86.9 (1.5)
roberta-large							
K=0	68.7 (6.2)	61.7 (7.2)	59.8 (5.4)	60.1 (8.8)	54.6 (1.9)	56.1 (1.9)	50.4 (0.6)
K=4	78.1 (6.6)	69.4 (5.4)	74.0 (5.5)	71.6 (4.4)	67.6 (3.4)	64.1 (5.1)	56.9 (5.7)
K=8	83.0 (5.2)	78.5 (3.0)	84.4 (3.8)	77.0 (6.0)	75.3 (3.2)	71.3 (3.1)	64.2 (7.6)
K=16	87.5 (2.4)	84.4 (2.4)	87.6 (2.3)	83.4 (3.5)	82.8 (1.9)	76.2 (2.5)	76.4 (3.3)
K=32	89.9 (1.2)	87.3 (1.9)	92.3 (0.7)	88.9 (1.6)	87.7 (1.6)	80.8 (3.8)	81.6 (2.2)
K=64	90.8 (1.4)	90.4 (0.8)	92.6 (0.7)	90.9 (1.2)	90.1 (0.7)	84.1 (1.4)	86.2 (2.2)
K=128	93.0 (0.8)	92.9 (0.8)	93.4 (0.5)	92.2 (0.5)	91.0 (0.7)	88.4 (1.1)	90.2（1.0）
full	97.5	95.4	97.8	98.7	98.1	95.8	98.8

表 7：roberta-base 和 roberta-large 在 biMNLI、Atomic SI 和 Complex SI任务上的完整结果，其中每个单元格均表示“平均准确率（标准差）”，但多数投票和全监督设置除外，因为这些设置没有标准差。

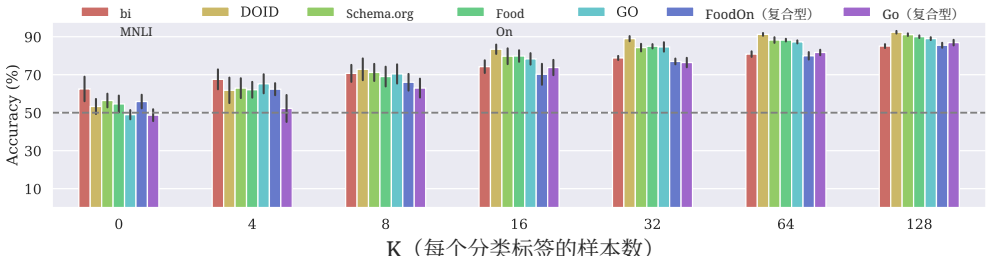

图 5：roberta-base 在 biMNLI、Atomic SI 和 Complex SI任务上的 K-shot 结果可视化，其中水平虚线表示多数投票。柱形的顺序与图例中的顺序相同。

比非包含关系 $\exists x.C(x) \wedge \neg D(x)$) 更严格。然而，本体通常在不相交性方面的规范不足，因此难以获得足够的负样本。为了折中，采用启发式方法是合理的。我们在论文正文第 3.1 节中采用的假定不相交性正是为此而设。在理想的设置中，我们在添加不相交公理后检查 C 和 D 的可满足性，并且要求 C 和 D没有共同后代，这样便可避免图 4 中的 (a) 和 (b) 情形，并降低出现 (d) 的可能性。即使采用本附录 E 提出的实用替代方案，不包含关系条件也能确保 (a) 和 (b) 不会被蕴含，而无共同后代和无共同实例条件则会降低出现 (d) 的可能性。因此，假定不相联合性是近似非包含关系的一种合理方法。

G 补充结果和图表

在论文正文中，我们报告了roberta-large 和 roberta-base 在 $K \in \{0, 4, 32, 128\}$上的部分结果（准确率分数和标准差）。在表 7中，我们展示了两个语言模型在 $K \in \{0, 4, 8, 16, 32, 64, 128\}$上的完整结果。

此外，我们在图 K中展示了 roberta-base 的 5次实验结果的可视化。观察结果与图 roberta-large中的结果一致，见图 3。