快速构建知识图谱：技术文本知识提取的高效标注工具 - QuickGraph

摘要

在多任务信息提取（MTIE）领域，获取高质量的标注语料库是一个既费时又昂贵的过程。尽管采用无监督技术进行自动化标注已经变得流行，但这些技术通常依赖于字典、地名集和知识库，这些资源在专业技术领域内却很稀缺。为了解决这个问题，本文介绍了QuickGraph，这是一个针对 MTIE 任务设计的首个协作式标注工具，它通过间接弱监督和聚类来最大化标注者的生产率。QuickGraph 的主要贡献是其一系列创新特性，这些特性通过快速且一致的复杂多任务实体和关系标注，实现了知识图谱的提取。本文讨论了这些关键特性，并定性比较了 QuickGraph 与现有标注工具。此外，我们还展示了系统的实际演示。对于未来的工作，我们计划提高工具的扩展性，优化注释传播过程，并考虑集成主动学习机制，以提高标注质量和效率。

关键词：知识图谱提取；技术文本；快速注释工具；QuickGraph；多任务信息提取；自动化标注技术；协作式标注工具；实体和关系注释；间接弱监督；文档聚类

1 引言

在当今信息爆炸的时代，技术文本数据的量级呈指数级增长。从这些数据中提取有价值的信息并构建知识图谱，对于深入理解和利用这些信息至关重要。知识图谱作为连接数据点的语义网络，使得数据间的关系可视化，大大促进了数据的可用性和可理解性。然而，对于专业技术领域而言，由于缺乏大规模标注语料库，以及现有工具不能有效处理专业术语和复杂关系，使得高效、准确地从技术文本中提取知识图谱变得极具挑战。

在这样的背景下，我选取了构建一个高效的知识图谱提取工具为复现课题。选题的依据在于现有的标注工具往往忽略了技术领域的特殊性，并且在标注效率和用户协作方面存在不足。而研究和开发针对技术文本的快速注释工具，可以显著提高标注效率，降低标注过程的复杂性，促进更多高质量知识图谱的生成。

本次复现的项目的意义在于填补技术领域知识图谱自动化构建的空白，特别是为机器学习和深度学习提供高质量训练数据的需求。通过 QuickGraph，作者提出了一个创新的解决方案，该工具整合了间接弱监督和聚类技术，支持复杂的多任务实体和关系标注，实现了快速、一致和高效的知识图谱提取。此外，通过促进协作式标注，QuickGraph 不仅提升了单一标注者的工作效率，也优化了团队标注工作流程，使得专家知识得以快速集成和共享。因此，该工具的开发不仅对技术领域的研究者和从业者具有重要意义，也对于推动知识图谱技术和深度学习领域的发展具有重要的战略价值。

2 相关工作

2.1 传统实体和关系注释工具

在自然语言处理领域，传统的实体和关系注释工具一直扮演着重要角色。这些工具通常专注于提供基础的注释功能，但在技术更新和用户体验方面存在一些局限。它们的主要特点包括技术老化和设置过程的复杂性。这些限制可能影响注释的效率和准确性，尤其是在处理大规模或复杂数据集时。

2.1.1 brat 工具

brat 工具是传统注释工具中的一个典型代表。它以其简洁的用户界面和广泛的使用而闻名。然而，brat 工具经常因其技术上的局限性和繁琐的设置过程而受到批评 [1]。

2.2 现代多功能注释工具

随着自然语言处理技术的快速发展，出现了一批现代化的多功能注释工具。这些工具不仅提供了传统注释功能，还集成了许多创新的特性，如知识图谱的构建、协作注释能力以及对大数据集的支持。这些工具通常具有更友好的用户界面，更加适应当前快速发展的技术需求。

2.2.1 SALKG 工具

SALKG 工具是现代多功能注释工具的一个代表，专注于知识图谱的构建和注释。尽管它在某些方面具有创新性，但也存在一些局限，如缺乏协作注释和裁决的功能。这些限制可能影响工具在大规模项目中的应用效果 [2]。

2.2.2 提高生产力的注释工具

为了提高注释效率和准确性，一些工具开始采用主动学习和预测学习方法。例如，APLenty 和 Paladin 工具就是这方面的先锋。它们通过智能算法预测注释者可能的标注决策，从而减少注释所需的时间和努力。这些工具的出现在很大程度上优化了注释流程，特别是在处理大型本体和复杂数据集时。然而，它们在多任务实体和关系注释方面的性能仍待进一步验证 [3,4]。

3 本文方法

3.1 本文方法概述

要复现论文中的工具，我需要创建一个多用户注释工具，该工具能够实时从用户的注释中构建知识图谱，并包含多个容器化组件。这个工具的主体框架如图 1所示。

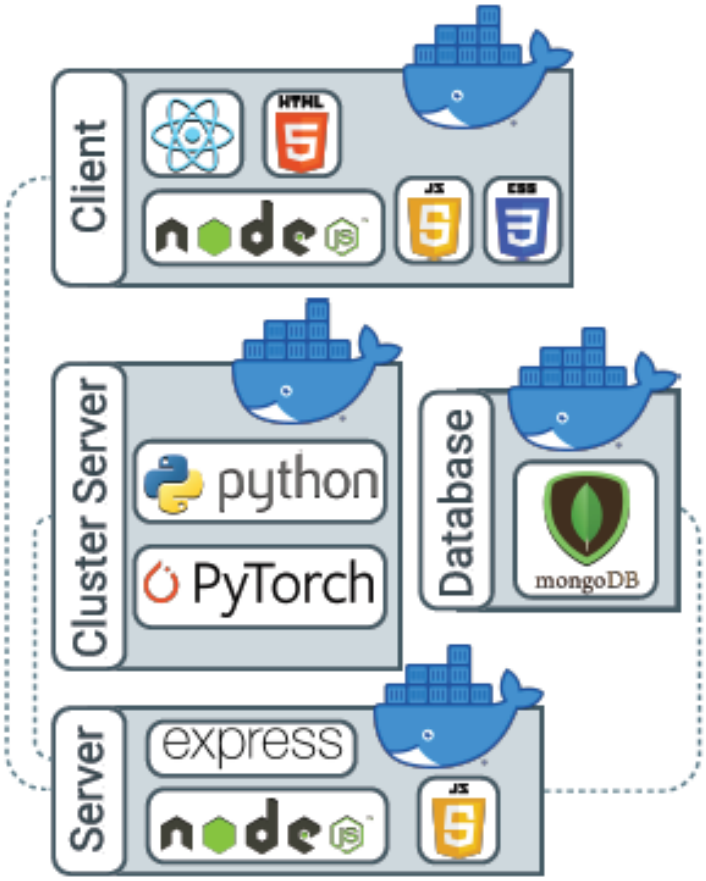

图 1. 方法示意图

3.2 Web 客户端开发

Web 客户端开发涉及创建一个直观的用户界面，使注释者能够轻松进行实体和关系的标注。这包括实现文本高亮、实体识别、关系标注等工具，以及提供清晰的视觉反馈。客户端还应允许用户轻松添加、编辑和删除注释，同时提供文档内导航的功能。

3.3 NoSQL 数据库的设置

NoSQL 数据库的设置需要您选择合适的数据库系统（如 MongoDB），设计一个能够存储注释数据、实体、关系以及用户交互信息的数据库模式。这也涉及到确保数据安全和完整性的措施，例如用户权限管理和数据备份。

3.4 服务器端开发

在服务器端开发方面，需要构建一个能够处理来自客户端的请求的后端服务器。这包括实现 API 接口，以便前端与服务器通信，以及负责数据处理逻辑，如注释的存储、检索和更新。

3.5 集群服务器的配置

集群服务器的配置是必需的，以支持大量并发用户和大规模文档集的处理，这需要实现高可用性、负载均衡和有效的文档管理系统。

3.6 实时知识图谱构建

实时知识图谱构建是另一个关键组成部分，它要求从注释中提取实体和关系信息，并实时构建知识图谱，同时提供不同类型的图谱视图（如聚合视图和分离视图）和交互功能。

3.7 系统架构的整体设计和实施

最后，系统架构的整体设计和实施需要使用 MERN 堆栈（MongoDB、Express.js、React、Node.js）以及 Python 和 Docker 来构建前后端交互、数据处理和分析，以及确保应用的可移植性和易部署性。整个架构设计应考虑可扩展性、维护性和安全性。

4 复现细节

4.1 与已有开源代码对比

复现工作及贡献声明：在本项目中，我的主要工作是复现论文《QuickGraph: A Rapid Annotation Tool for Knowledge Graph Extraction from Technical Text》中描述的工具。框架参考了 https://github.com/nlp-tlp/quickgraph 发布的代码，复现过程中代码细节则由自己编写完成。复现工作严格遵循了原始论文中描述的方法和架构，旨在深入理解并验证论文中提出的概念和技术实现。虽然本项目主要集中在复现现有工具上，但以下方面体现了我的工作和技术贡献：代码细节的补充：在整体框架的基础上自主地进行代码编写环境适应与调试：对于论文中未详细描述的部分，我进行了自主的调试和优化，确保系统在当前的软件和硬件环境中稳定运行。理解与实践的结合：通过这一复现过程，我不仅加深了对原理的理解，还获得了实际操作和问题解决的经验。总的来说，尽管本项目的核心是复现论文中的工具，但在整个过程中，我展现了对复杂系统的理解、软件开发能力以及适应和解决实际问题的能力。

4.2 实验环境搭建

要搭建 QuickGraph 实验环境，首先安装 Docker，可以从 Docker 官网下载并安装。接着，编写 QuickGraph 的源代码，包括客户端（client）、服务器（server）、MongoDB（mongo）和服务器集群（server cluster）的所有必要文件。然后，在 /server/ėnv 文件中设置环境变量，添加安全令牌到 TOKEN SECRET 字段，用于用户密码的哈希和加盐处理。打开命令行界面，导航到 QuickGraph 仓库的根目录，执行 make run 或 docker-compose -f docker-compose.yml up 来构建和启动服务。这将启动前端客户端（运行在端口 3020）、后端服务器（运行在端口 3010）、MongoDB 数据库（运行在端口 27018）和服务器集群（运行在端口 8000）。验证所有服务已正确启动后，您可以在浏览器中访问 localhost:3020 开始使用 QuickGraph。

4.3 界面分析与使用说明

在 QuickGraph 中，从项目仪表盘或页面通过点击“Annotate”按钮进入注释视图。一旦进入，可以通过点击或拖动数据集中的单词来选择它们，形成一个选中的文本跨度，选中的文本会以背景色变化表示。然后，通过点击侧边栏的实体层级或使用标签搜索模态框来给这些选中的文本分配实体标签。例如，将“Character”实体类型应用到文档中的“Alice”上。为了提高注释效率，QuickGraph 允许使用实体传播功能，即通过点击已应用到“Alice”上的实体并选择“Apply All”图标，可以将“Character”实体类型传播到语料库中所有“Alice”的实例。关于关系注释，需要切换到关系模式，选择一个源实体和目标实体来创建它们之间的关系，如“Knows”。创建这些关系后，点击保存图标，将数据集项标记为完成，从而高效地管理和标注数据集，构建结构化的知识图谱。

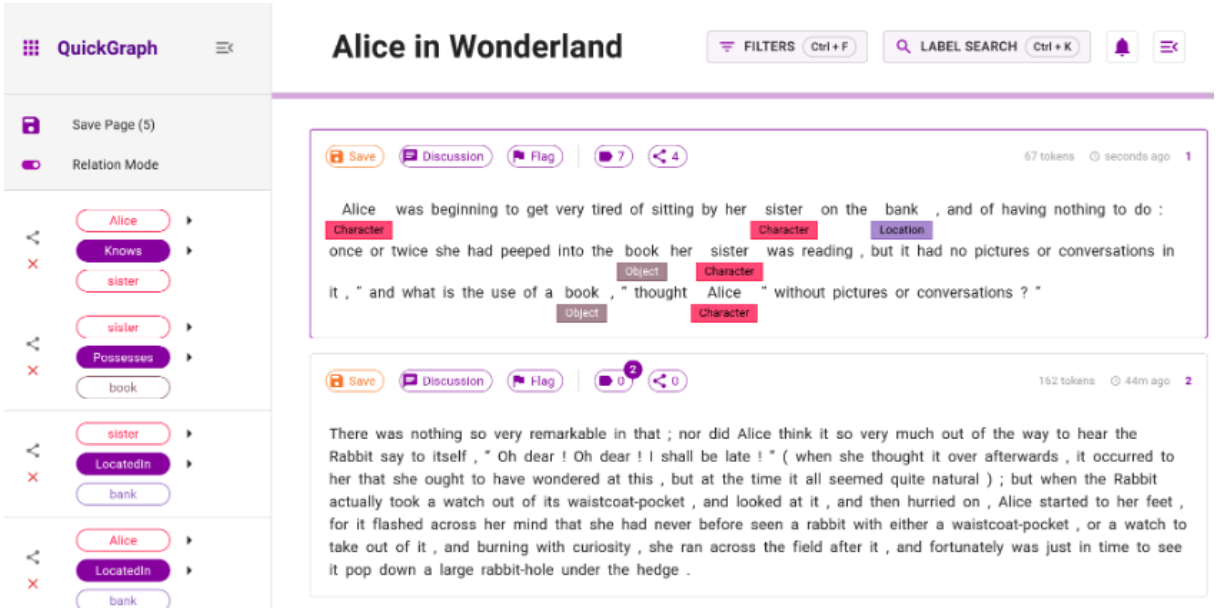

图 2. 操作界面示意

4.4 创新点

QuickGraph 引入了一种实时知识图谱构建的创新特性，它允许从文本注释直接构建出知识图谱。这种方法不需要外部资源如知识库、词典或者地名词汇表。通过这种方式，注释者可以即时看到他们工作的成果，并且实时地理解和优化他们的注释，这极大提高了注释任务的效率和质量。这个功能在处理和分析大量文档集时尤其有用，因为它支持注释者在维持高质量标准的同时快速工作 。

5 实验结果分析

成功复现项目后，我能够有效地从注释数据中提取实体之间的关系，并将这些关系以图形方式表示出来。这使我能够更清晰地了解实体之间的连接和相互作用，从而为知识发现和信息检索提供了有力的基础。通过知识图谱，我还能够发现实体之间的语义关联，例如，在医学领域中，能够发现药物与疾病之间的关联，从而为药物研发和疾病治疗提供洞察。此外，知识图谱的构建为信息检索和推荐系统提供了强大的支持，能够提供更精确的搜索结果和个

性化的推荐，从而提高用户体验和信息检索效果。另外，知识图谱的图形表示使我能够以直观的方式可视化数据，并发现其中的模式和洞察，对于决策制定和数据分析非常有价值，尤其是在大规模数据集的情况下。最后，知识图谱的构建方法具有良好的领域适应性和扩展性，可以根据不同领域和任务的需求进行扩展，以满足各种应用场景的需求。这些功能和发现为未来的研究和应用开发提供了坚实的基础。

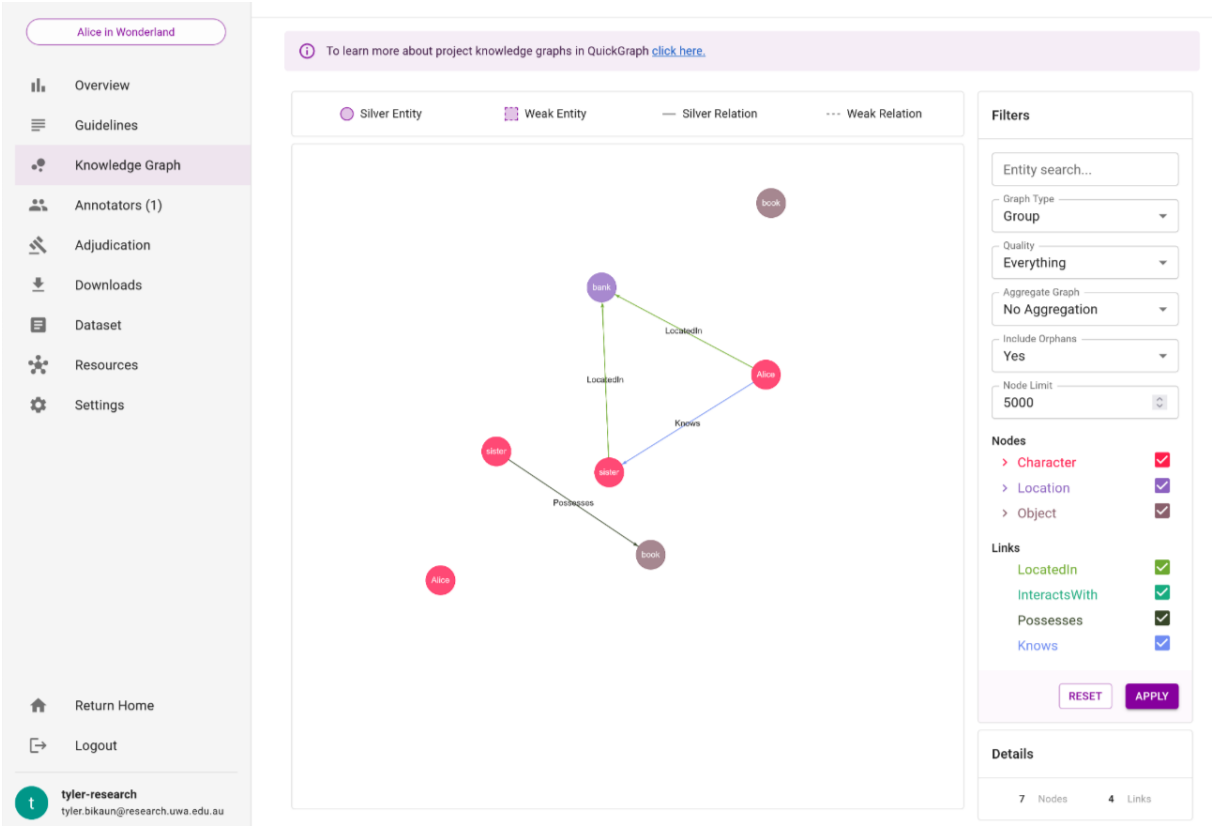

图 3. 实验结果示意

6 总结与展望

在本文中，我们介绍了一个名为 QuickGraph 的工具，该工具用于多任务信息提取（MTIE）领域的高效知识图谱提取。QuickGraph 通过实时构建知识图谱，提供了一种高效的方式来从技术文本中提取实体和关系信息。本文总结了 QuickGraph 的关键特点和创新点，并讨论了其与现有标注工具的比较。QuickGraph 是一个协作式标注工具，旨在提高专业技术领域中知识图谱提取的效率和质量。该工具引入了实时知识图谱构建的创新特性，不依赖外部资源，使注释者能够即时看到他们的工作成果。QuickGraph 还支持多任务实体和关系标注，通过实体传播功能和关系注释模式，提高了标注效率。本文通过复现工作展示了 QuickGraph 的实际应用，验证了其在知识图谱提取方面的性能和可行性。为了进一步提升 QuickGraph 工具的实用性和适用性，有几个关键方向需要着重考虑。首先，我们需要致力于提高工具的扩展性，使其能够轻松应用于更广泛的领域和任务，以满足不同应用场景的需求。其次，我们可以通过优化注释传播过程来改进实体传播和关系标注模型，从而提高标注的准确性和效率，确保从技术文本中提取的知识图谱更加精确和完整。此外，引入主动学习机制是一个关键步骤，它能够根据注释者的反馈和决策来不断改进标注质量，提高工具的自动化水平。另外，我们应该持

续优化用户界面，提升用户体验，使注释者更轻松地进行标注工作，进一步提高工具的实际可用性。最后，将 QuickGraph 应用于不同领域的实际问题，如医学、法律、金融等，有望发现新的知识和洞察，拓展工具的应用领域，使其更具广泛价值。综合考虑这些方向，QuickGraph 将能够持续发展并满足不断增长的信息提取需求和多样化的应用场景。

参考文献

[1] Pontus Stenetorp et al. Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the EACL 2012 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*. Association for Computational Linguistics, 2012.

[2] Aaron Chan et al. Salkg: Learning from knowledge graph explanations for commonsense reasoning. *arXiv preprint arXiv:2104.08793*, 2022.

[3] Minh-Quoc Nghiem and Sophia Ananiadou. Aplenty: annotation tool for creating high-quality datasets using active and proactive learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium, 2018. Association for Computational Linguistics.

[4] Minh-Quoc Nghiem, Paul Baylis, and Sophia Ananiadou. Paladin: an annotation tool based on active and proactive learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, Online, 2021. Association for Computational Linguistics.