NLP Workbench：高效且可扩展地集成最先进的文本挖掘工具

Peiran Yao、Matej Kosmajac、Abeer Waheed、Kostyantyn Guzhva、Natalie Hervieux、Denilson Barbosa阿尔伯塔大学计算机科学系{peiran, denilson}@ualberta.ca

arXiv:2303.01410v1 [cs.CL] 2 Mar 2023

摘要

NLP Workbench 是一个基于万维网的文本挖掘平台，非专业用户可借助最先进的文本挖掘模型，从大规模语料库中获得语义理解。该平台基于最新的预训练模型和来自学术界的开源系统构建，提供语义分析功能，包括但不限于实体链接、情感分析、语义解析和关系抽取。其可扩展的设计使研究人员和开发人员能够轻松替换现有模型或集成新模型。为提高效率，我们采用微服务架构，以便分配加速硬件并实现计算并行化。本文介绍了 NLP Workbench 的架构，并讨论了我们在设计过程中遇到的挑战。我们还探讨了 NLP Workbench 的多种用例，以及相较于其他方法使用该平台的优势。该平台目前仍在积极开发中，其源代码依据 MIT 许可证发布[1]。平台网站[2] 和一段简短的视频[3] 也可供观看，以展示我们的平台。

1 引言

文本挖掘，又称为文本分析或文本分析，是用户与机器辅助分析工具交互的过程。这些工具将自然语言文本转换为结构化数据，从而帮助用户从文本中获取洞见和新知识（Feldman 和 Sanger，2006年a）。二十多年来，人们一直在为商业智能、分析社会学和医学等不同领域构建文本挖掘系统（Hearst，1999年），充分展现了其不可替代的价值。文本挖掘中的分析工具通常采用机器学习（ML）和自然语言处理（NLP）模型的形式，涵盖广泛的机器学习和自然语言处理子领域，例如实体链接、情感分析、关系抽取和文本摘要。

近年来，参与文本挖掘的自然语言处理几乎所有子领域都在迅速发展，基准测试的纪录不断被刷新[4]。越来越多的研究人员开始采用一种良好做法，即向公众发布代码和模型[5]，以解决该领域的可复现性和可访问性问题[6]。尽管各方努力让新模型更易于使用，但非专业用户，例如数字人文研究者和商业分析师，在尝试应用最新模型时仍面临入门障碍。一些突出问题包括：(1) 运行模型所需的软件技术栈各不相同；(2) 输入和输出格式缺乏标准化且不一致；(3) 缺少便于应用模型和可视化结果的用户友好型界面；(4) 计算资源和网络资源受限。我们构建了 NLP Workbench，旨在解决这些问题，进一步弥合前沿开放式自然语言处理研究与非专业人士在文本挖掘应用中使用这些模型和工具之间的差距。

NLP Workbench 的设计遵循两项基本原则：对于开发者和自然语言处理研究人员，应能快速、轻松地适配现成的模型和工具；对于社会学家等非专业用户，则应提供便于使用的界面，以进行文档级和语料库级分析。基于这些原则，NLP Workbench 提供了以下主要功能：

平台 NLP Workbench 将语料库管理、文本挖掘工具和可视化整合于

EACL 2023 系统演示论文终稿。

[1] https://github.com/U-Alberta/NLPWorkbench/

[2] https://newskg.wdmuofa.ca

[3] https://vimeo.com/801006908

[4] http://nlpprogress.com

[5] https://paperswithcode.com

[6] https://aclrollingreview.org/responsibleNLPresearch/

一个平台之中。它提供不断扩充的模型和工具列表，这些模型和工具均基于前沿研究，目前支持命名实体识别、实体链接、关系抽取、语义解析、摘要生成、情感分析和社交网络分析等功能。

交互 系统包含一个网页界面，用于用户交互，并能够在文档和语料库层面可视化模型结果。用户可以选择在给定文档上交互式地应用模型，并保存结果以供未来查询；也可以对语料库中选定的文档批量应用模型。

架构 在开发方面，NLP Workbench 采用容器化，使新模型能够独立于现有模型的软件栈进行添加。在部署方面，其微服务架构允许将模型以分布式方式部署在满足各个模型计算和网络要求的机器上，从而实现水平扩展。

界面 NLP Workbench 中的工具可以通过多种方式访问。除网页界面外，非专业用户还可以通过浏览器扩展将新文档导入平台。对于开发者和研究人员，NLP Workbench 提供 RESTful API 和远程过程调用（RPC）接口，便于与其他应用程序和流水线集成。

2 相关工作

Hearst （1999年）和Cunningham 等人（2002年）指出，有效的文本挖掘系统包含三个关键方面：文本文档集合（语料库）的管理、对集合应用文本处理算法，以及结果的可视化。LINDI（Hearst，1999年）是此类系统的早期原型，用于基因功能发现。GATE（Cunningham 等人，2002年）是一个至今仍在维护的框架，为这三个方面提供了统一的架构。同样，NLP Workbench 也试图在单一平台中涵盖这三个方面。我们的平台使用容器化微服务，而非为每个处理模块使用 Java 类，从而避免了底层编程框架对实现自然语言处理算法造成的限制。Voyant Tools（Rockwell 和 Sinclair，2016年）是另一个基于万维网的平台，提供语料库分析工具以及编写用户定义脚本的功能。其内置工具大多局限于基于计数的统计和可视化，而我们正在集成大型深度学习模型。UIMA（Ferrucci 和 Lally，2004）试图为管理语料库和自然语言处理算法定义一种标准协议。与我们的应用不同，它更面向开发者；我们的应用则提供了一个完整系统，用户可以直接用它进行分析。

过去几十年中，人们开发了大量专注于构建自然语言处理流水线的自然语言处理工具包，包括 Stanford CoreNLP（Manning 等人，2014年）、OpenNLP[7], NLTK（Bird 等人，2009）、spaCy（Honnibal 等人，2020年）以及 Transformers（Wolf 等人，2020年）。这些工具包可用于解决文本挖掘系统中的文本处理算法问题，但无法提供完整的解决方案。将新模型纳入这些工具包的典型策略，是在工具包的框架中重新实现该模型；而我们则尽可能复用研究人员发布的代码和模型。

有些工具专门用于可视化。例如，Blloshmi 等人（2021年）和Cohen 等人（2021年）开发了用于可视化语义解析器结果的工具——我们的系统也提供这项功能。

有多个库和工具能够管理来自不同来源的语料库或模型。例如，Datasets 库（Lhoest 等人，2021年）提供了访问常见自然语言处理数据集的接口，而 DataLab（Xiao 等人，2022年）是一个用于检查和分析数据集的平台。Transformers（Wolf 等人，2020年）可以访问 Hugging Face Hub 上的模型和数据集[8]。除了语料库和模型，NLP Workbench 还整合了来自多个来源的代码。

3 架构

NLP Workbench 构建于多种开源软件之上，并整合了许多研究项目中的代码和模型。我们在设计架构时，充分利用这些软件和项目提供的现成功能，并尽量减少集成研究项目中新代码所需的工作量。

3.1 工作流程

图1 概述了用户如何与 NLP Workbench 交互，以及

[7] https://opennlp.apache.org [8] https://huggingface.co/docs/hub/index

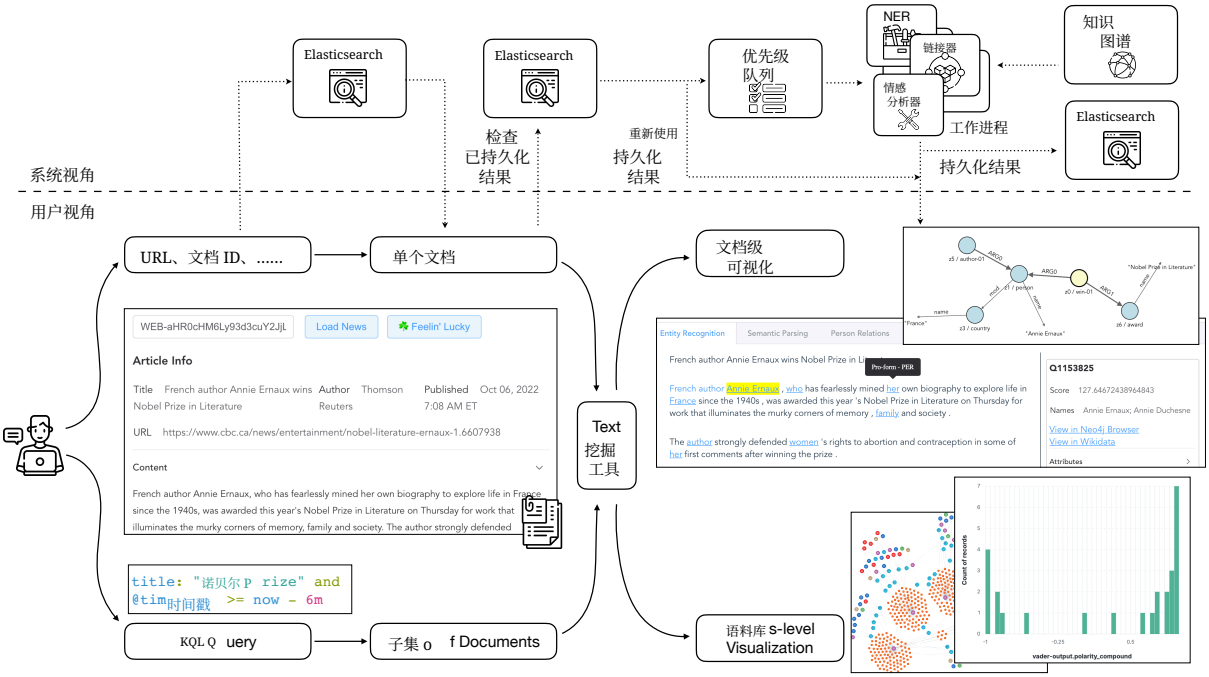

图1：自然语言处理工作台的工作流程，从用户和系统的角度进行描述，如 §3.1所述。对于文档级可视化，我们展示了命名实体识别、共指消解、实体链接和语义解析的用户界面。对于语料库级可视化，此图包含根据诺贝尔奖官方账号的一条推文构建的社交网络图，以及内置新闻语料库样本的情感极性分数分布。图标由 Freepik 在 Flaticon 上创作。

系统处理请求。

用户视角 从用户的角度来看，用户可以选择对由 URL 或 ID 指定的单个文档应用文本挖掘工具，也可以通过查询对一组文档批量应用这些工具。查询使用 Kibana 查询语言（KQL）[9], 编写，这是一种简单直观的文本查询语言。单个文档的工具输出可在网页界面中可视化，每种工具都有单独的面板。借助 Kibana Lens[10],，用户可以将根据多个文档的输出计算得出的统计数据可视化，例如情感极性分数的分布。连接的 Neo4j Browser[11] 提供了一个交互式网页界面，用于探索根据语料库构建的社交网络。

系统视角 从系统角度来看，语料库和文本挖掘工具的输出都会存储在 Elasticsearch 中并建立索引[12]。Elasticsearch 是一款文档索引、搜索和分析引擎。通过存储工具输出并为其建立索引，我们可以重复利用先前的结果，避免重复计算，从而提高效率。此外，Elasticsearch 还提供了便捷的工具，可根据文本挖掘工具的输出筛选文档并可视化统计数据，这些功能对下游分析非常有用。

如果确实有必要运行工具，任务就会被加入优先队列。用户检查单个文档并对其应用工具时发出的临时交互请求，优先级高于在后台运行的批量请求。每种工具或模型都有工作进程来处理队列中的任务。这样一来，即使工作进程数量有限（这在实践中通常如此，因为深度学习模型往往需要大量资源，无法并行运行多个实例），用户进行交互式分析时也能体验到较低的延迟。

3.2 流水线处理与调度

文本挖掘工具通常依赖其他工具或自然语言处理模型的输出，并以流水线的形式构建。例如，实体链接和关系抽取都需要命名实体识别和共指消解。为确保效率，应避免重新计算已有工具的输出，并且应尽可能并行运行工具。

[9]https://www.elastic.co/guide/en/kibana/current/kuery-query.html[10]
https://www.elastic.co/kibana/kibana-lens[11]
https://neo4j.com/developer/neo4j-browser/[12]
https://www.elastic.co/what-is/elasticsearch

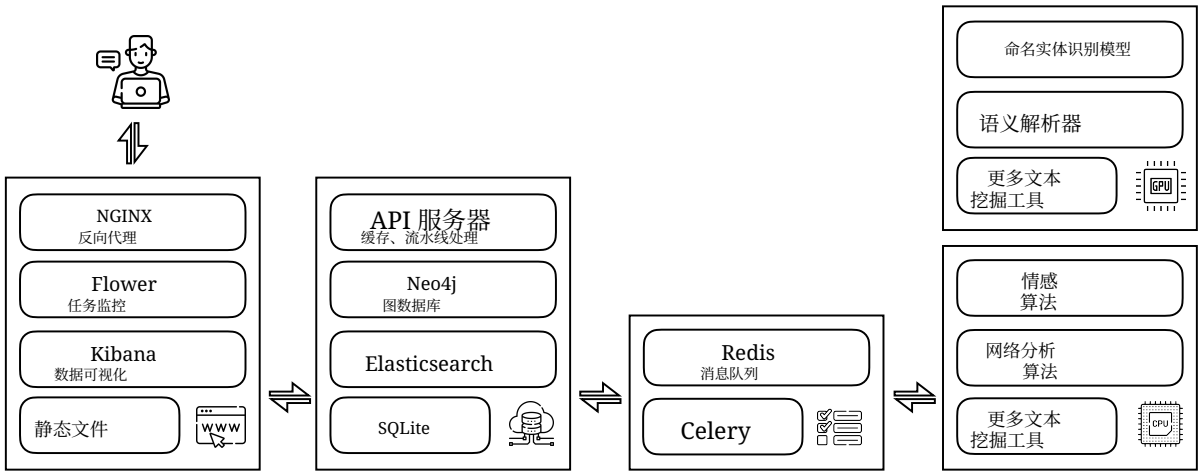

图2：NLP Workbench 的微服务架构。每个矩形表示一台物理机器，其能力由右下角的图标标示。每个圆角矩形表示一个容器，内部文字标示其提供的工具和功能。容器与物理机器的分配仅用于示意；系统部署到生产环境时，会根据需求进行调整。

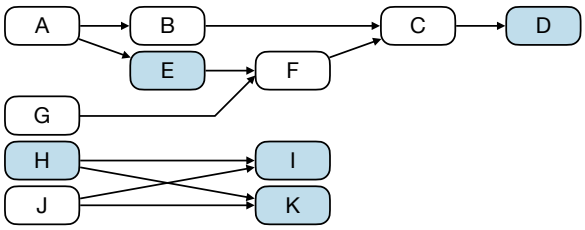

图3：批处理任务及其依赖关系的有向无环图示例。阴影节点表示用户请求在文档上运行的工具，非阴影节点表示为阴影节点提供输入所需的工具。

此外，除了按照第 3.1 节所述持久化并重复使用输出之外，我们还设计了一个流水线与调度系统，该系统能够自动检测工具之间的依赖关系，并以消除重复计算、促进并行处理的方式安排任务。

工具的依赖关系可以自然地表示为一个有向无环图（DAG），其中入边表示依赖关系。当用户请求在单个文档上运行多个工具时，我们会将这些工具的直接和传递依赖关系汇集到一个图中，如图3所示。图中的连通分量均为有向无环图。在每个有向无环图中，工具按拓扑顺序运行；彼此不相连的有向无环图则并行执行。

在图3所示的示例中，用户请求在一份文档上运行工具E、D、H、I和K 。调度器将自动查找所有依赖关系（A 到 K），并行运行两条链：A-B-E-G-F-C-D 和 H-J-I-K[13]。

3.3 容器化微服务

集成第三方代码的一大障碍是依赖地狱问题：要找到一组彼此兼容的软件库依赖的版本，这是一个 NP 完全问题（Burrows，2005年；Cox，2016年），而且实际上可能根本不存在兼容的版本组合。深度学习模型尤其如此（Han 等人，2020年；Huang 等人，2022年），它们通常需要特定版本的软件库。以流行的深度学习框架为例，TensorFlow（Abadi 等人，2016年）2.0 引入了不向后兼容的破坏性 API 变更。TensorFlow 和 PyTorch（Paszke 等人，2019年）都是使用特定版本的 CUDA（Nickolls 等人，2008年）和 cuDNN（Chetlur 等人，2014年）编译的，这使得不同版本的框架难以共存。手动修改代码以使其兼容某个特定版本的库，往往既费时又容易出错（Han 等人，2020年）。

在部署时，一个实际问题是，往往很难找到一台能够满足所有组件计算和网络需求的物理机器，或者这样做成本高昂：深度学习模型需要图形处理器进行推理，数据库管理系统会占用大量

[13]有向无环图存在多种有效的拓扑排序。

内存和磁盘空间，而万维网服务器需要访问互联网。我们设计的解决方案是能够将自然语言处理工作台的各个组件部署在多台机器上。

我们通过将文本挖掘工具和基础设施组件都部署为容器化微服务，一次性解决上述两个问题。每个组件都部署为 Docker 容器（Merkel，2014年），其中封装了软件及其依赖关系。容器通过 Celery 提供的 RPC 和消息队列功能相互通信[14]。图2 展示了容器化微服务如何部署在具有不同能力的独立机器上。这种微服务架构让我们能够克服技术异构性问题，并在需要时简化水平扩展（Newman，2015）。

4 个组件

NLP Workbench 已包含多种文本挖掘工具和模型。大多数组件源自各自子领域的前沿研究。其他组件则是基线实现，用于展示 NLP Workbench 的可扩展性，说明开发者可以轻松地集成新工具，并基于现有工具构建流水线。正如 §3 所述，灵活的模块化设计有一项优势：所有内置工具和模型都可以轻松替换或升级。NLP Workbench 中现有的工具和模型包括：

命名实体识别这项任务称为命名实体识别，即识别文本中提及的人物、组织和地点等实体。我们采用了 PURE 中的命名实体识别模型（Zhong 和 Chen，2021年），该模型只需微调 BERT（Devlin 等，2019年）便取得了良好的效果。

共指消解为了确定代词指代的是哪个实体，我们采用了Cunningham 等人（2002年）提出的启发式算法，该算法基于近因性和类型一致性。

实体链接 将文本中的提及消歧并链接到维基数据（Vran-deˇci´c 和 Krötzsch，2014年）中的实体。候选实体通过对名称进行模糊匹配生成。除了名称相似度之外，候选实体的排序还会利用上下文句子嵌入与候选实体在维基百科和维基数据中的描述之间的余弦相似度（Reimers 和 Gurevych，2019年）。

关系抽取用户可以从文本中抽取结构化事实，并以知识三元组的形式表示，例如（安妮·埃尔诺，国家，法国）。底层模型（Mesquita 等人，2019年）结合句法、语义特征和 BERT 嵌入，预测实体之间的关系。

语义解析语义解析将句子的含义转换为结构化表示，使用户无需关注句子的表达形式，即可获取诸如谁在何时何地对谁做了什么等信息。NLP Workbench 使用 AMR-BART（Bai 等人，2022年），这是一种基于 BART（Lewis 等人，2020年）并在大型图语料库上预训练的序列到序列模型，可将句子解析为 AMR图（Banarescu 等人，2013）。

摘要生成 我们基于语义解析构建了一个应用，用于生成文档中与人物相关事件的自然语言摘要，部分目的是展示在 NLP Workbench 中构建流水线的简便性。对于文档中的每个句子，我们都会对其 AMR图进行剪枝，仅保留符合主语-谓语-宾语模式的节点和边，其中主语或宾语必须是人物。然后，我们使用 AMRBART 将剪枝后的 AMR图转换为自然语言。

情感分析 VADER（Hutto 和 Gilbert，2014年）用于预测文档的情感。它是一种快速、准确且基于规则的算法，专为社交媒体帖子优化。该算法会生成情感极性分数，可用于将情感分类为正面、中性或负面。

社交网络分析 对于由社交媒体帖子组成的语料库，NLP Workbench 配备了一种工具，可根据帖子构建社交网络互动图。该工具由图数据库 Neo4j[15]，提供支持，可用于可视化网络并进行分析，例如运行 PageRank（Page 等人，1999年）等中心性算法，以识别有影响力的用户。

5 个用例

文本挖掘已被证明在众多领域中颇有价值，例如公司金融、专利检索、生命科学等领域（Feldman

[14] https://docs.celeryq.dev

[15] https://neo4j.com/

and Sanger，2006b）。NLP Workbench 作为一个成熟的文本挖掘平台，已经或有望应用于这些领域中的许多领域。本节将介绍其中的一些应用案例。

数字人文 易于使用且可靠的自然语言处理工具，对于 Linked Infrastructure for Networked Cultural Scholarship (LINCS)[16] 等数字人文项目十分有用。LINCS 项目的一个重要组成部分，是从自然语言文化遗产文本中生成关联数据。关键步骤包括命名实体识别、共指消解、实体链接和关系抽取，这些功能在 NLP Workbench 中均可使用。通过简洁的用户界面，数字人文学者及其学生可以选择适合自身数据的可用模型，并轻松地将这些步骤连接成自定义工作流程。

人工智能、数据与冲突中心 (CAIDAC)[17] 汇聚了关注武装团体和极端主义者如何利用社交媒体宣传其主张的人权学者。NLP Workbench 为这类研究人员提供了便捷的工具，用于收集推文[18] 并进行分析。研究人员可以使用感兴趣的自然语言处理工具处理帖子，还可以对语料库背后的社交网络进行可视化、探索和分析。所有这些任务都可以通过易于使用的界面完成，用户无需编程，因此他们能以远短于以往所需的时间开展规模更大、数量更多的研究。

商业分析商业分析师会提出这样的问题：“近期关于苹果公司的新闻报道是正面还是负面？”这类问题可以通过 NLP Workbench 轻松解答。对新闻文章进行命名实体识别与实体链接后，分析师可以进行**语义搜索**，找出与苹果公司相关的文章，而不是关于水果苹果的文章。然后，分析师可以使用情感分析工具，并通过 Kibana Lens 将情感极性分数的分布可视化，如图 1 右下角的截图所示1。

自然语言处理研究 NLP Workbench 中的所有自然语言处理模型均可通过 RESTful API 和 RPC 访问，也可直接作为容器使用。研究人员若希望使用自己的数据对这些模型进行推理，可使用其提供的接口，无需搭建运行模型所需的环境。

6 路线图

NLP Workbench 仍处于开发早期阶段，我们正在积极改进系统。除了提升可用性和稳定性以及提供安全更新外，我们还计划在不久的将来开发以下主要功能：

人在回路中的自然语言处理在网页界面中添加标注支持后，用户便可以对模型的输出提供反馈。这将有助于研究人员收集领域专用标注数据，并以人在回路的方式提升模型性能（Wang 等人，2021年）。

改进语料库管理 管理文档集合是文本挖掘中的一个关键方面（Hearst，1999年；Cunningham 等人，2002年）。目前，语料库是手动导入 Elasticsearch 的，或通过抓取社交媒体创建。我们希望改进用户访问文档集合的方式。为此，可以将 NLP Workbench 连接到 Datasets（Lhoest 等人，2021年）和 DataLab（Xiao 等人，2022年），这些平台已提供许多常用的文本数据集。除了从社交媒体抓取内容外，我们还计划支持通过搜索引擎进行网络搜索来创建新语料库。

更多文本挖掘工具 NLP Workbench 的可扩展设计使我们能够通过替换现有工具和模型，及时采用更好的模型，并将新兴的文本挖掘工具集成到系统中。例如，我们希望添加声明抽取模型，以便支持事实核查任务（Hassan 等人，2017年）。

多模态分析 社交媒体帖子通常会提及或包含其他受关注模态的信息（图像、视频、音频）。与此同时，人们越来越关注如何将自然语言处理模型和分析与从视频及其他来源提取的知识建立基础关联。虽然在 NLP Workbench 中添加对不同媒体的处理支持，只需增加更多自然语言处理工具即可，但我们有兴趣将这些模型集成起来，从而尽可能实现自动化的协同训练或基础关联。

7 结论

我们介绍了 NLP Workbench，这是一个满足文本挖掘系统三大主要需求的平台：

[16] https://lincsproject.ca [17] https://www.tracesofconflict.com/ [18] Telegram 和其他平台的支持正在开发中。

语料库管理、文本挖掘工具和用户界面。我们说明了 NLP Workbench 的哪些设计特点使其高效且可扩展，以及它如何应用于各种场景。

局限性

如 §6所述，我们已经指出 NLP Workbench 中尚未实现的若干重要功能：该平台需要具备支持人类参与式人工智能的标注功能；需要能够访问常用的公共语料库；还应包含文本挖掘工具，例如用于主张提取的工具。有些固有局限，即使是 NLP Workbench 中最先进的模型也无法解决。例如，知识图谱可能不包含长尾实体，而当前的实体链接模型无法识别知识图谱之外的实体（Shen et al., 2023年）。这将导致长尾实体始终被错误链接。除社交网络分析外，我们目前的设计没有为其他语料库级分析提供用户界面或模型，例如主题建模（Blei et al., 2003）。最后，NLP Workbench 中的所有自然语言处理模型和算法都针对英文文本。尽管我们已能够使用 Marian MT 将其他语言的语料库翻译成英文来进行处理（Junczys-Dowmuntetal., 2018年），但直接使用针对其他语言训练的模型能否提升性能，目前尚不明确。

伦理声明

通过封装模型，NLP Workbench 降低了非专业人士使用最先进 AI 模型的门槛。微服务架构允许将模型部署在具备不同能力的多台服务器上，而非单台全能服务器上，也让这个文本挖掘平台更易于使用。将第三方模型容器化，也有助于提高可复现性和透明度。人们已尝试使用 NLP Workbench 分析数据集，以帮助理解与战争和恐怖主义相关的宣传、错误信息和虚假信息。不过，必须提醒用户，NLP Workbench 使用未经修改的第三方数据和模型，因此通过 NLP Workbench 获得的输出不可避免地会受到数据集和模型中固有偏见的影响。

致谢

这项工作部分由加拿大自然科学与工程研究委员会（NSERC）、加拿大创新基金会（CFI）、AI4Society[19] 以及丰业银行的捐赠资助。部分计算资源由加拿大数字研究联盟提供。

参考文献

Martín Abadi、Paul Barham、Jianmin Chen、Zhifeng Chen、Andy Davis、Jeffrey Dean、Matthieu Devin、Sanjay Ghemawat、Geoffrey Irving、Michael Is- ard、Manjunath Kudlur、Josh Levenberg、Rajat Monga、Sherry Moore、Derek Gordon Murray、Benoit Steiner、Paul A. Tucker、Vijay Vasudevan、Pete Warden、Martin Wicke、Yuan Yu 和 Xiao- qiang Zheng。2016年。Tensorflow：一个用于大规模机器学习的系统。载于第*12*届 *USENIX* 操作系统设计与实现研讨会（*OSDI 2016*），美国佐治亚州萨凡纳，*2016*年*11*月*2*日至*4*日，第265–283页。USENIX 协会。

Xuefeng Bai、Yulong Chen 和 Yue Zhang。2022年。用于 AMR解析和生成的图预训练。载于计算语言学协会第*60*届年会论文集（第*1*卷：长篇论文），第6001–6015页，爱尔兰都柏林。计算语言学协会。

Laura Banarescu、Claire Bonial、Shu Cai、Madalina Georgescu、Kira Griffitt、Ulf Hermjakob、Kevin Knight、Philipp Koehn、Martha Palmer 和 Nathan Schneider。2013年。用于语义库构建的抽象意义表示。载于第七届语言学标注研讨会及话语互操作性会议论文集，第178–186页，保加利亚索非亚。计算语言学协会。

Steven Bird、Edward Loper 和 Ewan Klein。2009年。《*Python*自然语言处理》。O'Reilly Media Inc.

David M. Blei、Andrew Y. Ng 和 Michael I. Jordan。2003年。潜在狄利克雷分配。机器学习研究杂志，3：993–1022。

Rexhina Blloshmi、Michele Bevilacqua、Edoardo Fabiano、Valentina Caruso 和 Roberto Navigli。2021年。SPRING上线：端到端AMR解析与生成。载于*2021*年自然语言处理经验方法会议论文集：系统演示，第134–142页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。

Daniel Burrows。2005年。软件依赖关系的建模与解析。

[19] https://ai4society.ca/

Sharan Chetlur、Cliff Woolley、Philippe Vander mersch、Jonathan Cohen、John Tran、Bryan Catanzaro 和 EvanShelhamer。2014年。cuDNN：用于深度学习的高效原语。*CoRR*，abs/1410.0759。

Jaron Cohen、Roy Cohen、Edan Toledo 和 Jan Buys。2021年。RepGraph：均值表示图的可视化与分析。载于*2021*年自然语言处理经验方法会议论文集：系统演示，第79–86页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。

Russ Cox。2016年。VersionSAT。检索于2022年11月29日，网址：`https://research.swtch.com/version-sat`。

Hamish Cunningham、Diana Maynard、Kalina Bontcheva 和 Valentin Tablan。2002年。GATE：用于开发稳健人类语言技术应用的架构。载于计算语言学协会第*40*届年会论文集，第168–175页，美国宾夕法尼亚州费城。计算语言学协会。

Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。2019年。BERT：用于语言理解的深层双向变换器预训练。载于*2019*年计算语言学协会北美分会会议论文集：人类语言技术，*NAACL-HLT 2019*，美国明尼苏达州明尼阿波利斯，*2019*年*6*月*2*日至*7*日，第*1*卷（长篇和短篇论文），第4171–4186页。计算语言学协会。

Ronen Feldman 和 James Sanger。2006a。文本挖掘简介。载于《文本挖掘手册：分析非结构化数据的高级方法》，第 1–18 页。剑桥大学出版社。

Ronen Feldman 和 James Sanger。2006b。文本挖掘应用。载于《文本挖掘手册：分析非结构化数据的高级方法》，第 273–314 页。剑桥大学出版社。

David Ferrucci 和 Adam Lally。2004。UIMA：在企业研究环境中处理非结构化信息的架构方法。《自然语言工程》，10(3-4)：327–348。

Junxiao Han、Shuiguang Deng、David Lo、Chen Zhi、Jianwei Yin 和 Xin Xia。2020年。深度学习库依赖网络的实证研究。载于 *2020*年 *IEEE*软件维护与演化国际会议（*ICSME*），第 868–878 页。

Naeemul Hassan、Gensheng Zhang、Fatma Arslan、Josue Caraballo、Damian Jimenez、Siddhant Gawsane、Shohedul Hasan、Minumol Joseph、Aaditya Kulka- rni、Anil Kumar Nayak、Vikas Sable、Chengkai Li 和 Mark Tremayne。2017年。Claimbuster：首个端到端事实核查系统。《*VLDB*基金会会刊》，10(12):1945–1948。

Marti A. Hearst。1999年。理清文本数据挖掘。载于第*37*届计算语言学协会年会论文集，第3–10页，美国马里兰州大学公园市。计算语言学协会。

Mattew Honnibal、Ines Montani、Sofie Van Land- deghem 和 Adriane Boyd。2020年。spaCy：基于 Python 的工业级自然语言处理。

Kaifeng Huang、Bihuan Chen、Susheng Wu、Junmin Cao、LeiMa 和 Xin Peng。2022年。揭开深度学习技术栈中依赖关系错误的面纱。*CoRR*，abs/2207.10347。

Clayton J. Hutto 和 Eric Gilbert。2014年。VADER：一种简约的、基于规则的社交媒体文本情感分析模型。载于第八届国际博客与社交媒体会议论文集，*ICWSM 2014*，安娜堡，密歇根州，美国，*2014*年*6*月*1*日至*4*日。AAAI出版社。

Marcin Junczys-Dowmunt、Roman Grundkiewicz、Tomasz Dwojak、Hieu Hoang、Kenneth Heafield、Tom Neckermann、Frank Seide、Ulrich Germann、Alham Fikri Aji、Nikolay Bogoychev、André F. T. Martins 和 Alexandra Birch。2018年。Marian：基于 C 的快速神经机器翻译++。载于*ACL 2018*论文集，系统演示，第116–121页，澳大利亚墨尔本。计算语言学协会。

Mike Lewis、Yinhan Liu、Naman Goyal、Mar- jan Ghazvininejad、Abdelrahman Mohamed、Omer Levy、Veselin Stoyanov 和 Luke Zettlemoyer。2020年。BART：面向自然语言生成、翻译和理解的去噪序列到序列预训练。收录于第*58*届计算语言学协会年会论文集，*ACL 2020*，线上，*2020*年*7*月*5*日至*10*日，第7871–7880页。计算语言学协会。

Quentin Lhoest、Albert Villanova del Moral、Yacine Jernite、Abhishek Thakur、Patrick von Platen、Suraj Patil、Julien Chaumond、Mariama Drame、Julien Plu、Lewis Tunstall、Joe Davison、Mario Šaško、Gun- jan Chhablani、Bhavitvya Malik、Simon Brandeis、Teven Le Scao、Victor Sanh、Canwen Xu、Nicolas Patry、Angelina McMillan-Major、Philipp Schmid、Sylvain Gugger、Clément Delangue、Théo Matus- sière、Lysandre Debut、Stas Bekman、Pierric Cis- tac、Thibault Goehringer、Victor Mustar、François Lagunas、Alexander Rush 和 Thomas Wolf。2021年。Datasets：面向自然语言处理的社区库。收录于*2021*年自然语言处理实证方法会议论文集：系统演示，第175–184页，线上及多米尼加共和国蓬塔卡纳。计算语言学协会。

Christopher Manning、Mihai Surdeanu、John Bauer、JennyFinkel、Steven Bethard 和 DavidMcClosky。

2014年。 Stanford CoreNLP 自然语言处理工具包。载于第*52*届计算语言学协会年会论文集：系统演示，第55–60页，马里兰州巴尔的摩。计算语言学协会。

Dirk Merkel. 2014年 Docker：用于一致开发和部署的轻量级 Linux 容器。*Linux J.*，2014年(239)。

Filipe Mesquita、Matteo Cannaviccio、Jordan Schmidek、Paramita Mirza 和 Denilson Barbosa。2019年 Knowledgenet：用于知识库填充的基准数据集。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论文集，*EMNLP-IJCNLP 2019*，香港，中国，*2019*年*11*月*3*日至*7*日，第 749–758 页。计算语言学协会。

Sam Newman。2015年。微服务。载于 构建微服务。O'Reilly Media, Inc. John Nickolls、Ian Buck、Michael Garland 和 Kevin Skadron。2008年。使用 CUDA 进行可扩展并行编程。*ACMQueue*，6(2):40–53。Lawrence Page、Sergey Brin、Rajeev Motwani 和 Terry Winograd。1999年。PageRank 引文排名：为万维网带来秩序。技术报告 1999-66，斯坦福信息实验室。之前的编号 = SIDL-WP-1999-0120。Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga、Alban Desmaison、Andreas Köpf、Edward Z. Yang、Zachary DeVito、Martin Raison、Alykhan Tejani、Sasank Chilamkurthy、Benoit Steiner、Lu Fang、Junjie Bai 和 Soumith Chintala。2019年。PyTorch：一种命令式风格的高性能深度学习库。载于《神经信息处理系统进展》第*32*卷：*2019*年神经信息处理系统年会，*NeurIPS 2019*，*2019*年*12*月*8*日至*14*日，温哥华，不列颠哥伦比亚省，加拿大，第8024–8035页。Nils Reimers 和 Iryna Gurevych。2019年。Sentence-BERT：使用孪生 BERT 网络进行句子嵌入。载于 *2019*年自然语言处理实证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3982–3992页，中国香港。计算语言学协会。Geoffrey Rockwell 和 Stéfan Sinclair。2016年。*Hermeneutica*：人文学科中的计算机辅助阐释。麻省理工学院出版社。Wei Shen、Yuhan Li、Yinan Liu、Jiawei Han、Jianyong Wang 和 Xiaojie Yuan。2023年。实体链接与深度学习相结合：技术与解决方案。*IEEE* 知识与数据工程汇刊，35(3):2556–2578。

DennyVrande˘ci´c 和 Markus Krötzsch。2014年 Wikidata：一个免费的协作式知识库。 *Commun.ACM*，57(10)：78–85。

Zijie J. Wang、Dongjin Choi、Shenyu Xu 和 Diyi Yang。2021年 让人类参与自然语言处理流程：一项综述。载于 首届人机交互与自然语言处理衔接研讨会论文集，第 47–52 页，线上。计算语言学协会。

Thomas Wolf、Lysandre Debut、Victor Sanh、J ulien Chaumond、Clement Delangue、Anthony Moi、Pier- ric Cistac、Tim Rault、Remi Louf、Morgan Funtow- icz、Joe Davison、Sam Shleifer、Patrick von Platen、Clara Ma、Yacine Jernite、Julien Plu、Canwen Xu、Teven Le Scao、Sylvain Gugger、Mariama Drame、Quentin Lhoest 和 Alexander Rush。2020年。Transformers：最先进的自然语言处理技术。载于*2020*年自然语言处理经验方法会议论文集：系统演示，第38–45页，线上。计算语言学协会。

Yang Xiao、Jinlan Fu、Weizhe Yuan、Vijay Viswanathan、Zhoumianze Liu、Yixin Liu、Graham Neubig 和 Pengfei Liu。2022年。DataLab：一个用于数据分析和干预的平台。载于第*60*届计算语言学协会年会：系统演示论文集，第182–195页，爱尔兰都柏林。计算语言学协会。

Zexuan Zhong 和 Danqi Chen。2021年。一种令人惊讶地简单的实体和关系抽取方法。收录于计算语言学协会北美分会 *2021* 年会议论文集：人类语言技术，第 50–61 页，线上。计算语言学协会。