南京大學

NANJING UNIVERSITY

课程作业报告

大数据分析 11

学生姓名： 左皓升

学生学号： 211250074

2023 年 10 月

Link Prediction

Deep Bidirectional Language-Knowledge Graph Pretraining

1． 论文摘要 abstract 和 introduction 翻译

1.1 论文摘要 abstract 翻译

预训练的语言模型 (LM) 已经证明对各种下游 NLP 任务有帮助。最近的研究表明，知识图谱 (KG) 可以补充文本数据，提供结构化的背景知识，为推理提供有用的支撑。然而，这些研究还没有进行预训练，无法在大规模上学习这两种模式的深度融合，这限制了获取文本和 KG 完全联合表示的潜力。在这里，我们提出了 DRAGON（深度双向语言-知识图谱预训练），这是一种自我监督的方法，可以从大规模文本和 KG 中预训练深层联合语言-知识基础模型。具体来说，我们的模型将文本段和相关的 KG 子图作为输入，并双向融合来自两种模式的信息。我们通过统一两个自我监督的推理任务来预训练该模型，即掩码语言建模和 KG 链接预测。DRAGON 在各种下游任务上优于现有的 LM 和 LM+KG 模型，包括跨一般和生物医学领域的问答任务，平均绝对增益为 +5

1.2 论文 introduction 翻译

从大量原始数据中得到的预训练学习自监督表示，可以帮助各种下游任务。在大量文本数据上进行预训练的语言模型（LMs），例如 BERT 和 GPTs，在许多自然语言处理（NLP）任务中表现出色。这些模型的成功来自于通过自我监督大规模学习输入令牌的深度交互（情景化）表示。同时，大型知识图谱（KGs），如 Freebase、Wikidata 和 ConceptNet，可以向文本数据提供补充信息。KG 通过将实体表示为节点并将它们之间的关系表示为边缘来提供结构化的背景知识，还为结构化、多步骤推理实体提供了框架。文本数据和 KG 的双重优势激发了大规模预训练两个模态的深度交互表示的研究。

如何有效地将文本和知识图谱结合起来进行预训练是一个开放的问题，并且提出了挑战。对于文本和知识图谱，我们需要一个深层双向模型使两个模态进行交互，以及一个自我监督的目标以大规模对文本和知识图谱进行联合推理。现有的几项工作提出了自我监督预训练的方法，但它们以浅层或单向的方式融合文本和知识图谱。另一类工作提出了用于文本和知识图谱的双向模型，但这些模型侧重于在标记的下游任务上进行微调，而不是进行自我监督学习。因此，现有方法在建模和学习的深度交互方面可能有限制。

为了解决上述两个挑战并充分融合文本和知识图谱的优势，我们提出了 DRAGON（深度双向语言-知识图谱预训练），这是一种从文本和知识图谱中进行深度双向、自我监督预训练语言知识模型的方法。DRAGON 具有两个核心组件：一个跨模态模型，该模型以双向方式将文本和知识图谱作为输入，并学习两个模态之间的交互表示；以及一个双向自我监督目标，它通过最大化预测来自文本和知识图谱的跨模态掩码令牌的条件概率来学习联合推理。

具体来说，如图 1 所示，我们采用一个文本语料库和一个知识图谱作为原始数据，并通过从文本语料库中采样一个文本段并从知识图谱中提取相关子图来为模型创建输入，从而获得（文本，本地知识图谱）对。我们使用跨模态模型将此输入编码为融合表示，其中模型的每个层用语言模型编码文本并用图神经网络编码知识图谱，并用双向模态交互模块（GreaseLM）将两者融合在一起。我们通过统一两个自我监督推理任务来预训练此模型：（1）掩码语言建模（MLM），该任务对输入文本中的令牌进行掩码和预测；（2）链接预测，该任务在输入知识图谱中丢弃和预测边缘。这样做的直觉是，通过结合这两个任务，MLM 使模型将文本与结构化知识结合使用来进行文本中的掩码令牌推理（例如，在图 1 中，使用来自 KG 的“圆刷”-“艺术用品”多跳路径来帮助），链接预测使模型将知识图谱结构与文本上下文结合起来进行知识图谱中缺失链接的推理（例如，从文本中认识到“圆刷可用于头发”有助于这一点）。因此，这个联合目标使文本能够以知识图谱结构为基础，并使知识图谱能够以文本为背景同时进行情境化处理，从而产生一个深度统一的预训练语言知识模型，其中信息在文本和知识图谱之间进行双向流动以进行推理。

我们将在两个领域对 DRAGON 进行预训练：一是通用领域，使用 Bookcorpus 和 ConceptNet KG；二是生物医学领域，使用 PubMedcorpus 和 UMLSKG。我们发现，DRAGON 在跨域的各种下游任务上优于现有的 LM 和 LM + KG 模型。对于一般领域，DRAGON 在各种常识推理任务上，如 CSQA、OBQA、RiddleSense 和 HellaSwag 等，优于我们的基础 LM RoBERTa，平均绝对准确率提高了 8

知识增强 LM 预训练。知识集成是改进 LM 的积极研究方向。一条工作线是检索增强 LMs，它从语料检索相关文本，并将其作为附加知识集成到 LMs 中。与这些作品正交，我们专注于使用知识库作为背景知识，对实体和事实进行基础推理。

与我们的工作最相关的是将知识库集成到 LM 预训练中的工作。其中一项研究旨在向 LM 添加实体特征；一些工作使用 KG 实体信息或结构来创建额外的训练信号；一些工作将 KG 三元组信息直接添加到 LM 的输入中。虽然这些方法取得了重大进展，但它们通常以浅层或单向（例如，KG 到文本）的方式在文本和 KG 之间传递信息，这可能限制了完全联合推理两个模态的潜力。为了改进上述方法，我们提出通过深度跨模态模型和联合自我监督来双向交互文本和 KG，使文本和 KG 相互接地和情境化。我们发现这可以改善模型在各种推理任务上的性能（第 3 节）。另一个区别是，这个领域中的现有工作通常侧重于从 KG 中添加实体或三元组级别的知识到 LM 中，并专注于解决实体/关系分类任务。我们的工作显著扩大了这个范围，因为我们使用更大的 KG 子图（200 个节点）作为输入，以在 KG 和文本之间实现更丰富的情境化，我们在包括 QA、推理和文本分类任务在内的更广泛的 NLP 任务上实现了性能提升。

知识图谱增强问答。各种工作设计了 KG 增强的推理模型用于问答。特别是最近的工作，如 QA-GNN 和 GreaseLM ，表明知识图谱可以通过其图结构来支撑对实体的推理，并有助于复杂的问题解答（例如，否定，多跳推理）。这些工作通常侧重于在特定的 QA 数据集上训练或微调模型。相比之下，我们对此进行了泛化并将 KG 增强的推理集成到通用预训练中。我们的动机是自监督预训练允许模型从更大型和更多样化的数据中学习，这有助于学

习文本与 KG 之间更丰富的交互，并获得超越特定 QA 任务的更多样化的推理能力。我们发现我们的预训练方法（DRAGON）对各种下游任务的基线 QA 模型（例如 GreaseLM）提供了显著的推动力。这开辟了一个新的研究途径，将各种精心设计的 QA 模型扩展到预训练。

知识图谱表示学习。我们在预训练中使用的链接预测任务受到知识图谱表示学习的研究启发。链接预测是知识图谱的基本任务，各种工作研究学习 KG 实体和关系嵌入的方法用于链接预测，例如 TransE，DistMult 和 RotatE。一些作品还使用文本数据或预训练的 LM 来帮助学习 KG 嵌入和链接预测。虽然这些作品关注 KG 一侧的表示，但我们将范围扩展到使用 KG 一侧的目标（链接预测）与文本侧的目标（语言建模）联合训练相互交互的文本-KG 模型。

2． 问题描述

文本预训练在许多 NLP 任务中表现出色，同时，大型知识图谱（KGs）可以向文本数据提供补充信息，为实体间的关系提供结构化的背景知识。如何有效地将文本和知识图谱结合起来进行预训练是一个开放的问题。现有的工作要么研究的方向不是预训练，要么侧重于在标记的下游任务上进行微调。为了这个挑战并充分融合文本和知识图谱的优势，作者提出了 DRAGON 模型，深度双向语言-知识图谱预训练。

3． 输入、输出、模型算法描述

3.1 输入

在预训练中，输入的是文本数据和知识图谱

3.2 输出

在预训练中，输出的是文本——知识图谱对，以及模型的损失函数

3.3 模型算法描述

初始化模型参数。

对于每个预训练步骤，执行以下操作：从数据集中随机采样一个联合文本-知识图谱对。对于每个文本令牌 wi，以 15 执行掩码语言建模任务，使用上下文化令牌向量 Hi 和特殊令牌 [MASK] 来预测原始令牌 wi。执行知识图谱链接预测任务，保留输入知识图谱中的一些边缘并预测它们。计算模型的损失函数，包括掩码语言建模损失和知识图谱链接预测损失。使用反向传播算法更新模型的参数。如果达到了指定的训练步数或验证集性能指标，则终止预训练过程。

使用预训练的模型进行下游任务。

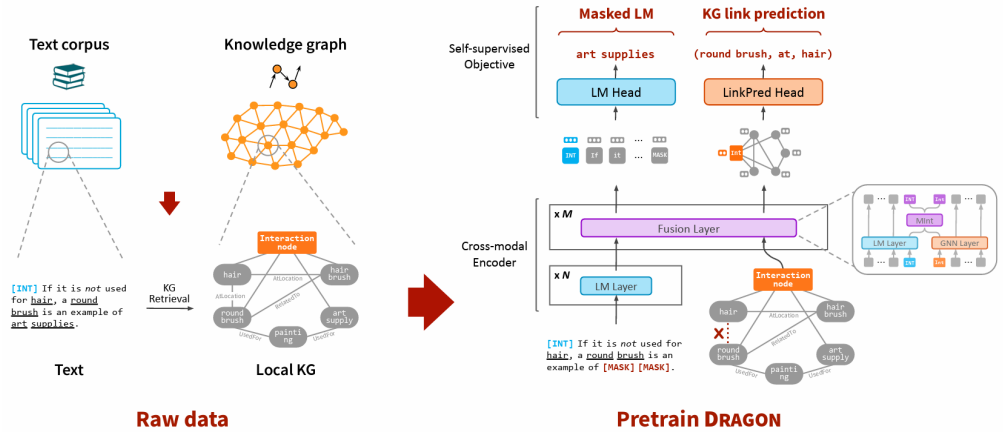

图 1:

3.4 评价指标及其计算公式

作者使用了两个子监督推理任务，MLM 和 Link prediction。两个任务各自有自己的损失函数。作者讲两个损失函数集合起来，给出了一个评价函数

$$\mathcal{L}_{\text{MLM}} = -\sum_{i \in M} \log p(w_i \mid \mathbf{H}_i).$$

图 2:

$$\mathcal{L}_{\text{LinkPred}} = \sum_{(h,r,t) \in S} \left(-\log \sigma(\phi_r(\mathbf{h}, \mathbf{t}) + \gamma) + \frac{1}{n} \sum_{(h',r,t')} \log \sigma(\phi_r(\mathbf{h}', \mathbf{t}') + \gamma) \right),$$

图 3:

$$\mathcal{L} = \mathcal{L}_{\text{MLM}} + \mathcal{L}_{\text{LinkPred}}.$$

图 4:

4． 对比方法及这些对比方法的引用论文出处

4.1 对比方法

RoBERTa QAGNN GreaseLm

4.2 对比方法的引用论文出处

RoBERTa：Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: Arobustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692, 2019.

QAGNN：Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. QAGNN: Reasoning with language models and knowledge graphs for question answering. In NorthAmerican Chapter ofthe Association for Computational Linguistics(NAACL), 2021.

GreaseLm： XikunZhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher DManning, and Jure Leskovec. Greaselm: Graph reasoning enhanced language models forquestion answering. In International Conference on Learning Representations(ICLR), 2022

5． 结果

作者一共提供了四个数据集，在 train 的过程中，前三个数据集都取得了和论文中一致的结果，但是第四个数据集并未取得和论文中相近的结果。由于在租用的 3090 服务器上，第四个数据集的 train 需要 20 多小时，没有足够时间再进行训练去排查问题可能所在。我们在前三个数据集的训练中，看到了和论文中十分接近的结果。

以下展示 wandb 网站截图和输出文件截图。完整的输出、log、json 文件都在代码 train 文件夹中

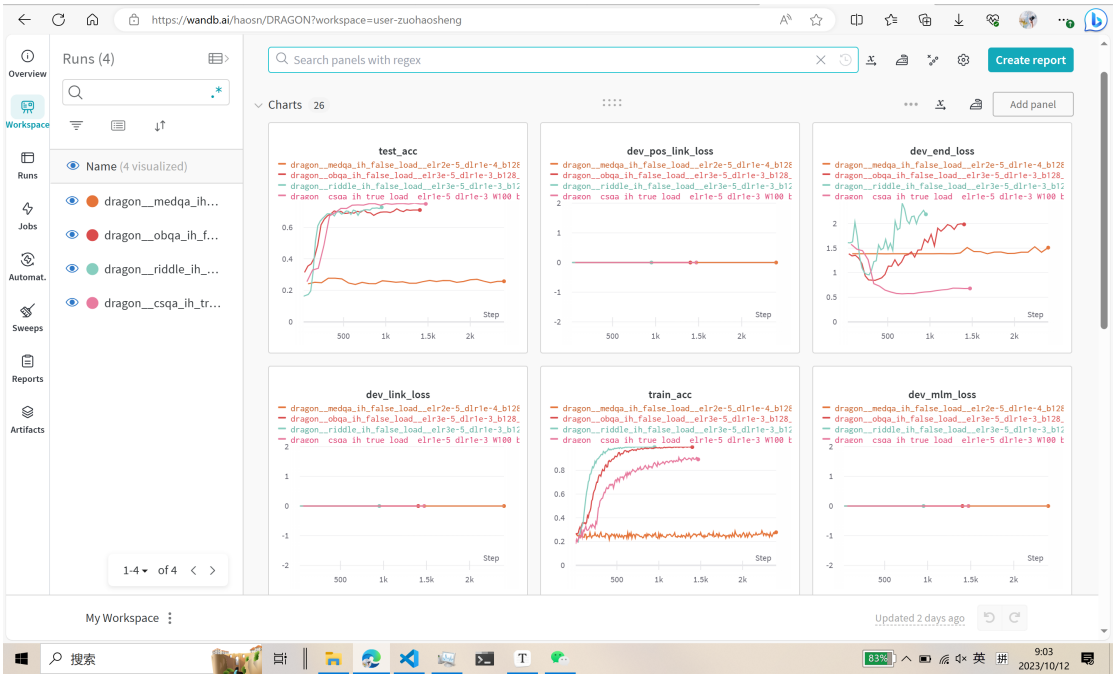

图 5:

```
test_acc 0.7493956486704271
-----------------------------------------------------------------------
| epoch  21 | step  1474 | dev_acc  0.7912 | test_acc  0.7494 |
-----------------------------------------------------------------------
```

图 6: csqa

```
test_acc 0.71
-----------------------------------------------------------------------
| epoch  35 | step  1404 | dev_acc  0.7180 | test_acc  0.7100 |
Epoch: 100%|
Epoch: 100%|
```

图 7: obqa

```
test_acc 0.7274509803921568
------------------------------------------------------------------------
| epoch  33 | step   952 | dev_acc  0.7006 | test_acc  0.7275 |
------------------------------------------------------------------------
```

图 8: riddle

```
test_acc 0.25765907305577374
------------------------------------------------------------------------
| epoch  29 | step  2400 | dev_acc  0.2508 | test_acc  0.2577 |
------------------------------------------------------------------------
```

图 9: medqa

对于作者提供的训练好的模型，进行下载检验准确度，获得结果和论文中结果一致

以下为部分控制台输出截图，完整截图在代码 eva 文件夹中

```
args.test_adj data/csqa/graph/test.graph.adj.pk
Dev/Test batch: 100%|
1/621 [00:43<00:00, 14.25it/s]
dev_acc  0.7920, test_acc  0.7615
--------------------------------------------------------------------------------
(dragon) root@autodl-container-2bae408949-d9eac5da:~/autodl-tmp/dragon#
```

图 10: csqa

```
Dev/Test batch: 100%|
Dev/Test batch: 100%|
--------------------------------------------------------------------------------
dev_acc  0.7080, test_acc  0.7280
--------------------------------------------------------------------------------
( ragon) root@autodl-container-645f11a73c-e71b4666:~/autodl-tmp/dragon#
```

图 11: obqa

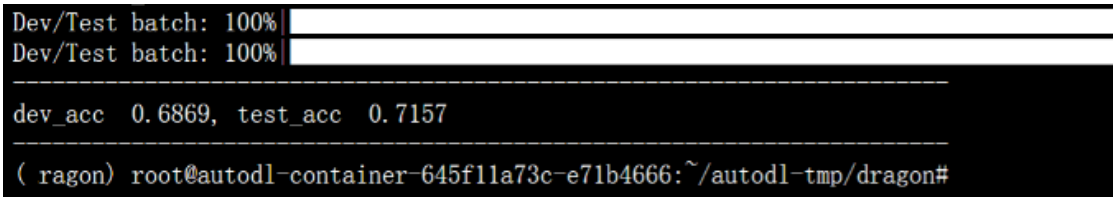

图 12: riddle

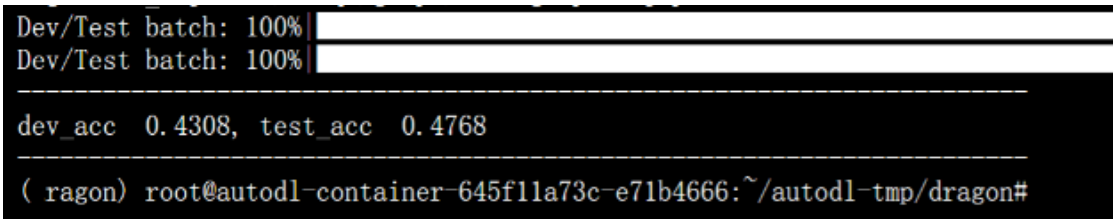

图 13: medqa