A Bootstrapping Method to Assess Software Impact in Full-text Papers

Erjia Yan[1] and Xuelian Pan[2]

[1] *ey86@drexel.edu*
Drexel University, College of Computing and Informatics, 3141 Chestnut Street, Philadelphia, PA 19104 (U.S.A.)

[2] *panxuelianmail@gmail.com*
Nanjing University, School of Information Management, Nanjing, 210093 (P.R. China)

Introduction and Motivation

There is a concerted effort to study science of science in multiple spheres. However, a clear gap exists in how to incorporate digital outputs, such as software, as an integral component in scholarly communication. This tension has become aggravated in recent years because software can be the end products in many scientific inquiries. Therefore, there is the need to build a framework to assess the impact of software in science. One cornerstone in the framework is the design of text-based methods to identify software entities in full-text corpora because these entities are largely mentioned in the text rather than formally cited in the way as their publications counterpart. This research-in-progress paper will serve this purpose by the development and evaluation of a bootstrapping method to automatically extract software entities from a full-text data set.

Despite the effort of indexing digital outputs such as Thomson Reuters' Data Citation Index or SageCite by University of Bath, U.K., the use of full-text data is necessary to identify patterns of software references because these digital outputs are referenced in unsystematical ways in scientific literature. They can be embedded in documents by digital object identifiers (DOIs), hyperlinks, and featured on dedicated websites or simply be mentioned in paragraphs, footnotes, endnotes, acknowledgements, or supplementary materials. A 2014 citation study on three oceanographic data sets showed that these digital outputs are more likely to be mentioned in the text than formally cited (Belter, 2014). Intuitively, one would think of curating a list of software names; however, it will not be feasible due to the velocity, variety, and volume of software that has been developed and applied constantly. Thus, merely using metadata or static listings is incapable of capturing the full extent of the impact of software. Instead, full-text publication data provide the crucial context for this purpose.

This study will use a bootstrapping method to identify software uses in a full-text data set. It will allow us to expand the impact and attribution mechanism by assessing the impact of software.

Methods

The bootstrapping method is used to extract software entities from full-text papers. It is a self-sustaining technique used to iteratively improve a classifier's performance through seed terms (Riloff & Jones, 1999; Riloff, Wiebe, & Wilson, 2003). The bootstrapping process contains the following steps: (1) Label seed terms or learned entities in the text. Seed terms are used in the first iteration, and learned entities are used in other iterations. (2) Generate contextual patterns of seed terms in the first iteration, and create contextual patterns of learned entities in other iterations. (3) Score these contextual patterns and select top ranked N patterns as candidate patterns. (4) Score entities extracted by candidate patterns and select top ranked M entities as learned entities. (5) Go back to the first step until the system cannot learn any new positive entities.

The calculation of pattern scores and entity scores determine the effectiveness of the bootstrapping method. If a pattern gets a higher score, then it is selected into the candidate pattern pool. Entities extracted by these candidate patterns are considered as candidate entities. To boost the performance, we incorporated three heuristic rules to the calculation of pattern scores. The first feature is an unlabeled entity containing at least one uppercase letter. An entity with this feature gets a score of 1 if it contains one or more uppercase alphabetic letters; otherwise, it gets a score less than 1. The second feature focuses on version numbers. An entity with this feature gets a score of 1 if a version number is collocated. The third and fourth features deal with the presence of trigger words: a score of 1 if the left context (third feature) or right context (fourth feature) of an entity contains trigger words.

Preliminary Results

To construct a corpus that has a good balance between sentences having software entity that mentions and does not mention, we selected 427 sentences that a particular software entity is mentioned from papers published between January 6 and December 29, 2013 in the data set. 573 sentences that do not contain software entities were also included in the corpus. We use this data collection method to attain a balanced experiment set to evaluate several entity extraction methods.

Experiments that use randomly sampled sentences will be pursued as future work. We used nine frequently occurring seed terms in the proposed bootstrapping method, including SAS, SPSS, MotIV, PAML, rGADEM, Limma, PICS, PHYLIP, and Minitab. To prepare the gold standard, we manually labeled software entities in the experiment data set and in total annotated 292 unique entities. The annotations are considered as the gold standard.

Table 1 displays the experimental results of the RlogF metric entity extraction system (Thelen & Riloff, 2002), Stanford Pattern-based Information Extraction and Diagnostics (SPIED), and our software extraction system. All methods in Table 1 used the same sets of seed terms, stop word list, and common word list.

Table 1. Experimental results of software extraction.

System	*Prec*	*Recall*	*F*
RlogF	91%	7%	0.12
SPIED	40%	28%	0.33
OurSystem	80%	62%	0.70

Table 1 shows that our system performed better than RlogF and SPIED based on the F score. Although RlogF has the highest precision, it missed a great number of software entities and resulted in the lowest recall. By comparing the software entities extracted by our system and the gold standard, we found seven of the one-time occurring entities were not identified by our system thus reducing the recall. We speculate that the recall may be improved when more sentences that contain low frequently occurring software entities are added to the data set such that the bootstrapping method will be able to learn their contexts.

Table 2. Popular software use in science.

Freq	*Software entities*
2	Prism, PASW, Vienna RNAfold, survival, Stata, SeqMan, rtracklayer, R2WinBUGS, Quantity One, PyPop, Origin, Microsoft Office Excel, JMP, GeneSpring GX, genefilter, FlowJo, Effective T3, Cytoscape, COMSTAT, CellquestPro, APE, ADE4, MetaMorph Imaging System
3	SigmaPlot, WinBUGS, T3SEpre, Statistica, MetaMorph, TiMAT2, stats, Statistical Package for the Social Sciences, STADEN, limma Bioconductor
4	HyPhy, IRanges, ImageJ, Affy, Vienna RNA
5	SigmaStat, MEGA, Vegan, Geneious
≥6	R, SAS, SPSS, MotIV, Bioconductor, Weka, PAML, rGADEM, Limma, PICS, PHYLIP, Minitab, Cellquest, RNAfold, Image J, GraphPad Prism

Table 2 shows 59 popular software entities in science which occurred more than once in the test corpus based on our extraction method. Statistical software packages are well presented in Table 2; however, we also see some domain-specific open access software tools—future impact assessment may primarily focus on these.

Conclusion and Future Work

The contemporary research landscape is changing: software has increasingly been developed and applied in many data-driven projects. Therefore, there is the need to assess its impact on science and to incorporate software in scientific evaluations. This paper is part of a larger effort to build a scientific assessment framework for digital outputs that include software and data. It has proposed a bootstrapping method to extract software entities in a full-text corpus. Results show that it has successfully extracted software entities with the F score at the 0.7 level which is an improvement over the baseline methods RlogF and SPIED. Future work will involve using the whole *PLOS ONE* full-text set and introducing more advanced features to further enhance the performance of the method. Research will also benefit from integrating the number of full-text software entity mentions with citation- and usage-based metrics to complement the impact assessment of software.

Acknowledgments

Erjia Yan is supported by the National Consortium for Data Science (NCDS) Data Fellows program for the project "*Assessing the Impact of Data and Software on Science Using Hybrid Metrics*". Xuelian Pan was a visiting PhD candidate at Drexel University, supported by China Scholarship Council, when this work was performed.

References

Belter, C. W. (2014). Measuring the Value of Research Data: A Citation Analysis of Oceanographic Data Sets. *PLOS ONE, 9*(3), e92590.

Riloff, E., & Jones, R. (1999). Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. *Proceedings of* AAAI-99. Menlo Park, CA: The AAAI Press.

Riloff, E., Wiebe, J., & Wilson, T. (2003). Learning subjective nouns using extraction pattern bootstrapping. *HLT-NAACL Association for Computational Linguistics,* (pp. 25-32).

Thelen, M., & Riloff, E. (2002). A bootstrapping method for learning semantic lexicons using extraction pattern contexts. *Proceedings of the ACL-02 Association for Computational Linguistics,* (pp. 214-221).