# Softcite: Automatic Extraction of Software Mentions in Research Literature

Caifan Du
*School of Information*
*University of Texas at Austin*

James Howison
*School of Information*
*University of Texas at Austin*

Patrice Lopez
*science-miner*

## 1. Motivation

The absence of a stable and widely adopted software citation system hampers the indexing, search, and acknowledgement of software contributions to science. Meanwhile, extracting relevant information about software used in research from scholarly texts could facilitate the knowledge representation of tools and procedures in research workflows. Text mining researchers have explored extracting software entities from scientific literature, but these efforts often are narrow in research domains and rely on simple rule-based approaches or bootstrapping techniques [5], [8]. Particularly, gold standard datasets of software mentions in research literature are rare and limited in scope [4], [6], [7], [3].

## 2. Progress

**Dataset**. We have constructed a gold standard dataset of software mentions in full-text research publications in biomedicine and economics, to enable data-oriented approaches for the recognition of software entities and usages in multiple research domains. Thus far, our trained annotation team has analyzed 5,553 research publications drawn from PubMed Central (PMC) Open Access Subset and Unpaywall Open Access dataset. Our first-round annotation results include 5,210 mentions of software entities along with 3,188 mentions of their attributes such as software *creator*, *version*, access *URL*, and whether the software was used in the described research. We are in preparation of releasing them as an open dataset in TEI/XML format, with software mentions presented in their original paragraph context and the corresponding article metadata. We hope this format could be immediately useful for the entity extraction research community.

**Implementation**. For validating and exploiting the Softcite dataset, we experimented with several Machine Learning approaches to sequence labeling, including linear CRF and several Deep Learning architectures, namely BiLSTM-CRF with Glove and ELmo embeddings, BERT fine-tuning using the general domain BERT-base model (bert-base-en) and BERT pre-trained on scientific texts (SciBERT), both with a CRF activation layer. We present the performance metrics of these models to demonstrate the usefulness of our gold standard dataset as a training dataset in Figure 1.

| **Labels** | **CRF** | | | **BiLSTM-CRF** | | | **BiLSTM-CRF+ELMo** | | |
|---|---|---|---|---|---|---|---|---|---|
| Metrics | Precis. | Recall | f-score | Precis. | Recall | f-score | Precis. | Recall | f-score |
| *<software>* | 86.5 | 72.24 | 78.67 | 79.70 | 75.21 | 77.37 | **86.87** | 80.72 | **83.63** |
| *<creator>* | 85.45 | 74.84 | 79.72 | 77.57 | 82.48 | 79.94 | **86.40** | **87.81** | **87.07** |
| *<version>* | **89.65** | 84.99 | 87.14 | 88.55 | **90.57** | **89.55** | 89.61 | 89.07 | 89.33 |
| *<url>* | **69.19** | 63.35 | **65.03** | 28.22 | 36.00 | 31.36 | 61.38 | **64.00** | 62.19 |
| micro-average | 82.7 | 73.85 | 77.64 | 79.62 | 78.59 | 79.09 | **86.72** | **83.14** | **84.87** |

| **Labels** | **bert-base-en+CRF** | | | **SciBERT+CRF** | | |
|---|---|---|---|---|---|---|
| Metrics | Precis. | Recall | f-score | Precis. | Recall | f-score |
| *<software>* | 75.58 | 71.64 | 73.55 | 84.85 | **82.43** | 83.62 |
| *<creator>* | 72.93 | 70.57 | 71.72 | 79.51 | 77.71 | 78.59 |
| *<version>* | 78.54 | 79.14 | 78.83 | **89.98** | 88.00 | 88.97 |
| *<url>* | 38.70 | 56.67 | 45.50 | 63.62 | **75.33** | **68.77** |
| micro-average | 74.48 | 72.67 | 73.56 | 84.42 | 82.69 | 83.54 |

Figure 1. The evaluation results for all models are obtained using 10-fold cross-validation

These different models are used in a complete pipeline for automatic extraction of software entities from research literature, implemented as a GROBID [1] submodule. GROBID is an open source Machine Learning library for extracting, parsing, and restructuring scholarly publication documents originally in raw format such as PDF. The software mention extraction[1] directly applies on the structured representation of documents produced by GROBID, with bibliographical references of the extracted mentions parsed and attached. The software entity candidates are disambiguated in context against Wikidata using entity-fishing [2] to filter out spurious non-software entities.

## 3. Applications and Future Directions

**Knowledge Base Construction**. The software mention extraction is currently applied at scale to several millions of open access research publications to populate a research software knowledge base. Extracted software mentions, possibly associated to extracted bibliographical references, are de-duplicated at entity and version level. The software entities are further matched to Wikidata entities to enable rich and interoperable linked knowledge representations. With this KB, we expect to analyze software use patterns in domain research practices, identify dependency risks in scientific software infrastructure, and develop new software-related services for researchers.

[1] Available at https://github.com/ourresearch/software-mentions

# References


[1] Grobid. https://github.com/kermitt2/grobid, 2008–2020.

[2] entity-fishing. https://github.com/kermitt2/entity-fishing, 2015–2020.

[3] Alice Allen, Peter J. Teuben, and P. Wesley Ryan. Schroedinger's Code: A Preliminary Study on Research Source Code Availability and Link Persistence in Astrophysics. *The Astrophysical Journal Supplement Series*, 236(1):10, May 2018. Publisher: American Astronomical Society.

[4] Geraint Duck, Aleksandar Kovacevic, David L. Robertson, Robert Stevens, and Goran Nenadic. Ambiguity and variability of database and software names in bioinformatics. *Journal of Biomedical Semantics*, 6(1):29, June 2015.

[5] Geraint Duck, Goran Nenadic, Michele Filannino, Andy Brass, David L. Robertson, and Robert Stevens. A Survey of Bioinformatics Database and Software Usage through Mining the Literature. *PLoS ONE*, 11(6), June 2016.

[6] James Howison and Julia Bullard. Software in the scientific literature: Problems with seeing, finding, and using software mentioned in the biology literature. *Journal of the Association for Information Science and Technology*, 67(9):2137–2155, 2016. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.23538.

[7] Udit Nangia and Daniel S. Katz. Understanding Software in Research: Initial Results from Examining Nature and a Call for Collaboration. In *2017 IEEE 13th International Conference on e-Science (e-Science)*, pages 486–487, October 2017.

[8] Xuelian Pan, Erjia Yan, Qianqian Wang, and Weina Hua. Assessing the impact of software on science: A bootstrapped learning of software entities in full-text papers. *Journal of Informetrics*, 9(4):860–871, October 2015.