Distant supervision for silver label generation of software mentions in social scientific publications *

Katarina Boland[1][0000−0003−2958−9712] and Frank Krüger[2][0000−0002−7925−3363]

[1] GESIS - Leibniz Institute for the Social Sciences, Cologne, Germany
[2] Institute of Communications Engineering, University of Rostock, Rostock, Germany
katarina.boland@gesis.org, frank.krueger@uni-rostock.de

Abstract. Many scientific investigations rely on software for a range of different tasks including statistical data analyses, data pre-processing and data presentation. The choice of software may have a great influence not only on the research process but also on the derived findings, e.g. when errors in the used software lead to incorrect computations or biases. In order to increase transparency of research and verifiability of findings, knowledge of the used software thus is crucial. However, explicit links between publications and used software are usually not available. In addition, software is, unlike literature, often not cited in a standardized way which makes the automatic generation of links difficult. While recent Named Entity Recognition (NER) approaches yield excellent results for a wide range of use-cases and tasks, they typically require large sets of annotated data which may be hard to acquire. In this paper, we investigate the use of weakly supervised approaches with distant supervision to create silver labels to train supervised software mention extraction methods using transfer learning. We show that by combining even only a small number of weakly supervised approaches, a silver standard corpus can be created that serves as a useful basis for transfer learning.

Keywords: Software Mention Extraction · Silverstandard · Mining of Scientific Publications · Open Science · Distant Supervision

1 Introduction

Today, software is used for a variety of tasks in all steps of the research process, e.g. from data collection and data analysis to presentation and dissemination of findings. Therefore, it can shape both the process and the outcomes of scientific investigations in significant ways. Eklund et al., for instance, discovered inflated false-positive rates during analysis of FMRI data when using standard FMRI analysis software packages [8]. Therefore, research findings relying on analyses

* This work was partially carried out at GESIS - Leibniz Institute for the Social Sciences and was financially supported by the GESIS research grant GG-2019-015 and the German Research Foundation (DFG) within the CRC 1270.

using these packages may be systematically flawed. Another problem that was recently identified concerns the automatic formatting of dates in Excel which is shown to mistakenly convert gene names [25] which may introduce errors into datasets. Provenance information including knowledge about the software that is involved in scientific investigations thus is crucial to create understandable, traceable, and reproducible research that meets the requirements of open science and enables the implementation of recently proposed mechanisms for quality control and reproducibility [12]. Links between software, created datasets, and research findings would enable explicit modelling of provenance information and tracing of biases and errors throughout all stages of the research process. Also, assessing the usage of software in scientific publications could serve as a basis for rewarding software as research output [20] further advancing open science.

However, such links are not easily identifiable. While software citation standards exist (e.g., by FORCE11 [23]), none of them has yet become universally established in scientific publications. Some researchers include only the name of the software, others use the name including information about the manufacturer and the version. This complicates automated extraction of such statements and thereby the automatic detection of links. Manual analyses of software mentions were done previously [12,19], but were limited to reduced sets of publications (90 and 40, respectively) due to the high costs of manual annotations.

Recently, deep neural networks have gained increasing interest in the domain of NER [13,2], which software mention identification can be seen as. The application of neural models for NER provides outstanding recognition results but requires a large training corpus with labelled entities. The provision of such labelled data is often the bottleneck when it comes to neural NER, as it is typically done in a manual process by different annotators. Different approaches have been proposed to overcome this issue, as for instance semi-supervised learning [26] and distant supervision [6]. Another approach is the usage of a so called silver standard corpus (SSC) [21], which in contrast to a gold standard corpus (GSC) is created by automatic labelling by a combination of different classifiers. The quality of SSCs is much lower than the quality of GSCs. However, recent work showed that neural NER can be improved by transfer learning, where the network is first trained on the SSC and later on a GSC [10]. This reduces the necessary size of the GSC, while at the same time increasing the recognition performance of the classifier.

The objective of this case-study is to investigate whether and how weakly supervised classifiers can be employed to create a SSC for the extraction of software mentions from scientific publications which can later be used for transfer learning. We apply three weakly supervised classifiers on a small manually created GSC in order to create *silver labels* which are then used for training a supervised classifier in order to predict the *gold labels* of the GSC.

The remainder of this paper is structured as follows. We first provide an overview of current approaches to NER in general and software mention identification in particular in Section 2. The applied weakly supervised classifiers for named entity extraction are described in Section 3, our method for combining

them in Section 4. The GSC is introduced in Section 5. Section 6 presents the evaluation and discussion of results before we conclude with Section 7.

2 Related work

Recent approaches to extracting software mentions from scientific publications can be divided into three groups: manual extraction, rule-based, and supervised machine learning-based approaches. Manual approaches, sometimes called content analysis, typically work on small corpora with less than 100 articles or focus on particular software. Li et al. analysed the usage of the statistical software R [16] and LAMMPS [14] in 400 articles, while Nangia and Katz [19], and Howison [12] concentrated on software in general in 40 and 90 articles, respectively. An automatic approach to software mention identification is implemented in the BioNerDS system by Duck et al. [7], who used a rule-based system based on syntactic token features and a dictionary of known software names. In later work, they employed a post-processing based on supervised machine learning which resulted in recognition rates of .67 F1. Another rule-based system to automatically identify mentions of R was implemented by Li and Yan [15]. Due to the particular focus on R and a dictionary of R packages, they were able to reach recognition rates of .94 F1. Pan et al. [20] introduced an iterative bootstrapping approach to software mention identification which achieves .58 F1. Additionally, approaches exist that analyse references to software and code based on the URL to repositories [1,22] However, there are currently no other supervised approaches for the identification of usage statements for software in scientific publications. One reason might be the lack of a dataset of sufficient size and quality.

For the related and similar task of extracting dataset references from scientific literature, we again find both semi-supervised and rule-based systems as well as supervised approaches. Boland et al. [4] employed a pattern-based iterative bootstrapping algorithm, named InfoLink, and were able to reach a precision of up to 1 with a very low recall of .3 on the downside. Another semi-supervised approach is introduced by Ghavimi et al. [9], which used a dictionary of dataset names and employed similarity scores for identification with a recognition rate of .85 F1. Lu et al. used supervised learning to identify datasets by use of a training set of 1,000 sections that were obtained by active learning and achieved a precision of .82 and a recall of .59 [17].

NER on scientific texts has been used with other targets in the literature with a particular interest in biomedical publications, as for instance for the identification of drugs, genes, proteins, and diseases [5]. This was also fostered by the BioCreative challenges that addressed gene names or chemicals and drug names. In this vein, Luo at al. employed neural models in order to identify chemical names from scientific texts with .91 F1 on a labelled corpus of 10,000 (training set: 3,500) abstracts with 84,355 labelled entities [18]. In detail, they used a BiLSTM-CRF including an additional attention layer with character-, word-, and dictionary embeddings and other linguistic features. Chemical names are different to software names when it comes to lexical structure as they typi-

cally exhibit combinations of characters, numbers, and special characters while software names are often composed from words from the lexicon. Beside the domain of scientific publications, neural NER methods have reached superior [13,2] recognition rates but often require large training sets with several thousands of labelled entities. In addition, often target entities with high occurrences are chosen, which make even small training sets more effective. As shown in Section 5, software mention statements, particularly in the social sciences, are very rare, which requires even larger training sets. One way to overcome this problem is the use of distant supervision to create large annotated corpora which enable the training of sophisticated methods for even very fine-grained entity typing tasks [6]. A different approach to overcome the lack of large training datasets is the application of transfer learning, which allows to transfer trained concepts, e.g. between different application domains or languages. Giorgi and Bader recently illustrated the benefit of transfer learning in biomedical NER [10] on datasets with a small number of labels, increasing the recognition rate substantially. They transferred a neural model for NER from a noisy SSC to a GSC, which lead to significant increases in the recognition rates.

To summarize, semi-supervised approaches can achieve high precision but suffer from low recall. Supervised approaches produce more reliable results but require large sets of labelled training data. The application of distant supervision and transfer learning allows the automatic creation of labelled datasets and exploiting them for pre-training of more high-performance supervised methods.

3 Weakly supervised Named Entity Extraction

To overcome the data acquisition bottleneck for labelled corpora, we choose a small selection of openly available named entity extraction tools for the creation of silver labels. As described in the related work section, weakly supervised tools naturally suffer from relatively low recall. However, since they implement different algorithms and use different features, we expect the different tools to produce diverging annotations, potentially complementing each other when combined.

3.1 BioNerds

Bioinformatics Named Entity Recogniser for Databases and Software (BioNerds) [7] is a rule-based system for recognition of software and databases from scientific publications in the domain of bioinformatics. Beside hard coded rules, it employs a dictionary of software and database names collected from Wikipedia, Bioconductor, and other sources. BioNerds implements a scoring system where the sum of the scores of the different features is used to decide upon the type of the entity, when a particular threshold is exceeded. The highest scores are provided by the dictionary matches, but also matches of Hearst patterns or positive head nouns achieve positive scores. Furthermore, the occurrence of a URL, a reference or a version number is considered as positive hint. Negative scores are provided, for instance, for matches with the English dictionary, negative head

nouns or partial word matches. The threshold to be exceeded in order to be classified positively was selected to be slightly below the score of a match with the dictionary of known entities. As a result, known entities are, given a positive context, almost certainly recognised.

3.2 InfoLink

InfoLink [4] is a weakly supervised iterative pattern-based bootstrapping approach developed for extracting dataset references from (social) scientific publications. Initially, seed words are searched in the corpus to identify patterns from their surrounding contexts. By alternating application of pattern identification and entity extraction, the dictionary of entities is increased iteratively. InfoLink relies on the surface form, i.e. the surrounding words of seed mentions, with some heuristics to normalize years and numbers and a frequency-based pattern scoring mechanism. Patterns consist of regular expressions and Lucene queries for increased efficiency.

3.3 Spied

The Stanford Pattern-based Information Extraction and Diagnostics [11] (SPIED) system also implements a semi-supervised approach to named entity recognition. In the main, it operates similarly to InfoLink but includes different and more complex scoring mechanisms and features such as edit distance-based features, distributional similarity, and TF-IDF weighting, the patterns include POS rather than relying solely on surface strings.

4 Method

We first apply each weakly supervised tool separately on the corpus to retrieve a list of patterns and terms classified as software mentions. We create one BIO[3] file for each tool and corpus. For this purpose, we search all retrieved terms in the input texts and treat each occurrence as a software mention. For InfoLink, we receive, in addition to the list of terms, as output a list of regular expression patterns that can easily be applied on the input texts without requiring additional pre-processing. We create a second BIO file for InfoLink searching the patterns in the input texts. Since this has the potential to disambiguate software mentions from homonymous other entities, we use these predictions in our combined classifier but keep the term search variant for comparison. Weakly supervised approaches depend to a large part on the usefulness of their given seeds. Since our aim is to generate a silver standard for conditions where no or little training data is available, we do not use knowledge on the distribution of software

[3] The BIO format is a common format for annotated texts in named entity recognition. For each token, either a **B**egin, **I**n, or **O**utside tag is provided signalling whether the token belongs to an entity of interest (as its first token (**B**) or a subsequent one (**I**) or whether it is not part of any entity to annotate (**O**)).

Table 1. Features used for the CRFs.

dependency tag, fine-grained POS tag, coarse POS tag, surface form, lemma, is_alpha, is_stop, shape, sentence length, sentence number, word number

mentions in the training data to construct a seed set. Instead, we use Wikidata for distant supervision. Since we are mainly interested in finding software that is used for processing and analysing data for social scientific publications to gain provenance information on generated data and findings, we query Wikidata for all instances belonging to the classes "statistical package" or "mathematical software". Note that while it is also possible to use an extensive list of all known software names, this would introduce more noise due to the fact that software names often consist of common nouns (see Section 2) while at the same time providing little extra information relevant to our use-case. We instead rely on the weakly supervised approaches for expanding the list of software names. We incorporate all language variants and alternative names listed in Wikidata. This results in a list of 47 software names of which 10 and 8 are mentioned in the training and test set at least once, respectively. In the second step, we combine the predictions of all tools and use their majority votes as silver labels.

As supervised approach, we model the extraction of software mentions as a sequence labelling task using Conditional Random Fields (CRFs). The CRF is trained on the silver labels and may use the tools' individual predictions and additional output as features. Additional output are confidence values for InfoLink and BioNerds as well as information on the employed rules for BioNerds. Adding to that, we permit the CRF to use a small number of simple features as additional cues. These are listed in Table 1.

The threshold for accepting or rejecting patterns has to be set manually for InfoLink. Since we do not want to rely on annotated data to do parameter tuning, we use the configuration which was optimal for the extraction of dataset references [4].

5 Dataset and Preprocessing

In order to measure the quality of our approach, we created a GSC of articles from the social sciences from PLoS[4]. Out of all articles having the keyword "*Social sciences*", we randomly selected 200. Following [7], we automatically extracted all "Methods and Materials" sections as software mentions are expected to primarily occur here. 8 articles were removed from the set as they did not contain a "Methods and Materials" section. The resulting texts were annotated with the brat annotation software [24] by six annotators that were instructed to annotate software names without mentions of additional information such as producers or versions. For about 10% of the sentences which were randomly selected from the sentences of all annotators, a second annotation was obtained in order

[4] https://www.plos.org/

Table 2. Number of articles with the given numbers of software mentions.

# software	0	1	2	3	4	5	6	>6	sum
# articles	45	46	43	21	12	7	6	12	192
% articles	23.4	24.0	22.4	10.9	6.3	3.6	3.1	6.3	100

Table 3. The 10 most common software mentions and their numbers of occurrences overall and in the training and test set, respectively. The ✓ signals whether the wikidata seeds contain the software name.

software	*SPSS*	*MATLAB*	*SAS*	*Stata*	*R*	*Prism*	*SPM8*	*Matlab*	*PLINK*	*MEGA*
# overall	17	16	15	13	12	12	11	10	9	9
# train	12	13	9	9	10	11	5	5	9	9
# test	5	3	6	4	2	1	6	5	-	-
wikidata	✓	✓	✓	✓	✓	-	-	✓	-	-

to assess the quality of the annotation. The inter-rater agreement was computed using Cohen's κ and reached *almost perfect* agreement of κ=.82. Overall, 462 (263 unique) software mention statements were found across all articles during annotation of the articles. Their distribution is detailed in Table 2. The number of articles that contained no software mentions at all was 45 (23 %). Table 3 lists the 10 most common software names including their frequencies in the training and test set. Note that software may be listed multiple times but with different spellings, e.g. for "Matlab" and "SPSS". Since our aim at this point is the identification of software mention statements rather than their disambiguation and linking, we do not align these different variants. Table 4 lists the number of unique software mentions that occurred at least n times in the corpus.

The annotated corpus was split into sentences using the Stanford NLTK Sentence Splitter [3], resulting in 12,480 sentences. Afterwards, a white space based tokenisation was done, resulting in 347,544 tokens. The annotated token sequence was finally represented as BIO sequence. 462 of these tokens were annotated with the *begin* and 120 with the *in* tag, the remaining with the *outside* tag. From this corpus, we created a training and test set with 75 and 25 percent of articles, respectively.

Table 4. Number of different software mentions occurring with the respective frequencies.

# occurrence	≥ 13	≥ 12	≥ 11	≥ 10	≥ 9	≥ 6	≥ 5	≥ 4	≥ 3	≥ 2	≥ 1
# software	1	2	3	4	8	9	11	14	21	52	215

6 Evaluation

6.1 Metrics

To measure the performance of the software mention detection task, we distinguish between exact and partial matches and compute precision, recall and F-measure considering each of these. Here, exact match means that the entire name of the software was recognised with the correct range, while partial matches signal that a certain overlap between the label and the prediction exists. We used the SemEval 2013 evaluation script[5] by David Batista.

6.2 Experimental setup

We measure the performance of the weakly supervised approaches, individually and in combination, as well as the direct distant supervision using labels from Wikidata and assess their applicability for transfer learning by using the silver labels as ground truth for training a CRF (Silver CRF). For combining the predictions of the weakly supervised approaches and creating silver labels, we test three different methods:

1. *majority*: majority vote of predicted labels
2. *conservative*: tokens are only labelled as belonging to a software mention if all classifiers agree on it belonging to this category
3. *greedy*: tokens are labelled as belonging to a software mention when at least one classifier labels it as such

The conservative and greedy conditions are expected to max out precision and recall, respectively. As an upper bound, we train a CRF on the gold labels of our GSC (Gold CRF). To evaluate the robustness of the approach with respect to seed selection and give insights on the usefulness of the Wikidata seeds, we illustrate the effects of choosing different seed sets for the weakly supervised approaches. For this, we create bins for software mentions depending on their number of occurrences in the training set. The intuition behind that is that the most frequently mentioned software names will also be the most well-known which can be identified without requiring the consultation of external knowledge sources. The less frequent a mention is, the less likely it will be incorporated into a seed set when the occurrence of software mentions in the corpus is not known in advance which is typically the case. Finally, we test the effects of using silver labels and outputs of the weakly supervised approaches as additional features for the gold CRF. For the weakly supervised approaches and the direct labelling of software mentions using Wikidata supervision, we evaluate both the performance on the training and the test set. The CRFs are trained on the training and evaluated on the test set.

[5] The original script can be obtained from https://github.com/davidsbatista/NER-Evaluation/blob/7de8a231d5fd94ced0ef10c42971a30cd3b744b3/ner_evaluation/ner_eval.py. (We adjusted the calculation of the overlapping range by an offset of 1 and added calculation of F1 scores.)

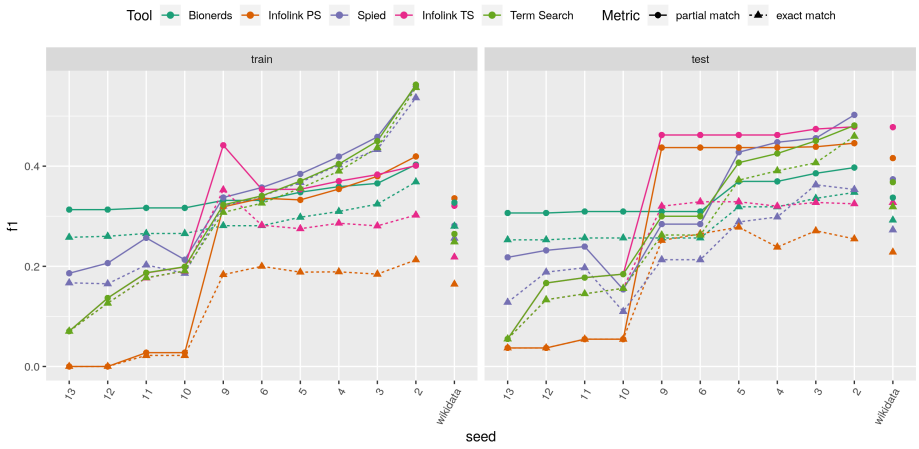

Fig. 1. F-scores of the weakly supervised tools with distant supervision using different seed sets.

6.3 Results

The performance of the weakly supervised tools with distant supervision and the influence of the choice of seeds is illustrated in Figure 1. The X axis represents the different seed sets used; 13 describes the set containing all software mentions occurring at least 13 times in the training set (the maximum number), 12 all mentions with at least 12 mentions and so forth (see Table 4). Wikidata represents the seed set obtained by querying Wikidata. As expected, performance generally increases when seeds are added. Especially when the number of seeds exceeds a certain threshold (4 and 9 in this case when only seeds occurring at least 10 or 6 times are used, respectively), there is a significant increase in performance. At the same time, adding seeds can harm performance for the pattern induction approaches as ambiguous and rare mentions may increase the likelihood of generating deficient extraction patterns. The seed set obtained from Wikidata leads to performance which is close to the optimal seed set which shows that distant supervision using lists of well-known software for seeding the algorithms is a feasible approach. The numbers for term search show the impact of the used seeds for comparison. When near-complete information on mentioned software is available, there is no or little gain from applying weakly supervised approaches in addition to searching the known names directly. However, even then precision may suffer from ambiguous names that may refer to software or other entities, such as with the software package “R” which has a high influence when used in a set with only 3 other less ambiguous seeds (set of seeds >=10 mentions). The performance of the two pattern generating approaches (SPIED and InfoLink) on the training set is considerably worse than on the test set. An analysis of the induced patterns reveals that this is due to the higher number of ambiguous software names in the former, more precisely, the high number of occurrences of

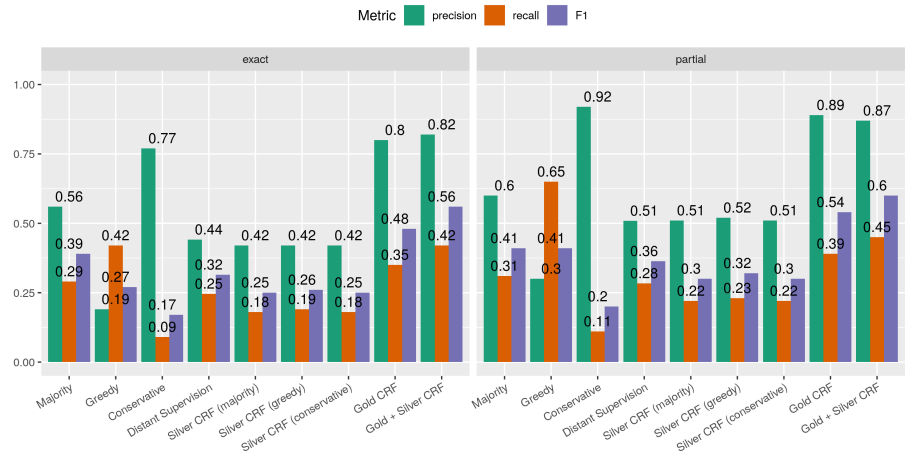

Fig. 2. Comparison of the performances of the different classifiers on the test set using the Wikidata software names as seeds / for distant supervision.

the software "R" which causes the generation of deficient patterns. For InfoLink, the pattern search variant succeeds in disambiguating software mentions from homonyms not referring to software as reflected by its higher precision compared to the term search variant. However, many software mentions are missed reducing recall considerably. InfoLink yields the best results for partial matches on both the training and test set. Yet, it also has the highest divergence in scores for exact vs. partial matches reflecting its strength in detecting mentions but its weakness in determining the exact boundaries of the matches. This is caused by its relying on surface features rather than incorporating knowledge gained from linguistic features such as POS tags.

The results for the combination of the different tools and their usage for silver standard generation are illustrated in Figure 2. The upper bound for the classifier (Gold CRF) reaches .54 F1 on the test set. The majority vote silver labels obtain .41 F1 with the recall being closer to the upper bound than the precision. The greedy variant achieves the same F-score but is biased towards maximizing recall at the cost of precision yielding higher recall values than the Gold CRF. The conservative variant suffers from low recall causing its F-score to be low (.2) while achieving a higher precision than the Gold CRF. The combination of the weakly supervised approaches with distant supervision outperforms the direct creation of silver labels from the Wikidata software names and the application of the approaches individually. The Silver CRFs achieve lower scores than the direct application of the weakly supervised approaches on the test set. We attribute this to the higher difficulty of the training set which results in decreased performance for the pattern induction approaches. These noisy labels are used for training the classifier which is then applied on the test set while the weakly supervised

approaches are applied on the easier test set directly. Finally, the best result is achieved by feeding the silver labels as additional features to the Gold CRF. While this has a slightly negative impact on precision, it increases recall by a higher magnitude resulting in .6 F1 with a still very high precision of 0.87.

7 Conclusion and Outlook

We investigated the use of weakly supervised classifiers and Wikidata for distant supervision for the extraction of software mentions from social scientific publications without requiring manual annotations. We compared the generation of silver labels by directly labelling mentions according to the Wikidata information to using them as seeds for different information extraction tools. We can show that in doing so, a silver standard with relatively high-precision annotations can be created that may serve to pre-train more powerful algorithms using transfer learning. With each classifier using different features and scoring mechanisms, their combination yields the best results showing that they partly complement each other. Furthermore, we show that predictions of weakly supervised classifiers may provide useful features for supervised methods which leads to good results even when using on a small training set. In this case-study, we employed a small set of basic features for the supervised approaches to demonstrate the feasibility of the approach. In future work, we will use more sophisticated features and supervised classifiers with transfer learning to exploit the generated SSC and extract software mentions from larger collections.

References

1. Allen, A., Teuben, P.J., Ryan, P.W.: 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). https://doi.org/10.3847/1538-4365/aab764
2. Beltagy, I., Cohan, A., Lo, K.: Scibert: Pretrained contextualized embeddings for scientific text (2019)
3. Bird, S., Loper, E., Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc (2009)
4. Boland, K., Ritze, D., Eckert, K., Mathiak, B.: Identifying references to datasets in publications. In: International Conference on Theory and Practice of Digital Libraries. pp. 150–161. Springer (2012)
5. Campos, D., Matos, S., Oliveira, J.L.: Biomedical named entity recognition: a survey of machine-learning tools. In: Theory and Applications for Advanced Text Mining. IntechOpen (2012)
6. Choi, E., Levy, O., Choi, Y., Zettlemoyer, L.: Ultra-fine entity typing. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 87–96. Association for Computational Linguistics (2018), http://aclweb.org/anthology/P18-1009
7. Duck, G., Kovacevic, A., Robertson, D.L., Stevens, R., Nenadic, G.: Ambiguity and variability of database and software names in bioinformatics. Journal of biomedical semantics **6**(1), 29 (2015)

8. Eklund, A., Nichols, T.E., Knutsson, H.: Cluster failure: why fmri inferences for spatial extent have inflated false-positive rates. Proceedings of the National Academy of Sciences p. 201602413 (2016)
9. Ghavimi, B., Mayr, P., Lange, C., Vahdati, S., Auer, S.: A semi-automatic approach for detecting dataset references in social science texts. Information Services & Use **36**(3-4), 171–187 (2016)
10. Giorgi, J.M., Bader, G.: Transfer learning for biomedical named entity recognition with neural networks. (feb 2018). https://doi.org/10.1101/262790
11. Gupta, S., Manning, C.D.: Improved pattern learning for bootstrapped entity extraction. In: Computational Natural Language Learning (CoNLL) (2014)
12. Howison, J., Bullard, J.: 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)
13. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics (2016). https://doi.org/10.18653/v1/n16-1030
14. Li, K., Lin, X., Greenberg, J.: Software citation, reuse and metadata considerations: An exploratory study examining lammps. Proceedings of the Association for Information Science and Technology **53**(1), 1–10 (2016)
15. Li, K., Yan, E.: Co-mention network of r packages: Scientific impact and clustering structure. Journal of Informetrics **12**(1), 87–100 (2018)
16. Li, K., Yan, E., Feng, Y.: How is r cited in research outputs? structure, impacts, and citation standard. Journal of Informetrics **11**(4), 989–1002 (2017)
17. Lu, M., Bangalore, S., Cormode, G., Hadjieleftheriou, M., Srivastava, D.: A dataset search engine for the research document corpus. In: Data Engineering (ICDE), 2012 IEEE 28th International Conference on. pp. 1237–1240. IEEE (2012)
18. Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., Wang, J.: An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. Bioinformatics **34**(8), 1381–1388 (nov 2017). https://doi.org/10.1093/bioinformatics/btx761
19. Nangia, U., Katz, D.S.: 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). pp. 486–487 (Oct 2017). https://doi.org/10.1109/eScience.2017.78
20. Pan, X., Yan, E., Wang, Q., Hua, W.: Assessing the impact of software on science: A bootstrapped learning of software entities in full-text papers. Journal of Informetrics **9**(4), 860–871 (2015)
21. Rebholz-Schuhmann, D., Yepes, A.J.J., Mulligen, E.M.V., Kang, N., Kors, J., Milward, D., Corbett, P., Buyko, E., Beisswanger, E., Hahn, U.: CALBC Silver Standard Corpus. Journal of Bioinformatics and Computational Biology **08**(01), 163–179 (feb 2010). https://doi.org/10.1142/s0219720010004562
22. Russell, P.H., Johnson, R.L., Ananthan, S., Harnke, B., Carlson, N.E.: A large-scale analysis of bioinformatics code on GitHub. PLOS ONE **13**(10), e0205898 (oct 2018). https://doi.org/10.1371/journal.pone.0205898
23. Smith, A.M., Katz, D.S., and, K.E.N.: Software citation principles. PeerJ Computer Science **2**, e86 (sep 2016). https://doi.org/10.7717/peerj-cs.86
24. Stenetorp, P., Pyysalo, S., Topic, G., Ohta, T., Ananiadou, S., Tsujii, J.: brat: a web-based tool for nlp-assisted text annotation. In: EACL 2012, 13th Conference of

the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012. pp. 102–107 (2012), http://aclweb.org/anthology/E/E12/E12-2021.pdf
25. Zeeberg, B.R., Riss, J., Kane, D.W., Bussey, K.J., Uchio, E., Linehan, W.M., Barrett, J.C., Weinstein, J.N.: Mistaken identifiers: Gene name errors can be introduced inadvertently when using excel in bioinformatics. BMC Bioinformatics **5**(1), 80 (2004). https://doi.org/10.1186/1471-2105-5-80
26. Zhou, Z.H.: A brief introduction to weakly supervised learning. National Science Review **5**(1), 44–53 (2017)