Evaluating LLMs for Named Entity Recognition in Scientific Domain with Fine-Tuning and Few-Shot Learning

Davide Buscaldi[1], Danilo Dessi[2], Francesco Osborne[3,4], Davide Piras[5] and Diego Reforgiato Recupero[5,*]

[1]Laboratoire d'Informatique de Paris Nord, Sorbonne Paris Nord University, Paris, France
[2]Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, UAE
[3]Knowledge Media Institute, The Open University, Walton Hall, Kents Hill, Milton Keynes, MK76AA, United Kingdom
[4]Department of Business and Law, University of Milano Bicocca, Milan, Italy
[5]Department of Mathematics and Computer Science, Via Ospedale 62, Cagliari, 09121, Italy

Abstract

Entity extraction is a crucial step in constructing Knowledge Graphs (KGs) from natural language text. In the scientific domain, Named Entity Recognition (NER) is widely used to analyze research papers and facilitate the generation of knowledge graphs that capture research concepts. Given the vast scale of contemporary research output, this task necessitates automated pipelines to maintain efficiency while ensuring the quality of the extracted knowledge. Large Language Models (LLMs) present a promising solution to this challenge. As such, this paper explores the effectiveness of LLMs for NER in scientific texts, using the SciERC dataset as a benchmark. Specifically, it evaluates different LLM architectures, including encoder-only, decoder-only, and encoder-decoder models, to identify the most effective approach for NER in the computer science domain. By examining the strengths and limitations of each model type, this study aims to provide deeper insights into the applicability of LLMs for entity extraction, ultimately improving the construction of domain-specific KGs.

Keywords

Large Language Models, Named Entity Recognition, Knowledge Graph Construction, Scholarly Domain

1. Introduction

Entity extraction is a key step in constructing knowledge graphs (KGs) from natural language text. A fundamental technique for this task is named entity recognition (NER), a natural language processing (NLP) method that identifies text spans referring to real-world entities [1] and assigns them to specific categories. In the scientific domain, NER plays a key role in processing research papers and facilitating the generation of KGs that encapsulate research concepts. As a result, NER is an essential component in scientific KG construction pipelines [2, 3, 4] and is widely employed for the semantic indexing of documents [5]. Large Language Models (LLMs) have been achieving significant success across a wide range of tasks. Their ability to understand general-purpose language is attributed to their extensive parameters, which are trained on vast amounts of data. The rise of models such as BERT [6], GPT [7], and T5 [8] has revolutionized NLP by providing robust tools that can be fine-tuned for specific tasks, including NER. These models leverage deep learning techniques to capture nuanced patterns in text, thus improving the performance of various NLP applications.

In this work, we study different types of LLMs on a NER task within the scholarly domain using the SciERC benchmark dataset [9]. Specifically, we investigate the performance of encoder-only models, encoder-decoder models, and decoder-only models in recognizing and classifying entities in scientific texts. The primary objectives of this comparison are to 1) determine which model type performs best on NER tasks within specialized domains and 2) understand the strengths and weaknesses of each

SemTech4STLD ESWC 2025, June 01, 2025, Portoroz, Slovenia
*Corresponding author.
davide.buscaldi@lipn.univ-paris13.fr (D. Buscaldi); ddessi@sharjah.ac.ae (D. Dessi); francesco.osborne@open.ac.uk (F. Osborne); d.piras38@studenti.unica.it (D. Piras); diego.reforgiato@unica.it (D. R. Recupero)

model type in handling domain-specific language. To achieve these goals, we focused on three different strategies: fine-tuning, zero-shot, and few-shot learning. The reason behind incorporating zero-shot and few-shot approaches alongside fine-tuning is to test the generalization capabilities of certain models. Generalization capabilities enable the possibility to apply LLMs on all those tasks that do not provide sufficient training data, making LLMs a valuable tool for several domains. Achieving good results with these techniques indicates that a model has human-like capabilities and can solve the task by leveraging its pre-existing knowledge without the need for extensive additional training. Our analysis aims to provide insights into the suitability of different LLM architectures for NER tasks for the automatic detection of scientific entities from natural language text.

The contribution of this paper is twofold. First, we evaluate the performance of three LLMs on a NER task in the scientific domain. Second, we provide insights into the architecture of these models and their effectiveness in addressing NER tasks within this domain. All the source code used for the analysis reported in this paper can be found at https://github.com/dpiras38/scierc_notebooks_ner.

2. Related Work

LLMs have been recently explored for NER applications in scientific writing. However, the task has proven to be difficult due to the writing style, nuances, and technical vocabulary used in academic texts.

Luan et al. (2018) [9] introduced the SciERC dataset, comprising 500 annotated scientific abstracts with entities, relationships, and coreference clusters. They proposed a multi-task model for jointly NER, relation extraction, and coreference resolution. This approach has proved to be effective for datasets like SciERC, where entities and relationships are closely linked. A later variation [10] replaced the multi-task strategy with entity-type prediction and incorporated cross-sentence context to enrich the input for the model. BERT (Bidirectional Encoder Representations from Transformers) [6] revolutionized NLP by leveraging a new architecture [11] that interprets word semantics based on their context. SciBERT [12] was one of the first adaptations for scientific texts, trained on research papers with a specialized vocabulary, SciVocab. In particular, only 42% of its terms overlap with the BERT's, highlighting existing differences between common and scientific language. SciBERT has since outperformed general models in entity and key term recognition on datasets like SciERC and BC5CDR [12].

Over the past five years, LLMs have significantly advanced the state of the art in NLP and information extraction across various domains [13, 14, 15, 16, 17]. In particular, several decoder-only models have demonstrated exceptional performance in tasks related to academic text, including research paper classification [18], citation recommendation [19, 20], automatic construction of research topics ontologies [21, 22], and literature review generation [23]. Conversely, encoder-decoder models, such as T5, have shown excellent performance in tasks such as molecule captioning [24] and scientific question answering via natural language to SPARQL translation [25]. In this paper, we compare these models on the task of NER applied to research papers for knowledge graph generation.

Indeed, the scientific and academic communities, which traditionally relied only on relatively simple knowledge organisation systems [26] for structuring research topics and indexing papers, have witnessed a substantial expansion in the use of KGs, which can accommodate a wider range of concepts and connections.

Several KGs have been developed using language models and transformer-based architectures. These efforts span fully automated pipelines [27, 28] as well as hybrid methodologies that incorporate human expertise into the construction and curation process [29, 30]. The resulting graphs are often further enhanced through link prediction techniques [31], which improve their completeness and semantic coherence by inferring missing relationships. Notable examples of these KGs include SemOpenAlex [32], AIDA-KG [3], OpenCitations [33], ORKG [34], AI-KG [35], CS-KG [36, 37], and Nano-publications [38]. These KGs enable a wide range of domain-specific applications, such as academic writing support [39], verification and completion of research claims [40], automated generation of research hypotheses [41], enriching metadata of scientific books [42], and the creation of specialised conversational agents [43**?** , 44], among others. Developing robust and accurate models for NER in scientific papers is crucial for

constructing and refining these knowledge bases.

3. The Used Dataset: SciERC

The SciERC dataset, introduced in 2018 [9], stands as a pivotal resource for generating models aimed at the extraction of entities, relationships, and coreference resolution within scientific texts. Models based on this dataset have been employed in KG construction with remarkable performances [2]. SciERC is manually crafted by annotating research articles from 12 major workshops and conferences, including the *Annual Meeting of the Association for Computational Linguistics (ACL)* and the *Empirical Methods in Natural Language Processing conference (EMNLP)*. SciERC diverged from earlier datasets by concentrating exclusively on computer science research articles, filling a significant gap in the field. The dataset comprises annotations covering: 6 types of entities (*Task*, *Metric*, *Method*, *Generic*, *OtherScientificTerm*, and *Material*), relationships between entities (`used-for`, `feature-of`, `hyponym-of`, `part-of`, `compare`, and `conjunction`), and coreference (identification of different text spans that refer to the same entity).

At the time of release, SciERC is pre-divided into training, validation, and test sets. Its data contains both entities and relationships. It can be downloaded in both raw and processed formats (tokenized and JSON) from http://nlp.cs.washington.edu/sciIE/. For the analysis presented in this paper, we only use the entities and their types, and leave the analysis on relationship extraction and coreference resolution tasks to future endeavors. Furthermore, since in SciERC an entity span can include other sub-entities (e.g., the entity *natural language processing* contains the entity *natural language*, we have pre-processed the dataset so that each entity is associated with the largest corresponding span, and all the other entities within that span are removed.

4. Large Language Models

In this section, we will illustrate the LLMs that we have used in this paper.

SciBERT. SciBERT [12] is built upon the BERT architecture [6], and uses an encode-only architecture. It is pre-trained on a corpus of 1.14 million papers randomly sampled from SemanticScholar (https://www.semanticscholar.org/). This corpus consists of 82% biomedical domain data and 18% computer science domain data. SciBERT leverages domain-specific knowledge well, making it a valuable tool for researchers for NLP tasks [12] in specialized domains.

Mistral. Mistral, introduced in [45], is a decoder-only model that balances high performance and efficiency in LLMs. Mistral builds on top of the transformer architecture and introduces key innovations when compared to other models such as LLama, including: i) Sliding Window Attention, which improves long-sequence processing by utilizing a window that allows the model to better exploit contextual information, ii) Rolling Buffer Cache, which optimizes the model memory by storing recently encountered tokens, and iii) Pre-fill and Chunking that split an input prompt into smaller segments and pre-compiles the cache, thus enabling fast processing of small chunks from a larger data. Mistral's performance has been tested against leading competitors such as LLaMA on various tasks, including commonsense reasoning, reading comprehension, code understanding, world knowledge, and math [45].

T5. T5, which stands for "Text-To-Text Transfer Transformer", is a model introduced by Google Research in [8]. It is based on the encoder-decoder architecture of Transformers and tackles tasks where the input and the output are text strings. T5 uses a sequence-to-sequence framework, which consists of an encoder that reads the input text and a decoder that generates the output text. Each task is formulated as a text transformation problem. T5 was pre-trained on the C4 corpus, which has been commonly used in the pre-training of other models like LLaMA.

5. LLMs Learning Methodologies

This section outlines the learning methodologies that we have employed within the LLMs.

5.1. Zero-Shot Learning

The zero-shot setting was employed with Mistral 7B model. Fig. 3 (in Appendix) shows the prompt used for the zero-shot setting. The prompt instructs the LLM about the task and the expected entity types, and requires the LLM to provide the answer in a structured format that can be processed automatically. This approach is not possible with T5 and SciBERT models due to their architectural differences.

After collecting all responses, a post-processing step was applied to remove errors unrelated to classification that could impact the results. These errors primarily involved inconsistencies in response formatting and, in some cases, word repetition. Additionally, after an empirical analysis, we limited the generated answers to 70 tokens to reduce the possibility of hallucination.

5.2. Few-Shot Learning

Few-shot learning provides the LLMs with a few examples to learn how to better solve a task. In this work, we focused on the variants with one example, referred to as one-shot, and three examples, referred to as three-shots. An example of a one-shot prompt is shown in Fig. 4 (in the Appendix). The process for developing this task was very similar to the zero-shot process, with the only difference being in the prompt used. For the one-shot task, we chose an example that represented the maximum number of diverse entities possible, while for the three-shot, three examples were randomly selected.

5.3. Fine Tuning

In this work, we fine-tuned Mistral 7B, T5, and SciBERT for the proposed task using the SciERC dataset. The parameters used for each model are detailed in the sections above.

Mistral 7B. We used a quantized 4-bit version provided by *TheBloke* on the Hugging Face Hub[1] and loaded it using the HuggingFace *AutoModelForCausalLM* class. Each record in the training and validation set was incorporated into a predefined prompt similar to a one-shot setup. However, for fine-tuning, we add the list of entities in the format *"Type of Entity: Entity;"* after the model response, as illustrated in Fig. 5 (in the Appendix). Fine-tuning was conducted using the *Trainer* Library[2] with the following parameters: learning_rate = 2.5e - 5, gradient_checkpointing=True, optim='paged_adamw_32bit' and num_train_epochs = 2. As for the previous prompts, we limited the answer to 70 tokens.

T5. For T5[3], we utilized the models directly provided by Google via the Hugging Face Hub, selecting the Base version to balance performance and hardware efficiency. This model was lightweight enough to run on our available hardware without requiring quantization while still being powerful enough for our tasks. We loaded the model using the *AutoModelForSeq2SeqLM*[4] library and applied a pre-configured prompt for each sample in the Training and Validation sets. Compared to the one used for Mistral, this prompt was significantly simpler. Figures 1 and 2 respectively illustrate a prompt used for inference and the corresponding response generated by T5. Additionally, the contents illustrated in both figures have been used together for the fine-tuning of T5.

```
NER: Color is known to be highly discriminative for many object recognition tasks , but is
difficult to infer from uncontrolled images in which the illuminant is not known
```

Figure 1: Example inference prompt for T5.

```
Material: uncontrolled images; Task: object recognition tasks;
OtherScientificTerm: illuminant, object;
```

Figure 2: Example of response for T5.

[1]https://huggingface.co/TheBloke/Mistral-7B-v0.1-GPTQ
[2]https://huggingface.co/docs/transformers/main_classes/trainer
[3]https://huggingface.co/google-t5/t5-base
[4]https://huggingface.co/docs/transformers/model_doc/auto#transformers.AutoModelForSeq2SeqLM

Language Model	Precision	Recall	F1 Score
Mistral 7B (Fine Tuning)	60.23%	56.56%	58.34%
T5 Base (Fine Tuning)	61.25%	62.17%	61.71%
SciBERT (Fine Tuning)	68.34%	71.15%	69.72%
Mistral 7B (Zero Shot)	3.93%	2.41%	2.90%
Mistral 7B (One Shot)	14.30%	17.29%	15.70%
Mistral 7B (Three Shot)	15.39%	19.00%	17.00%

Table 1
Performance results of the different indicated models.

SciBERT. Regarding SciBERT[5], we used the uncased version. The fine-tuning process was developed as follows. First, the model was loaded, through the class *AutoModelForTokenClassification*, then the corresponding tokenizer was created using the class *AutoTokenizer*, both from the *Transformers* library. For training, we used learning_rate = 5e-5 and num_train_epochs = 6. Finally, we instantiated a trainer using the class *Trainer* from *Transformers* library, and we proceeded with the training process.

6. Results

This section describes the evaluation setting as well as the outcome of our analysis.

6.1. Evaluation

To evaluate the different models, an entity is deemed correct if both its span boundaries and category are accurately identified. We defined: True Positives (TP): elements that are present in the predictions which are in the test set. False Positives (FP): elements present in the model's predictions but absent or different in the test set (in either span boundaries or category). False Negatives (FN): elements present in the test set but absent or different in the model's predictions.

6.2. Outcome Analysis

Table 1 reports the precision, recall, and F1 scores for each model. Surprisingly, the smallest model SciBERT outperforms the other models. This can be due to its pre-training on scientific data that enables a better understanding and recognition of the vocabulary, terminology, and context specific used in scientific texts.

The encoder-decoder and decoder-only models produced satisfactory results but were unable to surpass SciBERT. Among them, the encoder-decoder model T5 outperformed the decoder-only models, with an F1 score of 61.71% (see T5 Base (Fine Tuning)). In comparison, the decoder-only models scored 58.34% for Mistral as the best results in terms of F1 score.

This can be due to the architecture of the models: T5 has an encoder enabling it to use bidirectional context, whereas Mistral, being decoder-only, can only process text in a unidirectional manner, limiting its ability to understand the full context surrounding a word. Finally, prompts are much simpler on T5 than on Mistral, and prompts can significantly impact performance. Decoder-only models may not interpret and follow complex or nuanced prompts as effectively. Another important point to highlight is that, despite the significant difference in the number of parameters, between the decoder-only models and the encoder-decoder model T5, the latter still produced better results, demonstrating its quality. Specifically, the decoder-only models Mistral have 7 billion parameters while the encoder-decoder model has 220 million parameters (T5 Base). The effectiveness of a model greatly depends on its architecture and the decisions made during the pre-training and fine-tuning phases. Beyond the numerical scores, it is noteworthy that the results of the N-shot tasks significantly improve with an increase in examples provided in the prompt, going from a maximum value of 3.0% achieved by Mistral for the zero-shot task

[5]https://github.com/allenai/scibert

to 17% with three examples, again achieved by Mistral. Finally, it is evident from the results that LLMs cannot directly be used off the shelf and that fine-tuning is necessary to capture domain peculiarities and perform a satisfactory NER.

7. Conclusions

This work presented a comprehensive evaluation of various LLMs on the task of NER using the SciERC dataset as a benchmark. The results demonstrate that fine-tuning LLMs significantly enhances their performance on NER tasks. Among the models tested, SciBERT achieved the highest performance with an F1 score of 69.72%. These results highlight the importance of domain-specific pre-training in achieving better performance in scientific NER tasks. Besides, the decoder-only models performed worse than any other model, even with fine-tuning, demonstrating that model architecture and pre-training are critical performance factors. The results of zero-shot and few-shot learning approaches suggest that these models should not be employed for entity detection, confirming insights already detected in similar scenarios for KG construction [46]. Even the best few-shot approaches could not match the performance of fine-tuning, highlighting the challenges these models face when applied to scientific NER tasks without extensive training. In conclusion, our analysis suggests that i) SciBERT is still a reliable and valid option for constructing KGs in the computer science domain, and ii) specialized models can still be a better option for niche tasks. The insights of this work will be leveraged to improve the construction pipeline SCICERO [2] and to generate newer versions of the CS-KG [36].

Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT, Grammarly in order to: Grammar and spelling check, paraphrase and reword. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] B. Jehangir, S. Radhakrishnan, R. Agarwal, A survey on named entity recognition—datasets, tools, and methodologies, Natural Language Processing Journal 3 (2023) 100017.

[2] D. Dessí, F. Osborne, D. R. Recupero, D. Buscaldi, E. Motta, Scicero: A deep learning and nlp approach for generating scientific knowledge graphs in the computer science domain, Knowledge-Based Systems 258 (2022) 109945.

[3] S. Angioni, A. Salatino, F. Osborne, D. R. Recupero, E. Motta, Aida: A knowledge graph about research dynamics in academia and industry, Quantitative Science Studies 2 (2021) 1356–1398.

[4] M. Zloch, D. Dessì, J. D'Souza, et al., Research knowledge graphs: the shifting paradigm of scholarly information representation, in: The Semantic Web - 22nd International Conference, ESWC 2025, Springer, 2025.

[5] M. Ehrmann, A. Hamdi, E. L. Pontes, M. Romanello, A. Doucet, Named entity recognition and classification in historical documents: A survey, ACM Computing Surveys 56 (2023) 1–47.

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, North American Chapter of the Association for Computational Linguistics (NAACL) (2019).

[7] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving Language Understanding by Generative Pre-Training, Preprint (2018).

[8] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, arXiv (2019).

[9] Y. Luan, L. He, M. Ostendorf, H. Hajishirzi, Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction, Empirical Methods in Natural Language Processing (EMNLP) (2019).

[10] Z. Zhong, D. Chen, A Frustratingly Easy Approach for Entity and Relation Extraction, North American Chapter of the Association for Computational Linguistics (NAACL) (2021).
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, Neural Information Processing Systems (NeurIPS) (2017).
[12] I. Beltagy, K. Lo, A. Cohan, SciBERT: A Pretrained Language Model for Scientific Text, International Joint Conference on Natural Language Processing (IJCNLP) (2019).
[13] J. A. Omiye, H. Gui, S. J. Rezaei, J. Zou, R. Daneshjou, Large language models in medicine: the potentials and pitfalls: a narrative review, Annals of internal medicine 177 (2024) 210–220.
[14] E. Motta, F. Osborne, M. M. Pulici, A. Salatino, I. Naja, Capturing the viewpoint dynamics in the news domain, in: International Conference on Knowledge Engineering and Knowledge Management, Springer, 2024, pp. 18–34.
[15] C. W. Kosonocky, C. O. Wilke, E. M. Marcotte, A. D. Ellington, Mining patents with large language models elucidates the chemical function landscape, Digital Discovery 3 (2024) 1150–1159.
[16] K. Yang, T. Zhang, Z. Kuang, Q. Xie, J. Huang, S. Ananiadou, Mentallama: interpretable mental health analysis on social media with large language models, in: Proceedings of the ACM Web Conference 2024, 2024, pp. 4489–4500.
[17] A. Chessa, G. Fenu, E. Motta, F. Osborne, D. R. Recupero, A. Salatino, L. Secchi, Data-driven methodology for knowledge graph generation within the tourism domain, IEEE Access 11 (2023) 67567–67599.
[18] A. Cadeddu, A. Chessa, V. D. Leo, G. Fenu, E. Motta, F. Osborne, D. R. Recupero, A. Salatino, L. Secchi, Optimizing tourism accommodation offers by integrating language models and knowledge graph technologies, Information 15 (2024) 398.
[19] D. Buscaldi, D. Dessí, E. Motta, M. Murgia, F. Osborne, D. R. Recupero, Citation prediction by leveraging transformers and natural language processing heuristics, Information Processing & Management 61 (2024) 103583.
[20] Y. Zhang, Y. Wang, K. Wang, Q. Z. Sheng, L. Yao, A. Mahmood, W. E. Zhang, R. Zhao, When large language models meet citation: A survey, arXiv preprint arXiv:2309.09727 (2023).
[21] H. Babaei Giglou, J. D'Souza, S. Auer, Llms4ol: Large language models for ontology learning, in: International Semantic Web Conference, Springer, 2023, pp. 408–427.
[22] T. Aggarwal, A. Salatino, F. Osborne, E. Motta, Large language models for scholarly ontology generation: An extensive analysis in the engineering field, arXiv preprint arXiv:2412.08258 (2024).
[23] F. Bolanos, A. Salatino, F. Osborne, E. Motta, Artificial intelligence for literature reviews: Opportunities and challenges, Artificial Intelligence Review 57 (2024).
[24] C. Edwards, T. Lai, K. Ros, G. Honke, K. Cho, H. Ji, Translation between molecules and natural language, arXiv preprint arXiv:2204.11817 (2022).
[25] J. Lehmann, A. Meloni, E. Motta, F. Osborne, D. R. Recupero, A. A. Salatino, S. Vahdati, Large language models for scientific question answering: An extensive analysis of the sciqa benchmark, in: European Semantic Web Conference, Springer, 2024, pp. 199–217.
[26] A. Salatino, T. Aggarwal, A. Mannocci, F. Osborne, E. Motta, A survey on knowledge organization systems of research fields: Resources and challenges, Quantitative Science Studies (2025) 1–37.
[27] D. Dessì, F. Osborne, D. R. Recupero, D. Buscaldi, E. Motta, Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain, Future Generation Computer Systems 116 (2021) 253–264.
[28] L. Zhong, J. Wu, Q. Li, H. Peng, X. Wu, A comprehensive survey on automatic knowledge graph construction, ACM Computing Surveys 56 (2023) 1–62.
[29] S. Tsaneva, D. Dessì, F. Osborne, M. Sabou, Knowledge graph validation by integrating llms and human-in-the-loop, Information Processing & Management 62 (2025) 104145.
[30] A. Brack, A. Hoppe, M. Stocker, S. Auer, R. Ewerth, Analysing the requirements for an open research knowledge graph: use cases, quality requirements, and construction strategies, International Journal on Digital Libraries 23 (2022) 33–55.
[31] M. Nayyeri, G. M. Cil, S. Vahdati, F. Osborne, M. Rahman, S. Angioni, A. Salatino, D. R. Recupero, N. Vassilyeva, E. Motta, et al., Trans4e: Link prediction on scholarly knowledge graphs,

Neurocomputing 461 (2021) 530–542.
[32] M. Färber, D. Lamprecht, J. Krause, L. Aung, P. Haase, Semopenalex: The scientific landscape in 26 billion rdf triples, in: International Semantic Web Conference, Springer, 2023, pp. 94–112.
[33] M. Daquino, S. Peroni, D. Shotton, G. Colavizza, B. Ghavimi, A. Lauscher, P. Mayr, M. Romanello, P. Zumstein, The opencitations data model, in: International semantic web conference, Springer, 2020, pp. 447–463.
[34] M. Y. Jaradeh, A. Oelen, K. E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker, S. Auer, Open research knowledge graph: next generation infrastructure for semantic scholarly knowledge, in: Proceedings of the 10th international conference on knowledge capture, 2019, pp. 243–246.
[35] D. Dessì, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, H. Sack, Ai-kg: an automatically generated knowledge graph of artificial intelligence, in: The Semantic Web–ISWC 2020: 19th International Semantic Web Conference, Springer, 2020, pp. 127–143.
[36] D. Dessí, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, Cs-kg: A large-scale knowledge graph of research entities and claims in computer science, in: International Semantic Web Conference, Springer, 2022, pp. 678–696.
[37] D. Dessí, F. Osborne, D. Buscaldi, D. Reforgiato Recupero, E. Motta, Cs-kg 2.0: A large-scale knowledge graph of computer science, Scientific Data 12 (2025) 1–16.
[38] T. Kuhn, C. Chichester, M. Krauthammer, N. Queralt-Rosinach, R. Verborgh, G. Giannakopoulos, A.-C. N. Ngomo, R. Viglianti, M. Dumontier, Decentralized provenance-aware publishing with nanopublications, PeerJ Computer Science 2 (2016) e78.
[39] S. Brody, Scite, Journal of the Medical Library Association: JMLA 109 (2021) 707.
[40] A. Borrego, D. Dessi, I. Hernández, et al., Completing scientific facts in knowledge graphs of research concepts, IEEE Access 10 (2022) 125867–125880.
[41] A. Borrego, D. Dessì, D. Ayala, I. Hernández, F. Osborne, D. R. Recupero, D. Buscaldi, D. Ruiz, E. Motta, Research hypothesis generation over scientific knowledge graphs, Knowledge-Based Systems 315 (2025) 113280.
[42] A. A. Salatino, F. Osborne, A. Birukou, E. Motta, Improving editorial workflow and metadata quality at springer nature, in: The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18, Springer, 2019, pp. 507–525.
[43] R. Alonso, D. Dessí, A. Meloni, M. Murgia, D. R. Recupero, G. Scarpi, A seamless chatgpt knowledge plug-in for the labour market, IEEE Access (2024).
[44] L. Laranjo, A. G. Dunn, H. L. Tong, A. B. Kocaballi, J. Chen, R. Bashir, D. Surian, B. Gallego, F. Magrabi, A. Y. Lau, et al., Conversational agents in healthcare: a systematic review, Journal of the American Medical Informatics Association 25 (2018) 1248–1258.
[45] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, W. E. Sayed, Mistral 7B, arXiv (2023).
[46] L. Gan, M. Blum, D. Dessì, B. Mathiak, R. Schenkel, S. Dietze, Hidden entity detection from github leveraging large language models, volume 3894 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3894/dl4kg_paper4.pdf.

Appendix

A. Zero-Shot Learning

```
### Question: Identify and extract without rephrasing all entities that belongs to
the categories Method, Metric, Material, Task, OtherScientificTerm, Generic from the
following sentence in a list format like "Category: Entity":

 "In this paper , we present an unlexicalized parser for German which employs
smoothing and suffix analysis to achieve a labelled bracket F-score of 76.2 , higher
than previously reported results on the NEGRA corpus ."


These are the annotation guideline to extract entities:

Task are Applications, problems to solve, systems to construct;
Method are Methods , models, systems to use, or tools, components of a system;
Evaluation Metric are Metrics, measures, or entities that can express quality of a
system/method;
Material are Data, datasets, resources, Corpus, Knowledge base;
OtherScientificTerms are Phrases that are a scientific terms but do not fall into any
of the above classes;
Generic are General terms or pronouns that may refer to a entity but are not
themselves informative.

###Answer:
```

Figure 3: Prompt utilized for Zero-Shot Learning.

B. Few-Shot Learning

```
### Question: Identify and extract, without rephrasing, all entities that belong to the categories
Method, Metric, Material, Task, OtherScientificTerm, and Generic from the test sentence.
These are the annotation guidelines to extract entities:
- *Task:* Applications, problems to solve, systems to construct.
- *Method:* Methods, models, systems to use, or tools, components of a system.
- *Evaluation Metric:* Metrics, measures, or entities that can express the quality of a system/method.
- *Material:* Data, datasets, resources, Corpus, Knowledge base.
- *OtherScientificTerms:* Phrases that are scientific terms but do not fall into any of the above
classes.
- *Generic:* General terms or pronouns that may refer to an entity but are not themselves informative.

Below are some examples:

Input Sentence: "We show results from multi-modal super-resolution and face recognition experiments
across different imaging modalities , using low-resolution images as testing inputs and demonstrate
improved recognition rates over standard tensorface and eigenface representations"
Output Sentence: Task: multi-modal super-resolution and face recognition; OtherScientificTerm: imaging
modalities; Material: low-resolution images; Metric: recognition rates; Method: tensorface and
eigenface representations;

Test Sentence:"Recognition of proper nouns in Japanese text has been studied as a part of the more
general problem of morphological analysis in Japanese text processing -LRB- -LSB- 1 -RSB- -LSB- 2 -
RSB- -RRB- "
### Answer:
```

Figure 4: Prompt utilized for One-Shot Learning.

C. Fine Tuning

```
### Question: Identify and extract without rephrasing all entities that belongs to
the categories Method, Metric, Material, Task, OtherScientificTerm, Generic from the
following sentence in a list format like "Category: Entity":

"We introduce the multi-view color constancy problem , and present a method to
recover estimates of underlying surface re-flectance based on joint estimation of
these surface properties and the illuminants present in multiple images"

These are the annotation guideline to extract entities:

Task are Applications, problems to solve, systems to construct;
Method are Methods , models, systems to use, or tools, components of a system;
Evaluation Metric are Metrics, measures, or entities that can express quality of a
system/method;
Material are Data, datasets, resources, Corpus, Knowledge base;
OtherScientificTerms are Phrases that are a scientific terms but do not fall into any
of the above classes;
Generic are General terms or pronouns that may refer to a entity but are not
themselves informative.
### Answer: Task: multi-view color constancy problem; Task: estimates of underlying
surface re-flectance; OtherScientificTerm: surface properties; OtherScientificTerm:
illuminants; Generic: method;
```

Figure 5: Example prompt for the fine-tuning of Mistral 7B.