SOMD2025: A Challenging Shared Task for Software Related Information Extraction

Sharmila Upadhyaya[1] **Wolfgang Otto**[1] **Frank Krüger**[2] **Stefan Dietze**[1,3]
[1]GESIS — Leibniz Institute for the Social Sciences, Cologne, Germany
[2]Wismar University of Applied Sciences, Wismar, Germany
[3]Heinrich-Heine-University Düsseldorf, Germany
{sharmila.upadhyaya, wolfgang.otto, stefan.dietze}@gesis.org
frank.krueger@hs-wismar.de

Abstract

The use of software in acquiring, analyzing, and interpreting research data underscores its role as an essential artifact of scientific inquiry. Understanding and tracing the provenance of software in research helps in reproducible and collaborative research works. In this paper, we present an overview of our second iteration of the **So**ftware **M**ention **D**etection (SOMD) shared task as a part of the Scholarly Document Processing (SDP) workshop, that will be held in conjunction with ACL in 2025. The objective of this shared task is to encourage participants to reevaluate the methodologies employed in the tasks of joint named entity recognition (NER) and relation extraction (RE) for software mentions using the gold standard benchmark that has been provided. Our shared task has two phases of challenges. First, the participants focus on implementing a joint framework for NER and RE for the given dataset. Furthermore, the second phase encompasses an out-of-distribution dataset, which is utilized to assess the generalizability of the methodologies proposed in Phase I. The competition, which transpired from March to April of 2025, garnered the participation of 18 individuals and spanned a duration of two months. Four teams have finished the competition and submitted full system descriptions. Participants applied various approaches, including joint and pipeline models, and explored data augmentation with LLM-generated samples. The evaluation was based on a macro F1 score for both NER and RE, with the average reported as the SOMD score. The winning teams achieved a SOMD score of 0.89 in Phase I and 0.63 in Phase II, demonstrating the challenge of generalization.

1 Introduction

Scientific research is becoming progressively data-centric, and software plays an important role across disciplines by enabling the analysis, processing, and modeling of research data. As such, it has emerged as a key scholarly artifact, essential not only for conducting research but also for ensuring the reproducibility and advancement of scientific knowledge. To ensure transparency and reproducibility of scientific work, it is essential to identify the software used and trace its provenance, thus encouraging collaboration among scientists/researchers. Software mentions in scholarly publications are heterogeneous, informal, and in widespread use. Therefore, identifying and disambiguating software mentions, while attending to its metadata, is an essential yet challenging task. Various Knowledge Graph resources, such as OpenAire (Manghi et al., 2019) and SoftwareKG (Schindler et al., 2020), link open-access articles to the software used, supporting the need for robust methods to identify, extract, link, and disambiguate software mentions.

Various existing citation principles regarding software usage and mentions(Katz et al., 2021; Smith et al., 2016) promote knowledge sharing and innovation. However, these principles are not always strictly followed in all works, resulting in informal and incomplete information regarding the software mentioned or used (Schindler et al., 2024). Robust Information Extraction (IE) methods help to detect and disambiguate software mentions and related metadata. SoMeSci (Schindler et al., 2021) is a manually curated gold standard corpus about software mentioned in scientific articles, providing training samples for Named Entity Recognition (NER), Relation Extraction(RE), Entity Disambiguation(ED), and Entity Linking(EL). Based on this dataset, the SOMD2024 shared task was organized to advance research on automatic detection and analysis of software mentions in scholarly articles. The task challenged participants to develop methods for (i) Detecting Software Mentions, (ii) Identifying Associated Attributes, and (iii) Classifying the Relations between Software

Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025), pages 137–145

and their Attributes.

In this paper, we present the Software Mention Detection shared task (SOMD2025)—the successor to SOMD2024 (Krüger et al., 2024). The goal is to advance the field through community-driven development and evaluation of new methods. SOMD2025 builds on the success of the previous edition. But while the first iteration focused on establishing NER, attribute detection and RE for software mentions in separate subtasks, SOMD2025 emphasizes a joint evaluation of these subtasks. Our task advances the development of a pipeline for IE components (NER; RE) for scientific knowledge. These pipelines serve as an initial step for functions such as metadata enrichment, semantic linking, and knowledge graph construction from scholarly articles, aligning with NFDI4DS's[1] and BERD@NFDI's[2] broader mission of supporting the research data lifecycle and providing infrastructures. We focus on the discovery and traceability of the software mentioned in research publications—a crucial step in the reproducibility of research.

In addition to learning and evaluating the joint NER and RE framework, we introduce an out-of-distribution (OOD) test set to assess the generalizability of the models—a significantly more challenging benchmark compared to the in-distribution data. We hosted the two subsequent phases of the competition in the CodaBench platform (Xu et al., 2022). Phase I aims at model development, where we provide gold-standard training and test splits to the participants. Phase II challenges participants to apply their models from Phase I on an out-of-distribution dataset comprised of scholarly documents that were not part of the training or test set used in Phase I. Although 18 participants registered, only three teams submitted for Phase I and five teams made submission for Phase II. Four of them submitted a system description for the workshop proceedings. To encourage future research, we have transformed Phase 2 into an Open Submission Phase that will allow further development of IE systems for our task.[3]

We provide the competition details in the rest of the paper. We include the task description and the evaluation metrics in section 3 and a description of the dataset for both phases in section 3.2. We summarize results in section 5, where we compare the methods of different participants.

[1]https://www.nfdi4datascience.de/

[2]https://www.berd-nfdi.de/

[3]https://www.codabench.org/competitions/5840/

2 Related Work

Software Mention Recognition. Early efforts in recognizing software mentions in scientific articles relied on manual analysis of small corpora (Howison and Bullard, 2016; Nangia and Katz, 2017) or targeted extraction of specific tools (Li et al., 2017, 2016). Automatic approaches such as rule-based systems and bootstrapping offered moderate performance (Pan et al., 2015; Duck et al., 2016). Deep learning models, particularly BiLSTM-CRF architectures (Schindler et al., 2020), improved accuracy but required more robust annotated datasets. Recently, transformer-based models like SciBERT trained on the SoMeSci corpus (Schindler et al., 2022, 2021) achieved state-of-the-art NER results. Importantly, SoMeSci also includes annotations for software attributes (e.g., version, license) (Schindler et al., 2021) and their links to software mentions, providing a foundation for relation extraction. Similarly, SoftwareKG (Schindler et al., 2020) offers a knowledge graph of software entities and metadata mined from scientific literature, further highlighting the need for integrated NER and RE.

SOMD Shared Task. The SOMD2024 shared task built upon these efforts by targeting software mention detection, attribute recognition, and relation classification using the SoMeSci corpus (Schindler et al., 2021). Participants explored diverse modeling approaches, including large language models and encoder/decoder architectures (Khan et al., 2024; Otto et al., 2024; Thi et al., 2024; Nguyen Xuan et al., 2024). Unlike the prior edition, which handled tasks independently, this year's task emphasizes joint learning and evaluation of NER and RE to encourage integrated solutions.

Joint NER and Relation Extraction. Joint learning of NER and RE has emerged as a robust alternative to traditional pipeline approaches, which often suffer from error propagation. Integrated models have demonstrated improved accuracy and efficiency by simultaneously extracting entities and their relations (Hennen et al., 2024; Huguet Cabot and Navigli, 2021; Wadden et al., 2019; Ye et al., 2022a). While these models have been widely adopted in general and biomedical domains, only a few efforts—such as SoMeSci and SoftwareKG—explicitly address relation-level modeling in the software domain. Their contributions underscore

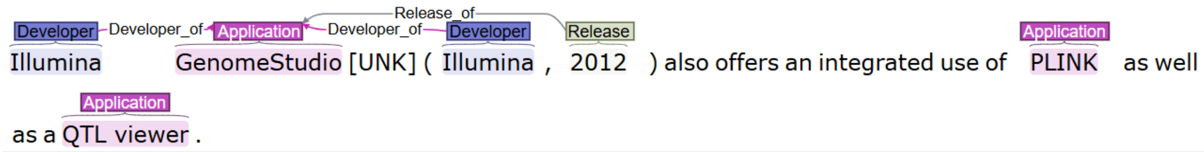

Figure 1: Illustration of NER and relation extraction annotations in the input data.

the growing importance of joint models in domain-specific information extraction. Prior studies consistently show joint frameworks outperform pipeline systems (Li and Ji, 2014), addressing limitations in earlier methods (Zeng et al., 2014; Zhang et al., 2017).

3 Task Description

We focus on the discovery and traceability of the software mentioned in research publications—a crucial step to ensure the reproducibility of research. For this purpose, we propose Information Extraction of software and related metadata, including Named Entity Recognition and Relation Extraction. We approach the concept of software as a form of research artifact, with software-related IE serving as a foundational element in the construction of Research Knowledge Graphs (RKGs) (Schindler et al., 2021; Karmakar et al., 2023). These RKGs, in turn, are built upon a foundation of scholarly articles, thereby facilitating the aggregation and organization of research findings. We encourage participants to build robust and generalizable NLP methods, i.e., models for software mentions, attribute detection, and relation extraction. An instance of a sentence with annotated software mentions, attributes, and relations is illustrated in Figure1.
SOMD2024 (Krüger et al., 2024) had hosted these problems as three independent subtasks. SOMD2025 combines these three subtasks into an end-to-end setup for training and evaluation. We endorse jointly learning the automatic extraction of software mentions, its attributes, and their relations from scholarly documents. Our task belongs to the well-known problem in information extraction, i.e., Joint Learning Paradigm for NER and RE (Li and Ji, 2014; Huguet Cabot and Navigli, 2021; Hennen et al., 2024). We have two phases of competition. We provide the labeled dataset for phase I, supporting model training. It contains three aligned files per instance: a tokenized text file, a NER label file, and a relation label file, each line corresponding to a sentence. The NER file uses IOB2 tagging with entity types such as Application, Abbreviation, and Version. The relation file encodes binary relations as <relation_type> <head_index> <tail_index>, with indices referencing the starting tokens (0-based). The test set in Phase I and II includes only the tokenized text, and participants are required to submit predicted NER and relation files. Full format details are available on the competition page[4].

3.1 Shared Task Schedule

Phase I: Model Development. Given the labeled gold dataset, participants develop an initial model for joint NER and RE for the gold standard dataset. Participants submit their outcomes on an unlabeled test/development set with the same distribution as a training set as they belong to the same original dataset.

Phase II: To test the generalizability of Phase I approaches, we deliver in Phase II an out-of-distribution test set for the same task. The goal of this phase is to adapt and refine models designed for Phase I to handle out-of-distribution data effectively.

Open Submission Phase: After the end of the competition, we initialize an open submission phase inviting researchers to submit their results on the benchmark dataset from Phase II. This phase is not part of the competition but an initiative encouraging ongoing collaboration and facilitating long-term engagement within the research community.

3.2 Dataset

We utilize the gold standard SoMeSci (Schindler et al., 2021) corpus for Phase I, which comprises 3756 manually annotated software mentions from 1367 PubMed Central articles. It supports Named Entity Recognition, Relation Extraction, Entity Disambiguation, and Entity Linking. Annotations include software version, developer, URL, citations,

[4] https://www.codabench.org/competitions/5840/

mention type (e.g., usage, creation), and software type (e.g., application, plugin). There are a total of 7237 labeled entities across 47,524 sentences. We resample the original corpus to create predefined training and testing splits for NER and RE. We manually add negative samples, i.e., sentences without entities and relations, to better simulate real-world data scenarios. We show the statistics of the overall dataset and individual entity label and relation label distributions in Table 3.

For Phase II, we sample PubMed Central Open Access scientific articles. We automatically annotate these articles using a state-of-the-art model (Schindler et al., 2022) based on SciBERT (Beltagy et al., 2019), trained on the SoMeSci gold-standard benchmark dataset (Schindler et al., 2021), to extract software mentions, their attributes, and the relations between them. We consider this a weakly labeled dataset. The overall statistics of detected named entities and relations are provided in the Table 4. To create a gold standard labeled test set, five annotators; three are master's students in relevant fields, and two PhD candidates reviewed and corrected the weakly labeled test set. We use the same annotation guidelines as the original SoMeSci corpus to ensure consistency. Table 4 compares dataset statistics before (weakly annotated) and after review.

3.3 Scoring Metric

We use the same evaluation metrics for all phases. We evaluate the NER and RE performance using the F1 score on exact matches. We opted for the macro F1 score as our dataset is imbalanced, as shown in Table 3 and 4. This decision ensures equal evaluation importance for all classes, regardless of the class frequency. As a final metric to evaluate the competing approaches, we use the mean of macro F1 for NER and macro F1 for RE. This 'F1 SOMD' called metric favors IE systems, which are able to perform well on both tasks, i.e., NER and RE.

3.4 Submissions

The shared task competition encompassed two phases from February 24 to April 3, 2025. Registration began on February 24, followed by the initial training and test data release on February 27. Phase I ran from February 27 to March 25, during which participants could submit up to 5 daily runs. Phase II started with a new dataset release on March 25 and closed on April 3, allowing five daily submissions. The open post-evaluation phase on Codabench allows 10 daily submissions per participant, enabling further experimentation and result refinement.

4 Participants and Approaches

A total of 18 teams registered for the SOMD2025 shared task. Three teams participated fully by submitting results in both Phase I and Phase II, as well as providing a system description. These teams were the TU Graz Data Team (TUGraz), a team from the Nepal-based company EKbana, and one participant from the Universidade de Aveiro (UAveiro). Additionally, there was one late participation, consisting of a master's student from the Georgia Institute of Technology and an independent researcher (psr123), who submitted only for Phase II and provided a system description.

These four teams are referred to as the final participants in this paper. One further participant submitted results in Phase II but did not provide a system description and is therefore not discussed further. All final participants used the open submission phase to further test and refine their approaches after the conclusion of Phase II. In this section, we introduce the four final approaches alongside two baseline models.

4.1 Approaches

All final participants employed finetuning approaches, with some leveraging additional training data, as detailed in Table 1. All teams utilized pretrained language models (PLMs). The largest model used for finetuning was DeBERTa v3, comprising up to 418 million parameters, including the embedding layer (He et al., 2021a,b). The largest model applied for generating embeddings without layer finetuning was the Multilingual E5 instruct model with 560 million parameters (Wang et al., 2024). One participant incorporated a graph neural network (GNN) based on the words of parsed input sentences, with edges defined by their dependency tree (UAveiro). Additionally, this team used DeepSeek v3 (Liu et al., 2024) to classify detected relation types. Regarding loss strategies, only one team (TUGraz) and our baseline approach adopted a joint loss for NER and RE. The remaining teams trained RE and NER modules separately and applied a pipeline approach for inference on the test data.

In terms of data augmentation and generated training data, two out of four approaches utilized addi-

Table 1: Overview of approaches used in SOMD2025. The loss strategy reflect the usage of joint learning for the NER and RE task in contrast to train separate models with separate losses.

Team	Model Architecture	PLM	Loss Strategy	Data Augmentation
TUGraz	Transformer	DeBERTa v3	joint	—
EKbana	Transformer + Adapter	ModernBERT	separate	SOMD2024 + LLM Generated
psr123	Transformer	DeBERTa v3	separate	Negative Samples + LLM Generated
UAveiro	GNN + Transformer	Multilingual E5	separate	—
Baseline	HGERE (Transf. + GNN)	SciBERT	joint	—

tional training data. One team (psr123) augmented the training data specifically with sentences from the same domain as the test set that contain no mentions, to expose the model to negative examples. Another team (EKbana) used the SOMD2024 dataset as additional training data. Furthermore, new training samples were generated using large language models (LLMs) by both EKbana and psr123.

4.2 Baseline Model

Recent work has shown that supervised NER and RE with small language models can achieve strong performance on scholarly information extraction tasks (Yan et al., 2023; Zhang et al., 2024). Among the current approaches, joint models that unify entity and relation prediction have gained attention for their ability to capture dependencies between tasks. In our experiments, we adopt HGERE (Yan et al., 2023) as a joint baseline model. HGERE extends the marker-based PL-Marker framework (Ye et al., 2022b) by introducing a hypergraph neural network that models interactions between subjects, objects, and relations. We selected HGERE due to its effective integration of task components and its demonstrated performance in similar domains.

5 Results

In this section, we present the results of the SOMD2025 shared task, including performance scores for both phases of the competition. For this section, we focus on the more challenging Phase II test set because it better illustrates the generalization capabilities of the used IE models. We compare the results of all final participating teams, highlight the top-performing systems, and contrast them with baseline models. Additionally, we include results from the non-competitive open submission phase as including unpublished Codabench results reported in the corresponding system descriptions of the teams. This provides further insight into model improvements beyond the official evaluation period. The main results can be found in Table 2, illustrating TUGraz as the winner of the challenging Phase II with a SOMD score of 0.63. The TU Graz team used a joint loss for NER and RE and was not dependent on data augmentation to achieve that score.

Note that two of four teams were not able to submit RE results in time, illustrating the hurdle to overcome to switch from well-established NER models to RE models. The leading competing approaches, TUGraz (0.69 SOMD score in the non-competitive version for Phase II) and our proposed baseline model HGERE (0.62 SOMD score for Phase II), both employ a joint loss for the NER and RE tasks without utilizing additional training data. The UAverio performance results report that an unconventional approach utilizing a dependency graph-based representation of language is not able to achieve the same results as transformer-based approaches. Transformer-based approaches are able to use attention to mitigate information between all tokens directly.

6 Discussion

6.1 The Role of LLMs

None of the participants used prompting of LLMs as a final competition approach. But some of the approaches used LLMs in other roles. TUGraz is the only team reporting performances for prompting approaches without any finetuning. They tested only NER in Phase I, achieving a macro F1 of 0.39 with Gemini 2 in a zero-shot approach. Additionally, they reported results of LLaMA 3 8B (Grattafiori et al., 2024) finetuning for Phase I, a SOMD score of 0.66. Compared to finetuning approaches based on smaller language models, these results led the team to the decision not to pursue this direction further.

Two other teams, EKbana and psr123, experimented with synthetic training data generated by LLMs. Team psr123 used existing entities from available training samples and asked models to

Table 2: macro F1 score Results for SOMD2025 Shared Task. SOMD score is the mean of NER and RE macro F1.

Submission	Team	Phase I (macro F1)			Phase II (macro F1)		
		SOMD	NER	RE	SOMD	NER	RE
official	TUGraz	88	90	**85**	**63**	**68**	**57**
	EKbana	***89***	***93***	84	55	64	46
	psr123	–	–	–	32	65	0
	UAverio	39	45	34	15	30	0
non-competitive	TUGraz	–	–	–	*69*	*77*	*62*
	Baseline	*89*	91	*87*	62	68	57
	EKbana	–	–	–	60	69	50
	psr123	–	–	–	56	65	47
	UAverio	–	–	–	22	44	0

produce new contexts mentioning the same entities. They experimented with synthetic data from three different models, with the best configuration (samples generated with Mistral 7B) resulting in a performance gain of 6% points for macro F1 NER. The observed performance gain can be primarily attributed to a significant increase in precision. Team EKbana attempted to tweak results in the out-of-distribution based Phase II by searching for new vocabulary in the Phase II test set sentences compared to Phase I data. They then used these new terms as input to produce new training examples, aiming to adapt their Phase I model to the new distribution of the test data. This approach led to a performance boost of 0.09 SOMD score after several experiments utilizing this data. Whether this approach is generalizable to other distribution shifts, such as domain shift, remains to be proven in future research.

The last usage example among participants was the role of a relation classifier in UAverio's approach. Their model outputs relation candidates in the form of entity mentions and they prompted a Deepseek v3 model to identify the correct relation direction and label. Nonetheless, the overall mediocre performance does not provide valuable interpretability regarding the promise of this approach.

6.2 The Impact of Additional Training Data

Team psr123 showed that adding negative sentences (i.e., sentences without any software mentions) significantly improved the performance of their RE model, from 0.15 to 0.47 macro F1 on the Phase II test set. However, team TUGraz demonstrated that a similar experiment using DeBERTa v3 (He et al., 2021a) with a separate loss for RE achieved a higher RE macro F1 of 0.56 without additional negative samples. This suggests that a deeper analysis of implementation details and hyperparameter settings is needed to accurately assess the impact of adding negative examples.

Team EKbana's use of SOMD2024 data as additional training data for Phase I deserves special attention. As described in Section 3.2, the SOMD2025 Phase I data is a resampled version of the SOMD2024 dataset. This results in data leakage when SOMD2024 data is used for training. EKbana's Phase I result in Table 2 should be interpreted with this in mind.

6.3 Loss Strategy

Team TUGraz highlighted the effectiveness of using a joint loss for NER and RE in their system description. Their experiments showed a performance improvement of 1 to 10 SOMD score points compared to training with separate losses. Our Baseline approach, which also relies on a joint loss, supports the conclusion that selecting an appropriate model architecture—and in particular, the loss function—is more critical than adding extra training data, whether synthetically generated or composed of additional negative sentences. A well-defined experimental setup and careful design choices enabled team TUGraz to achieve the best performance and win the shared task.

7 Conclusion

The SOMD2025 shared task addressed the challenge of extracting software mentions, attributes, and relations from scientific articles using a joint NER and RE framework. With two evaluation phases, including an out-of-distribution test set, the task emphasized both extraction accuracy and model generalizability. Participating teams employed diverse strategies, including pretrained language models, graph-based architectures, and data augmentation using LLMs. Results show that while current methods perform well on in-distribution data, generalization remains a significant challenge.

Table 3: Dataset Overview: Sentence Statistics, Entity and Relation Label Distributions(Phase I)

(a) Dataset Split Summary. **Pos.** denotes the number of sentences that have both entities and relations. Neg. denotes the number of sentences that have no relation label. Negatives are split into (i) sentences with entities but no relations, and (ii) sentences with neither entities nor relations.

Split	# Sents	Pos.	Neg. (Ent/None)
Train	1149	1021	16 / 112
Test	203	182	3 / 18

(b) Entity Label Distribution

Entity	Train	Test
Application	1232	217
Version	904	168
Developer	616	125
Citation	382	53
ProgrammingEnvironment	234	37
URL	216	32
PlugIn	211	34
OperatingSystem	146	22
Release	69	13
Abbreviation	58	4
Extension	43	7
License	43	7
SoftwareCoreference	14	1
AlternativeName	14	2

(c) Relation Label Distribution

Relation	Train	Test
Version_of	904	168
Developer_of	623	126
Citation_of	387	53
URL_of	218	32
PlugIn_of	141	25
Release_of	69	13
Abbreviation_of	58	4
Specification_of	53	14
Extension_of	44	7
License_of	40	7
AlternativeName_of	14	2

The top systems showed improvements over previous baselines, particularly in Phase I, and provided valuable insights into joint learning strategies and training data choices. The shared task supports sustained progress in software-related information extraction from scholarly texts by continuing with an open, ongoing submission phase.

8 Limitations

The current setup of the SOMD shared task is constrained by the lack of a representative distribution of negative samples across both the training and test sets. Furthermore, the scope of the research is limited to the biomedical domain, as determined by the selection of relevant open access publications. Additionally, the methodology of the shared task does not incorporate a disambiguation step, which is identified as a direction for future work.

9 Acknowledgment

This work has received funding through the DFG projects NFDI4DS (grant number 460234259) and BERD@NFDI (grant number 460037581). We thank both NFDI4DS and BERD@NFDI for their funding and support. Special thanks go to all institutions and individuals contributing to the association and its goals.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. Scibert: A pretrained language model for scientific text. In *Proceedings of EMNLP*.

Geraint Duck, Goran Nenadic, Michele Filannino, Andy Brass, David L Robertson, and Robert Stevens. 2016. A survey of bioinformatics database and software usage through mining the literature. *PloS one*, 11(6):e0157989.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Table 4: Phase-2 Dataset Overview: Sentence Statistics, Entity and Relation Label Distributions. The reviewed set is a manually corrected subset of the weakly labeled data.

(a) Entity Label Distribution

Entity Type	Weak	Reviewed
Application	662	363
Version	135	96
Developer	47	20
Citation	216	187
ProgrammingEnvironment	70	24
URL	84	70
PlugIn	38	20
OperatingSystem	7	2
Release	10	10
Abbreviation	19	12
Extension	7	6
License	-	-
SoftwareCoreference	4	3
AlternativeName	18	17

(b) Relation Label Distribution

Relation Type	Weak	Reviewed
Version_of	134	96
Developer_of	41	20
Citation_of	173	187
URL_of	72	70
PlugIn_of	22	13
Release_of	8	10
Abbreviation_of	16	12
Specification_of	12	-
Extension_of	5	6
License_of	-	7
AlternativeName_of	14	17

Moritz Hennen, Florian Babl, and Michaela Geierhos. 2024. ITER: Iterative transformer-based entity recognition and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11209–11223, Miami, Florida, USA. Association for Computational Linguistics.

James Howison and Julia Bullard. 2016. 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.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Saurav Karmakar, Matthäus Zloch, Fidan Limani, Benjamin Zapilko, Sharmila Upadhyaya, Jennifer D'Souza, Leyla J. Castro, Georg Rehm, Marcel R. Ackermann, Harald Sack, Zeyd Boukhers, Sonja Schimmler, Danilo Dessí, Peter Mutschke, and Stefan Dietze. 2023. Research knowledge graphs in nfdi4ds. In *INFORMATIK 2023 - Designing Futures: Zukünfte gestalten*, pages 909–918. Gesellschaft für Informatik e.V., Bonn.

Daniel S Katz, Neil P Chue Hong, Tim Clark, August Muench, Shelley Stall, Daina Bouquin, Matthew Cannon, Scott Edmunds, Telli Faez, Patricia Feeney, and 1 others. 2021. Recognizing the value of software: a software citation guide. *F1000Research*, 9:1257.

AmeerAli Khan, Qusai Ramadan, Cong Yang, and Zeyd Boukhers. 2024. Falcon 7b for software mention detection in scholarly documents. In *International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs*, pages 278–288. Springer Nature Switzerland Cham.

Frank Krüger, Saurav Karmakar, and Stefan Dietze. 2024. Somd@ nslp2024: Overview and insights from the software mention detection shared task. In *International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs*, pages 247–256. Springer.

Frank Krüger, Saurav Karmakar, and Stefan Dietze. 2024. Somd@nslp2024: Overview and insights from the software mention detection shared task. In *Natural Scientific Language Processing and Research Knowledge Graphs: First International Workshop, NSLP 2024, Hersonissos, Crete, Greece, May 27, 2024, Proceedings*, page 247–256, Berlin, Heidelberg. Springer-Verlag.

Kai Li, Xia Lin, and Jane Greenberg. 2016. Software citation, reuse and metadata considerations: An exploratory study examining lammps. *Proceedings of the Association for Information Science and Technology*, 53(1):1–10.

Kai Li, Erjia Yan, and Yuanyuan Feng. 2017. How is r cited in research outputs? structure, impacts, and citation standard. *Journal of Informetrics*, 11(4):989–1002.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Paolo Manghi, Alessia Bardi, Claudio Atzori, Miriam Baglioni, Natalia Manola, Jochen Schirrwagen, Pedro Principe, Michele Artini, Amelie Becker, Michele De Bonis, and 1 others. 2019. The openaire research graph data model. *Zenodo*.

Udit Nangia and Daniel S Katz. 2017. 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. IEEE.

Phi Nguyen Xuan, Quang Tran Minh, and Thin Dang Van. 2024. Abcd team at somd 2024: Software mention detection in scholarly publications with large language models. In *International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs*, pages 267–277. Springer Nature Switzerland Cham.

Wolfgang Otto, Sharmila Upadhyaya, and Stefan Dietze. 2024. Enhancing software-related information extraction via single-choice question answering with large language models. In *International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs*, pages 289–306. Springer.

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

David Schindler, Felix Bensmann, Stefan Dietze, and Frank Krüger. 2021. Somesci- a 5 star open data gold standard knowledge graph of software mentions in scientific articles. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 4574–4583, New York, NY, USA. Association for Computing Machinery.

David Schindler, Felix Bensmann, Stefan Dietze, and Frank Krüger. 2022. The role of software in science: a knowledge graph-based analysis of software mentions in pubmed central. *PeerJ Computer Science*, 8:e835.

David Schindler, Tazin Hossain, Sascha Spors, and Frank Krüger. 2024. A multilevel analysis of data quality for formal software citation. *Quantitative Science Studies*, 5(3):637–667.

David Schindler, Benjamin Zapilko, and Frank Krüger. 2020. Investigating software usage in the social sciences: A knowledge graph approach. In *European Semantic Web Conference*, pages 271–286. Springer.

Arfon M Smith, Daniel S Katz, and Kyle E Niemeyer. 2016. Software citation principles. *PeerJ Computer Science*, 2:e86.

Thuy Nguyen Thi, Anh Nguyen Viet, and Thin Dang Van. 2024. Software mention recognition with a three-stage framework based on bertology models at somd 2024. *Natural Scientific Language Processing and Research Knowledge Graphs*, page 257.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5783–5788.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.

Zhen Xu, Sergio Escalera, Adrien Pavão, Magali Richard, Wei-Wei Tu, Quanming Yao, Huan Zhao, and Isabelle Guyon. 2022. Codabench: Flexible, easy-to-use, and reproducible meta-benchmark platform. *Patterns*, 3(7).

Zhaohui Yan, Songlin Yang, Wei Liu, and Kewei Tu. 2023. Joint entity and relation extraction with span pruning and hypergraph neural networks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7512–7526, Singapore. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022a. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022b. Packed levitated marker for entity and relation extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4904–4917, Dublin, Ireland. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guoliang Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 2335–2344.

Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. 2024. SciER: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13083–13100, Miami, Florida, USA. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 35–45.