Soft-Search：用于研究研究软件识别与生产的两个数据集

Eva Maxfield Brownevamxb@uw.edu 华盛顿大学信息学院，美国华盛顿州西雅图

Lindsey Schwartzlsschwar@uw.edu 华盛顿大学信息学院，美国华盛顿州西雅图

Richard Lewei Huang lwhuang@uw.edu 华盛顿大学信息学院，美国华盛顿州西雅图

Nicholas Weber mweber@uw.edu 华盛顿大学 信息学院 西雅图，华盛顿州，美国

arXiv:2302.14177v1 [cs.DL] 27 Feb 2023

摘要

软件是学术工作的重要工具，但研究中产生的软件在许多情况下并不容易被识别或发现。将研究与软件联系起来的一个潜在第一步是识别软件。本文介绍两个用于研究研究软件识别与生产的数据集。第一个数据集包含近 1000 条由人工标注的注释，记录了美国国家科学基金会（NSF）资助项目的软件生产情况。我们使用该数据集训练用于预测软件生产的模型。第二个数据集是将训练好的预测模型应用于 2010 至 2023 年间所有 NSF 资助项目的摘要和项目成果报告后生成的。最终得到一个推断数据集，涵盖超过 150,000 项 NSF 资助项目的软件生产情况。我们发布 Soft-Search 数据集，以帮助识别和理解研究软件生产：https://github.com/si2-urssi/eager

CCS 概念

• **信息系统** → **数字图书馆与档案馆**；聚类与分类；• **软件及其工程** → 软件创建与管理；软件库与代码仓库。

关键词

数据集、文本分类、研究软件

1 引言

软件的生产、使用与再利用日益成为学术工作中至关重要的一环 [1, 10]。尽管过去对软件引用的重视不足，但随着软件引用新标准的出台 [2, 6] 以及从现有文献中提取软件参考文献的相关工作 [5]，在研究过程中引用和参考所用软件正变得越来越普遍。然而，软件生产记录不像同行评审出版物或其他学术成果那样，能够轻易识别或大规模获取 [8]。为推动这一问题的解决，我们推出两个相关数据集，用于研究和推断作为研究工作一部分而产出的软件，并将其统称为 Soft-Search 数据集。

Soft-Search 数据集旨在识别在获得联邦资助期间可能产出了软件的研究项目。我们首先找出承认获得至少一项美国国家科学基金会（NSF）资助项目资助的 GitHub 仓库。然后，我们为找到的每个 GitHub 仓库标注其内容属于以下哪一类：软件或非软件（二者择一；例如，并非所有 GitHub 仓库都包含软件，它们也可能包含研究笔记、课程材料等）。接着，我们将每个已标注的 GitHub 仓库与其 README.md 文件中提及的特定 NSF 资助项目编号相关联。最后，我们根据每个 GitHub 仓库的标注，以及关联的 NSF 资助项目摘要和项目成果报告中的文本，编制 Soft-Search 训练数据集。

我们使用 Soft-Search 训练数据集训练多种模型，以 NSF 资助项目摘要或项目成果报告文本为输入，预测软件生产情况。随后，我们使用表现最佳的模型，对美国国家科学基金会在 2010 至 2023 年间资助的所有资助项目进行软件生产推断（更多详情见第2节）。我们汇总了 2010 至 2023 年间每项 NSF 资助项目的预测结果和元数据，构成 Soft-Search 推断数据集。

总体而言，我们新的 Soft-Search 数据集包括以下贡献：

1. Soft-Search 训练集：一个利用已关联的 NSF奖项 和 GitHub 仓库编制的真实数据集，这些数据已针对软件生产进行标注。2. 多个分类器，可分别根据 NSF 奖项摘要或项目成果报告的文本推断软件生产。3. Soft-Search 推断集：一个包含 2010 年至 2023 年间超过 150,000 个 NSF 资助项目的数据集。每个资助项目都有两项关于软件生产的预测：一项基于摘要文本进行预测，另一项基于项目成果报告文本进行预测。

本文其余部分安排如下。在第2 节中，我们详细介绍用于创建 Soft-Search 训练数据集的数据收集与标注过程。在第3 节中，我们简要描述模型训练过程并报告结果。在第4 节中，我们为 Soft-Search 推断数据集提供汇总统计，并观察软件产出随时间的变化趋势。最后，我们讨论该方法的局限性以及未来工作的机会。

2 数据收集与标注

2.1 查找由 NSF 资助项目产出的软件

我们数据收集过程的第一步，是从国家科学基金会（NSF）资助的研究中寻找软件产出。这一步有两种可能的方法。第一种方法是人工查找 NSF 资助项目摘要、项目成果报告以及各项资助所支持的论文中对软件生产的引用和承诺。这种方法劳动强度较高，并且可能容易产生标注错误，因为尽管这些文档中可能承诺会进行软件生产，但未必能够核实此类软件最终确实被生产出来。另一种方法是使用训练好的模型预测软件生产情况。我们采用这种方法，但也承认其中可能存在标签错误。

为了收集可验证软件生产的实例，我们编写了一个 Python 脚本，利用 GitHub API 搜索 README.md 文件中提及 NSF 资助项目的代码仓库。具体而言，脚本查询 README.md 文件中包含以下任一文本片段的代码仓库：“National Science Foundation”“NSF Award”“NSF Grant”“Supported by NSF”或“Supported by the NSF”。选择 GitHub 作为搜索基础，是因为它被广泛采用，并且在学术出版物中经常被提及 [3]。此次搜索发现了 1520 个独立代码仓库，其 README.md 文件中提到了 NSF。

2.2 软件生产标注

数据收集过程的下一步是将找到的每个 GitHub 仓库标注为“软件”或“非软件”。在最初审查收集到的代码仓库时，我们发现其中的内容五花八门，从文档、实验笔记、课程材料、研究项目期间编写的一次性脚本集合，到更典型的软件库（带有安装说明、测试，以及社区支持和使用）。

我们以关于何为研究软件的现有定义作为制定标注标准的基础 [7, 9], ，并在数据样本上进行了多轮试编码。我们使用 Fleiss’ kappa 检验研究团队成员对于十个 GitHub 仓库是否包含“软件”的判断是否一致。每轮试编码中，我们都会从数据集中随机选取十个 GitHub 仓库，供研究团队的每位成员独立标注。在评估代码仓库时，研究团队成员可以使用代码仓库中的任何信息来确定标注结果（例如 README.md 文件的内容、代码仓库的活动情况、文档是否可用等）。

最后一轮试编码显示，研究团队的意见几乎完全一致（K=0.892）[11]。

我们的最终标注标准总体上倾向于将代码仓库纳入软件类别，而是设定了具体的排除标准，符合这些标准的代码仓库会被标记为“非软件”。具体而言，当代码仓库的主要内容为以下内容时，会被标记为“非软件”：

1. 项目文档或研究笔记 2. 研讨会或课程的教学材料

表1：预测模型结果（使用摘要文本训练）

	模型	准确率	精确率	召回率	f1
0	TF-IDF逻辑回归	0.674	0.674	0.674	0.673
1	Transformer	0.636	0.608	0.697	0.649
2	语义逻辑回归	0.630	0.630	0.630	0.630
3	正则表达式	0.516	0.515	0.516	0.514

3. 项目或研究实验室网站的源代码 4. 专门用于分析单个实验的脚本集合，不考虑其进一步的通用性 5. 用于访问数据、但不提供任何额外处理能力的实用函数

随后，我们根据商定的标注标准，将数据集中的所有 GitHub 仓库标注为“软件”或“非软件”。

2.3 将 GitHub 仓库与 NSF奖项关联

数据收集过程的最后一步，是将已标注的 GitHub 仓库关联回具体的 NSF 资助项目。为此，我们创建了一个脚本，用于加载每个 GitHub 仓库的网页，抓取该仓库 README 的内容，并查找其中引用的具体 NSF 资助项目编号。随着数据集的标注以及该脚本的运行，数据集规模有所缩减，因为我们发现，初始搜索返回的一些仓库之所以被检索出来，是由于其他同样资助研究的非美国政府机构使用了“NSF”这一缩写。

在处理每个代码仓库时，我们的 Python 脚本会加载 README 内容，使用正则表达式搜索 NSF 资助项目编号模式，然后通过从美国国家科学基金会资助 API 请求该资助项目的元数据，验证找到的每个 NSF 资助项目编号是否有效。

随后，我们获取了每个资助项目的摘要和项目成果报告文本。这是我们数据收集流程的最后一步，使我们得以创建一个数据集，其中包含446个标注为“产出了软件”的独特 NSF奖项，以及471个标注为“未产出软件”的独特 NSF 奖项。

3 预测模型

利用汇编完成的 Soft-Search 训练数据集，我们使用资助项目摘要或项目成果报告中的文本训练了三种不同的模型。所训练的模型包括：使用词频-逆文档频率词嵌入训练的逻辑回归模型（TF-IDF逻辑回归）、使用语义嵌入训练的逻辑回归模型（语义逻辑回归），以及经过微调的 Transformer（Transformer）。语义嵌入以及我们用于微调自身 Transformer 模型的基础模型，是“distilbert-base-uncased- finetuned-sst-2-english”模型 [4]。每个模型均使用 Soft-Search 训练数据集的80%进行训练。随后，我们测试每个模型，并使用 F1 分数对各模型的性能进行排序。

表1 展示了以摘要文本为输入进行训练的结果。表现最佳的模型是 TF-IDF逻辑回归 ，其 F1 分数为 0.673。

表2：预测模型结果（使用 ProjectOut-comes 报告文本训练）

	模型	准确率	精确率	召回率	f1
0	TF-IDF逻辑回归	0.745	0.745	0.745	0.745
1	Transformer	0.673	0.638	0.771	0.698
2	语义逻辑回归	0.633	0.633	0.633	0.632
3	正则表达式	0.510	0.507	0.510	0.482

表3：NSF Soft Search数据集的构成

	项目	# 奖项	# 软件	% 软件
0	MPS	32885	19178	0.583184
1	CISE	24633	13274	0.538871
2	ENG	22900	11242	0.490917
3	GEO	17822	5142	0.288520
4	BIO	16990	6013	0.353914
5	EHR	13703	575	0.041962
6	SBE	13318	1966	0.147620
7	TIP	8597	4501	0.523555
8	OISE	2329	636	0.273079
9	OIA	498	123	0.246988

表2 展示了以项目成果报告为输入进行训练的结果。表现最佳的模型是TF-IDF逻辑回归，其 F1 分数为 0.745。

虽然使用项目成果报告训练的模型所用的数据较少，但该组表现最佳的模型，其 F1 分数高于所有使用摘要训练的模型。我们尚未对此作进一步调查，但认为这是因为项目成果报告更直接地引用了已产出的软件，而摘要只是承诺将进行软件生产。

用于训练的数据以及复现这些模型的函数，可通过我们的 Python 软件包获取：`soft-search`。

4 SOFT-SEARCH 数据集

利用预测模型，我们编制了 Soft-Search 推断数据集，其中包含 2010—2023年期间所有 NSF 资助项目的元数据、摘要文本和项目结题报告文本。Soft-Search 推断数据集还分别根据这两类文本提供了我们对软件生产情况的预测。

4.1 趋势与观察

利用 Soft-Search 推断数据集，我们可以观察软件生产随时间的变化。图1 展示了我们预测产生了软件的资助项目占比（根据资助项目的摘要进行预测）随时间的变化。尽管预测的软件生产情况每年略有波动，但我们观察到，“数学与物理科学”（MPS）资助项目中，我们预测会产出软件的项目数量最多；“计算机与信息科学和工程”（CISE）和“工程学”（ENG）紧随其后。

我们还可以观察资助项目期限延长时软件生产的趋势。图2 绘制了资助项目的百分比

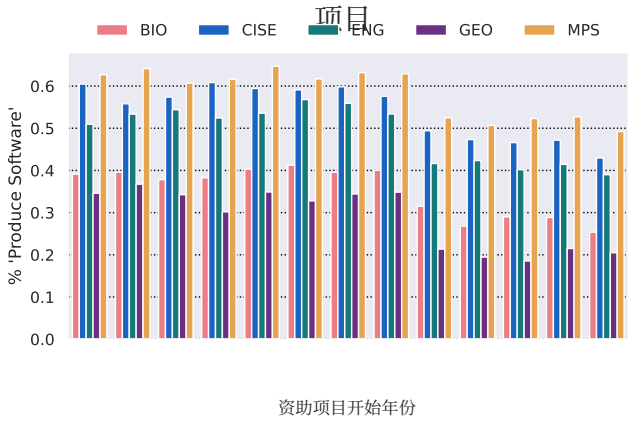

图1：软件产出随时间的变化（基于摘要预测）

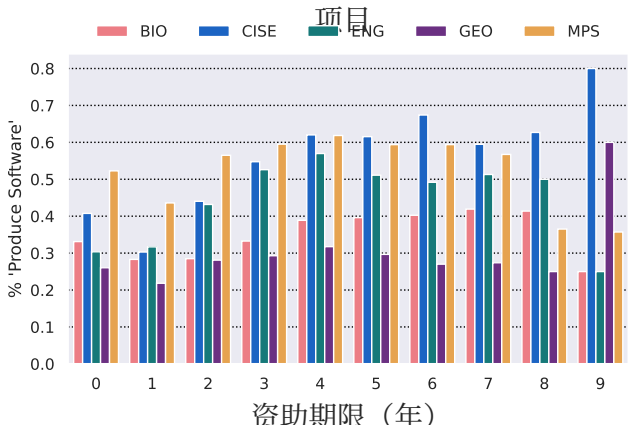

图2：按资助期限分组的软件产出（基于摘要预测）

根据资助项目的摘要，我们预测这些项目产生了软件，并按资助项目期限（年）分组。我们注意到，随着资助项目期限延长，被预测为产生了软件的资助项目所占比例也往往随之增加。

5 结论

我们介绍了 Soft-Search，这是一组用于研究 NSF 资助项目中软件生产情况的新型数据集。Soft-Search 训练数据集是一个由人工标注、包含近 1000 个示例的数据集，用于训练模型，从 NSF 资助项目摘要文本或项目成果报告文本中预测软件生产情况。我们使用这些模型生成了 Soft-Search 推断数据集。Soft-Search 推断数据集包含项目元数据、资助项目摘要和项目成果报告，以及对 2010 至 2023 年间每个 NSF 资助项目的软件生产情况预测。我们希望 Soft-Search 能推动更多研究与发现，帮助人们理解软件开发在学术出版物中发挥的作用。

本研究生成的所有数据集和预测模型均可从我们的 Git Hub 代码库获取：si2-urssi/eager。

5.1 局限性

如第 2 节所述，Soft-Search 训练数据集完全由 NSF 资助项目组成，这些项目最终在 GitHub 上发布或托管了软件（及其他研究成果）。由于我们的数据收集策略，每个预测模型学到的可能并非预测 NSF 资助项目是否会产出软件，而是预测 NSF 资助项目是否会产出托管在 GitHub 上的软件。

5.2 未来工作

如第 2.1 节所述，我们最初尝试寻找 NSF 资助项目所产出的研究软件时，采用的方法是在每个资助项目的摘要、项目成果报告及所附论文中查找有关软件生产的引用和承诺。在尝试通过这种方法创建数据集时，我们发现，许多提及计算方法的资助项目和论文并未提供指向其代码仓库或网站的链接。在某些情况下，我们通过 Google 和 GitHub 搜索，自行找到了与某个资助项目或论文相关的代码仓库。虽然我们支持在资助项目摘要、成果报告和论文中加入代码仓库链接，但未来还应研究如何实现论文及其软件产出的自动关联。

6 致谢

我们感谢 USRSSI 团队，尤其是 Karthik Ram 提供的意见。本材料基于美国国家科学基金会资助项目 2211275 支持的工作。

参考文献

[1] Prajjwal Bhattarai、Mohammed Ghassemi 和 Tuka Alhanai。2022。开源代码仓库属性预测计算机科学研究的影响。载于第 *22* 届 *ACM/IEEE* 数字图书馆联合会议论文集 （德国科隆）（*JCDL '22*）。计算机协会，纽约，纽约州，美国，文章 16, 7 10 1145 3529372 3530927 页。https://doi.org/。 /。[2] Cai Fan Du、Johanna Cohoon、Patrice Lopez 和 James Howison。2022。理解软件引用的发展：CORD-19 语料库中的软件引用研究。*PeerJ* 计算机科学 8 (2022)。[3] Emily Escamilla、Martin Klein、Talya Cooper、Vicky Rampin、Michele C. Weigle 和 Michael L. Nelson。2022。GitHub 在学术出版物中的兴起。https://doi.org/ 10 48550 2208 04895。 /ARXIV.。HF Canonical Model Maintainers。。2022。[4] distilbert-base-uncased-finetuned-sst-2-english（修订版 bfdd 146 10 57967 0181）。https://doi.org/。 /[5] hf/Ana-Maria Istrate、Donghui Li、Dario Taraborelli、Michaela Torkar、Boris Veyts-man 和 Ivana Williams。2022。生物医学文献中软件提及的大型数据集。https://doi.org/ 10 48550 2209 00693。 /ARXIV.。Daniel S. Katz、[6] Neil P. Chue Hong、Tim Clark、August Muench、Shelley Stall、Daina R. Bouquin、Matthew Cannon、Scott C. Edmunds、Telli Faez、Patricia Feeney、Martin Fenner、Michael Friedman、Gerry Grenier、Melissa Harrison、Joerg Heber、Adam Leary、Catriona J. MacCallum、Hollydawn Murray、Erika Pastrana、Kath Perry、Douglas C. Schuster、Martina Stockhause 和 Jake S. Yeston。2021。认识到软件的价值：软件引用指南。*F1000Research* 9（2021）。[7] Carlos Martinez-Ortiz、Paula Martinez Lavanchy、Laurents Sesink、Brett G. Olivier、James Meakin、Maaike de Jong 和 Maria Cruz。2022。软件管理计划实用指南。https://doi.org/ 10 5281 7185371。 /[8] zenodo。David Schindler、Felix Bensmann、Stefan Dietze 和 Frank Krüger。2022。软件在科学中的作用：基于知识图谱分析 PubMed Central 中的软件提及。*PeerJ* 计算机科学 8（2022）。[9] Vanessa Sochat、Nicholas May、Ian Cosden、Carlos Martinez-Ortiz 和 Sadie Bartholomew。2022。研究软件百科全书：定义研究软件的社区框架。《开放研究软件期刊》（2022）。[10] Ana Trisovic、Matthew K. Lau、Thomas Pasquier 和 Mercè Crosas。2021。关于研究代码质量与执行的大规模研究。*ScientificData* 9（2021）。[11] Anthony J Viera、Joanne M Garrett 等。2005。理解观察者间一致性：κ统计量。*Fam med* 37, 5（2005），360–363。