From In-Distribution to Out-of-Distribution: Joint Loss for Improving Generalization in Software Mention and Relation Extraction

Stasa Mandic[1], Georg Niess[1], Roman Kern[1,2]
[1]Graz University of Technology, [2]Know Center Research GmbH
Correspondence: stasa.mandic@student.tugraz.at

Abstract

Identifying software entities and their semantic relations in scientific texts contributes to improved reproducibility and allows for the construction machine-readable knowledge graphs. However, models struggle with domain variability and sparse supervision. We address this by evaluating joint Named Entity Recognition (NER) and Relation Extraction (RE) models on the SOMD 2025 shared task, emphasizing generalization to out-of-distribution scholarly texts. We propose a unified training objective that jointly optimizes both tasks using a shared loss function and demonstrates that joint loss formulations can improve out-of-distribution robustness compared to disjoint training. Our results reveal significant performance gaps between in- and out-of-distribution settings, prompting critical reflections on modeling strategies for software knowledge extraction. Notably, our approach ranked 1st in Phase 2 (out-of-distribution) and 2nd in Phase 1 (in-distribution) in the SOMD 2025 shared task, showing strong generalization and robust performance across domains. All code is publicly available.[1]

1 Introduction

Software is crucial to scientific work, but identifying its mentions and relations in text is difficult due to ambiguity, limited supervision, and domain variation (Howison and Bullard, 2016; Pan et al., 2015). The Software Mention Detection (SOMD) shared task series addresses these challenges through benchmark datasets and evaluation frameworks. While the 2024 edition focused on pipeline approaches using full-text articles from the SoMeSci corpus (Dietze et al., 2024), the 2025 task shifts to joint modeling of NER and RE at the sentence level to reduce cascading errors (Li and Ji, 2014; Zeng et al., 2014; Huguet Cabot and Navigli, 2021). In this work, we evaluate joint NER and RE models for software knowledge extraction under domain shift. We compare span-based (GLiNER (Zaratiana et al., 2024)), encoder-based (BERT, SciBERT, and DeBERTa (Devlin et al., 2019; Beltagy et al., 2019; He et al., 2023)), and instruction-tuned architectures (Gemini and Llama (Comanici et al., 2025; Grattafiori et al., 2024)) on the SOMD 2025 benchmark. Our central research question is: **Does a joint loss objective improve generalization in multitask NER and RE models?** We find that joint loss boosts in-distribution performance and consistently mitigates degradation in out-of-distribution settings. This highlights its utility as a simple yet effective mechanism for improving robustness in extractive multitask learning.

2 Related Work

Extracting software mentions and their semantic relations from scientific texts is crucial for reproducibility and knowledge organization, yet software is often referenced informally, posing challenges for automatic identification and disambiguation (Schindler et al., 2021). Early approaches treated NER and RE as separate pipeline stages (Li and Ji, 2014; Zeng et al., 2014), but this modularity frequently led to cascading errors, particularly when entity boundaries were misidentified (Zhang et al., 2017). To mitigate these issues, recent work has shifted toward joint models that unify NER and RE within a single architecture. Span-based methods leverage contextualized representations to jointly encode entities and their relations (Wadden et al., 2019; Ye et al., 2022), while generation-based architectures such as REBEL (Huguet Cabot and Navigli, 2021) and iterative decoding frameworks (Hennen et al., 2024) aim to improve expressiveness and compositional generalization.
In addition, a growing body of work has focused

[1]https://github.com/sm9ta/somd2025-joint-loss

Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025), pages 146–153

on discourse-aware and document-level models, which extend the context window beyond a single sentence. For example, Wadden et al. (2019) introduce a span-based architecture that propagates information across sentences using global context graphs, while Ye et al. (2022) use levitated markers to retain global coreference and discourse signals. Models like SciREX (Jain et al., 2020) further highlight the importance of integrating paragraph-level and document-level context to improve entity linking and relation reasoning in scientific documents. These approaches demonstrate that sentence-local models are often insufficient for resolving long-range dependencies, a limitation we also observe in our results. We extend this line of research by comparing joint and disjoint loss formulations across model families and analyzing their impact on generalization, particularly under distribution shift.

In the SOMD 2024 shared task, participants explored alternatives to traditional pipeline systems. Thi et al. (2024) proposed a three-stage BERT-based pipeline, while Otto et al. (2024) used an instruction-tuned LLM for QA-style extraction. Others investigated few-shot adaptation with GPT-3.5/4 (Istrate et al., 2024) or token-level fine-tuning with Falcon-7B (Khan et al., 2024), highlighting the challenges of aligning LLMs with structured tasks. Building on these efforts, our work unifies evaluation across model types and emphasizes generalization under domain shift, responding to recent calls for more robust extraction frameworks (Krüger et al., 2024).

3 Task and Dataset

The SOMD 2025 task involves a two-phase sentence-level joint NER and RE on annotated scientific texts from the SoMeSci corpus (Schindler et al., 2021). The annotation schema includes 14 entity types and 11 relation types that can be found in Appendix A. In **Phase 1 (In-Distribution Validation Set)** models are trained and evaluated on 1,432 annotated sentences (train) and 256 test sentences, while in **Phase 2 (Out-of-Distribution Test Set)** 457 new test sentences from unseen domains are released without gold labels. Models are evaluated via leaderboard submissions, with an additional focus on generalization.

4 Method

4.1 Model Selection

In early experiments, we evaluated instruction-tuned decoder-based models (Gemini 2 and LLaMA 3 8B) in zero and few-shot configurations without fine-tuning. However, their performance was not satisfactory as they failed to follow SOMD's strict schema, hallucinated outputs, and lacked token-level precision, issues also noted in prior work (Otto et al., 2024; Istrate et al., 2024), which highlights the limitations of instruction-following LLMs in structured extraction. We therefore excluded them from joint training and focused on encoder-based models with proven suitability for NER and RE.

We selected DeBERTa-v3, SciBERT, and GLiNER based on their complementary strengths and empirical performance in previous NER and RE benchmarks. SciBERT is pretrained on 1.14M scientific papers from Semantic Scholar and is specifically optimized for the scientific domain (Beltagy et al., 2019), making it particularly suited to the SOMD corpus. DeBERTa-v3 incorporates disentangled attention and enhanced mask decoding, which improve generalization across domains, especially in out-of-distribution settings (He et al., 2023). GLiNER, a span-based model, provides strong performance on fine-grained entity recognition due to its ability to directly model entity spans without relying on token-level tagging (Zaratiana et al., 2024). This is particularly useful in software-related texts where entity boundaries may be ambiguous.

Other prominent pretrained models such as RoBERTa or BioBERT were not included in our final evaluation due to domain misalignment (e.g., biomedical corpora in BioBERT (Lee et al., 2020)) or redundancy with DeBERTa in terms of architectural class. We also did not include models such as T5 (Raffel et al., 2020) or BART (Lewis et al., 2020), as their sequence-to-sequence format is less compatible with structured joint token- and span-level prediction. In the end, our goal was not exhaustive benchmarking, but rather a focused comparison across representative architectures: span-based (GLiNER), domain-specialized encoder (SciBERT), and general-purpose contextual encoder (DeBERTa) to assess how model inductive biases influence generalization under joint optimization.

4.2 Model Architecture

Given an input sentence $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, the encoder produces hidden states $\mathbf{h} = \{h_1, h_2, \ldots, h_N\}$, which serve as the basis for downstream prediction tasks. On top of the encoder, we add two task-specific classification heads. The first is a token-level classifier for NER that predicts BIO-encoded entity labels for each token. The second head is a relation classifier that predicts the type of semantic relation between entity pairs, based on the concatenation of their respective token representations. For each candidate pair (i, j) of detected entities, their corresponding embeddings h_i and h_j are concatenated and passed to a feedforward classification layer. In the end, we evaluate multiple encoder backbones, including SciBERT, DeBERTa-v3-large, and GLiNER.

Figure 1 illustrates the architecture and information flow in our joint NER and RE model, including task-specific heads and shared optimization through a unified loss.

4.3 Joint Loss Objective

To encourage shared representations between NER and RE, we optimize a joint loss function. The NER loss is a masked token-level cross-entropy over K entity classes:

$$\mathcal{L}_{\text{NER}} = -\sum_{i=1}^{N} m_i \log \frac{\exp(z_{i,y_i})}{\sum_{k=1}^{K} \exp(z_{i,k})} \quad (1)$$

where m_i masks out incomplete tokens, z_i are the logits, and y_i are gold labels. The RE loss is computed over a set of candidate entity pairs $\mathcal{R}$, with L relation types (including a "no-relation" class):

$$\mathcal{L}_{\text{RE}} = -\sum_{(i,j)\in\mathcal{R}} \log \frac{\exp(z_{ij,r_{ij}})}{\sum_{l=1}^{L} \exp(z_{ij,l})} \quad (2)$$

The total loss is:

$$\mathcal{L} = \mathcal{L}_{\text{NER}} + \lambda\, \mathcal{L}_{\text{RE}} \quad (3)$$

where λ is a tunable weight (default $\lambda = 1$).

4.4 Training Setup

We fine-tune all models using AdamW with warm-up and early stopping. The training is done for 3 to 6 epochs depending on the model with batch size 8 and learning rate 3×10^{-5}. Dropout and gradient clipping are applied, and the best checkpoints are selected via dev set macro F1.

4.4.1 Joint Training Procedure

Algorithm 1 outlines our joint training procedure. For each batch, contextualized embeddings are computed, token-level predictions and loss for NER are obtained, and relation classification is performed on candidate entity pairs. The two losses are then combined into a single objective, and a joint backward pass is used to update all model parameters simultaneously. To evaluate the effect of shared optimization, we also implement a disjoint training variant where NER and RE tasks are optimized separately in isolated stages and gradients are not shared across tasks during training.

Algorithm 1 Joint Training for NER and Relation Extraction

Require: Training dataset $\mathcal{D} = \{(X, Y, \mathcal{R})\}$, where X denotes the input token sequence, Y the corresponding token labels, and $\mathcal{R}$ the relation annotations.

Initialize model parameters Θ (shared encoder, NER head, and RE head).
for each epoch **do**
  for each batch $(X, Y, \mathcal{R}) \in \mathcal{D}$ **do**
    $H \leftarrow \text{Encoder}(X)$ {Compute contextual representations}
    $Z^{\text{NER}} \leftarrow \text{NER\_Head}(H)$
    Compute masked NER loss using the cross-entropy loss $\text{CE}(\cdot)$:

$$\mathcal{L}_{\text{NER}} \leftarrow \sum_{i=1}^{N} m_i \,\text{CE}(Z_i^{\text{NER}}, y_i)$$

    Generate candidate entity pairs $\mathcal{C}$ from Y (using gold labels during training or predicted labels at inference).
    for each candidate pair $(i, j) \in \mathcal{C}$ **do**
      $r_{ij} \leftarrow \text{Concat}(H_i, H_j)$
      $Z_{ij}^{\text{RE}} \leftarrow \text{RE\_Head}(r_{ij})$
    end for
    Compute RE loss:

$$\mathcal{L}_{\text{RE}} \leftarrow \sum_{(i,j)\in\mathcal{C}} \text{CE}(Z_{ij}^{\text{RE}}, r_{ij}^{\text{gt}})$$

    Compute joint loss: $\mathcal{L} \leftarrow \mathcal{L}_{\text{NER}} + \mathcal{L}_{\text{RE}}$
    Perform backpropagation: update Θ using $\nabla\mathcal{L}$ (e.g., via AdamW with a linear scheduler).
  end for
end for

5 Results

We evaluate joint NER and RE models on the SOMD 2025 benchmark using span-based (GLiNER), encoder-based (SciBERT, DeBERTa-v3). All models are jointly fine-tuned using both a disjoint and unified loss objective and evaluated across in-distribution (Phase 1) and out-of-distribution (Phase 2) subsets while their performance is summarized in Table 1. We report total and macro-averaged F1, precision, and recall as per SOMD 2025 guidelines. NER is evaluated using token-level exact matches under the IOB2 scheme, while RE requires exact entity span and relation label matches. Phase 2 uses leaderboard submissions due to the current unavailability of gold annotations.

Phase 1: In-Distribution Performance We fine-tuned GLiNER, DeBERTa v3, and SciBERT on the official training split and evaluated them on the development set. GLiNER (base and large-v2.1) was fine-tuned with improved label alignment; SciBERT incorporated a feedforward classifier over span start tokens; and DeBERTa v3 (tasksource-nli) demonstrated stronger stability than its Microsoft variant. In the end, we report that GLiNER outperforms other models with a total F1 of 0.88, followed by DeBERTa-v3 (0.83) and SciBERT (0.79).

Phase 2: Out-of-Distribution Generalization In Phase 2, all models showed a noticeable drop in performance, highlighting the challenge of out-of-distribution generalization. DeBERTa v3 performed the best (Total F1: 0.69; NER: 0.79; RE: 0.62), likely due to its strong contextual modeling and robust sentence-level semantics. GLiNER followed with a Total F1 of 0.60, suggesting that it struggled more with semantic variability; and SciBERT remained competitive with Total F1 of 0.59, showing stable results across tasks despite its limited adaptability to unseen domains.

5.1 Effect of Joint Loss

Across all models, training with joint loss improved both in-distribution and out-of-distribution performance (Table 1). This indicates that joint loss might allow models to use interdependencies between entity recognition and relation extraction more effectively.

5.2 Error Analysis

All models experienced a notable performance decline from Phase 1 to Phase 2, underscoring the challenge of generalizing to out-of-distribution data. GLiNER, DeBERTa v3, and SciBERT dropped from total F1 scores of 0.88, 0.83, and 0.79 to 0.60, 0.69, and 0.59, respectively, corresponding to reductions of approximately 0.20–0.25. This drop was especially pronounced in relation extraction, where domain shifts introduced unfamiliar entity formats, longer and more nested mentions (e.g., *Stata Statistical Software: Release 13*), and cross-clause relations that proved difficult to capture. SciBERT was most affected, suggesting that domain-specific pretraining alone is insufficient to guarantee robustness across scientific subdomains. A detailed entity- and relation-level analysis in Table 2 and 3 shows that GLiNER and DeBERTa v3 performed well on frequent and syntactically unambiguous entities such as URL, SoftwareCoreference, and OperatingSystem, where the high number of training examples and clear structure provided strong learning signals. They also handled common relation types like Citation_of and Developer_of with high accuracy. However, notable differences emerged in semantically complex and low-resource categories. DeBERTa v3 outperformed GLiNER on rare entities like Extension and its associated relation Extension_of, likely due to its stronger contextual representations and attention mechanisms. Conversely, GLiNER performed better on PlugIn and PlugIn_of, that might be indicating advantages of span-based architectures in handling regular syntactic patterns. Both models struggled with underrepresented or domain-specific relations such as License_of, AlternativeName_of, and Specification_of, where F1 scores dropped to zero, highlighting shared limitations in handling class imbalance and semantic drift.

In addition, an important constraint across all models seemed to be their reliance on sentence-level inputs, which prevented them from resolving longer-range dependencies or cross-sentence relationships. This restricted their ability to fully capture the context necessary for accurate entity and relation extraction in complex scholarly texts. Overall, these findings suggest that improving out-of-distribution generalization requires stronger pretraining objectives, more balanced annotation schemes, and model architectures capable of discourse-level reasoning.

Model	Phase 1 In-Distribution Validation Set							Phase 2 Out-of-Distribution Test Set						
	Total F1	Entity			Relation			Total F1	Entity			Relation		
		F1	P	R	F1	P	R		F1	P	R	F1	P	R
Disjoint Loss														
GLiNER	0.78	0.77	0.77	0.78	0.80	0.81	0.81	0.59	0.61	0.60	0.69	0.57	0.61	0.63
DeBERTa v3	0.80	0.78	0.77	0.80	0.81	0.81	0.84	0.59	0.63	0.60	0.70	0.56	0.59	0.60
SciBERT	0.75	0.79	0.79	0.79	0.72	0.68	0.77	0.53	0.56	0.59	0.59	0.50	0.59	0.52
Joint Loss														
GLiNER	0.88	0.90	0.87	0.94	0.85	0.88	0.85	0.60	0.66	0.65	0.73	0.53	0.58	0.56
DeBERTa v3	0.83	0.83	0.84	0.84	0.82	0.83	0.83	0.69	0.79	0.74	0.84	0.62	0.62	0.63
SciBERT	0.79	0.85	0.84	0.87	0.73	0.72	0.74	0.59	0.61	0.55	0.68	0.58	0.48	0.72

Table 1: Performance metrics for selected models in Phase 1 and Phase 2.

Entity Recognition (F1)		
Entity Type	**GLiNER**	**DeBERTa v3**
Application	0.765	0.732
Citation	0.837	0.903
Developer	0.667	0.667
PlugIn	0.435	0.346
Version	0.210	0.794
Extension	0.133	0.400
Release	0.667	0.727
URL	1.000	1.000
Abbreviation	0.529	0.750
ProgrammingEnvironment	0.957	0.936
OperatingSystem	1.000	1.000
SoftwareCoreference	1.000	1.000
AlternativeName	0.457	0.769

Table 2: Entity recognition F1 scores for GLiNER and DeBERTa v3.

Relation Extraction (F1)		
Relation Type	**GLiNER**	**DeBERTa v3**
Developer_of	0.650	0.652
Citation_of	0.682	0.683
Version_of	0.573	0.615
PlugIn_of	0.647	0.579
URL_of	0.787	0.595
Abbreviation_of	0.640	0.667
Release_of	0.522	0.705
Extension_of	0.250	0.545
AlternativeName_of	0.000	0.500
License_of	0.000	0.000
Specification_of	0.000	0.000

Table 3: Relation extraction F1 scores for GLiNER and DeBERTa v3.

6 Conclusion

We evaluate joint NER and RE models for extracting software mentions and relations in scientific texts, with a focus on out-of-distribution generalization. Our results show that a shared loss objective consistently boosts performance across architectures, indicating that multitask learning benefits not only from architectural integration but also from coupled optimization. This joint loss approach reduces error propagation and enhances robustness, making it a simple yet effective strategy for structured scientific information extraction.

Limitations

While our study offers valuable insights into joint NER and RE model generalization, it has limitations. First, restricting inputs to the sentence level may hinder models from capturing broader context or long-range dependencies, and future work should explore paragraph- or document-level modeling. Second, baseline Gemini and LLaMA were evaluated only in the first phase and due to low results were excluded from further training; adapting them to the domain may yield better results. Third, the limited size of the SOMD 2025 dataset may reduce the effectiveness of large models, especially for rare entities and relations.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context,

and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Stefan Dietze, Frank Krüger, and Saurav Karmarkar. 2024. SOMD: Software mention detection in scholarly publications. In *Proceedings of the International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, St. Julians, Malta. Springer.

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. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

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.

Ana-Maria Istrate, Joshua Fisher, Xinyu Yang, Kara Moraw, Kai Li, Donghui Li, and Martin Klein. 2024. Scientific software citation intent classification using large language models. In *Proceedings of the International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, pages 80–99. Springer Nature Switzerland, Cham.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. SciREX: A challenge dataset for document-level information extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online. Association for Computational Linguistics.

AmeerAli Khan, Qusai Ramadan, Cong Yang, and Zeyd Boukhers. 2024. Falcon 7b for software mention detection in scholarly documents. In *Proceedings of the International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, 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 *Proceedings of the International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, pages 247–256. Springer.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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.

Wolfgang Otto, Sharmila Upadhyaya, and Stefan Dietze. 2024. Enhancing software-related information extraction via single-choice question answering with large language models. In *Proceedings of the International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, 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.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

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 and Knowledge Management (CIKM)*, pages 4574–4583, New York, NY, USA. Association for Computing Machinery.

Thuy Nguyen Thi, Anh Nguyen Viet, Thin Dang Van, and Ngan Luu-Thuy Nguyen. 2024. Software mention recognition with a three-stage framework based on BERTology models at SOMD 2024. In *Proceedings of the International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2024)*, pages 257–266. Springer.

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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Deming Ye, Yankai Lin, Peng Li, and Maosong Sun. 2022. 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.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2024. GLiNER: Generalist model for named entity recognition using bidirectional transformer. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5364–5376, Mexico City, Mexico. Association for Computational Linguistics.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and 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*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

A Appendix

Model	Phase 1 In-Distribution Validation Set							Phase 2 Out-of-Distribution Test Set						
	Total F1	Entity			Relation			Total F1	Entity			Relation		
		F1	P	R	F1	P	R		F1	P	R	F1	P	R
BERT Uncased	0.67	0.75	0.73	0.79	0.60	0.59	0.61	–	–	–	–	–	–	–
Llama 3 8b Finetune	0.66	0.63	0.62	0.65	0.68	0.72	0.66	–	0.52	0.54	0.53	–	–	–
Gemini 2 Zero-Shot	–	0.39	0.37	0.44	–	–	–	–	–	–	–	–	–	–

Table 4: Performance metrics for the additional models we tested in Phase 1 and 2. Not all experiments were conducted for all models.

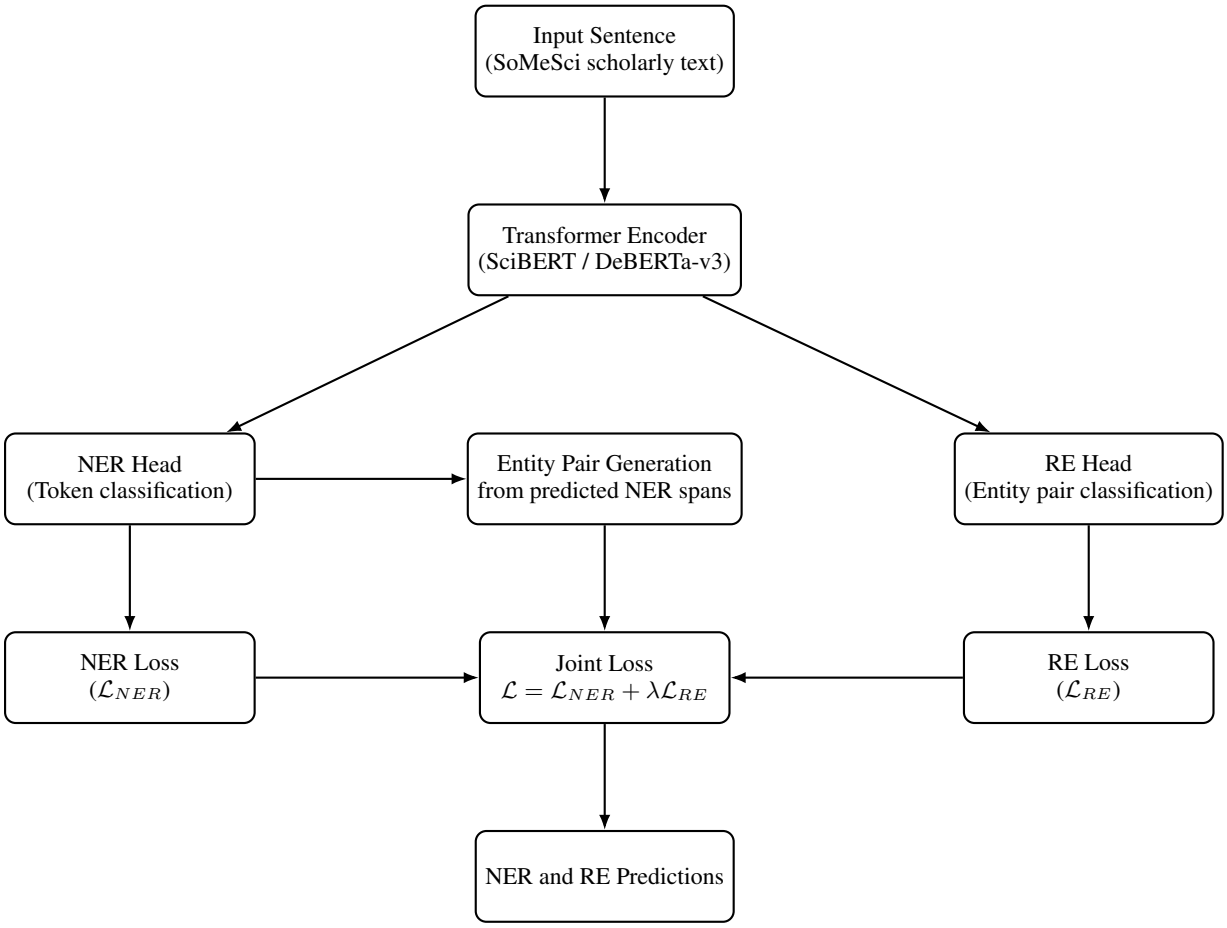

Figure 1: Architecture and data flow of our joint NER and RE model. The Transformer Encoder processes input tokens, which are used by two task-specific heads. NER predictions are used to generate entity pairs for RE. Each task contributes to the total loss, enabling joint optimization.