

Paper2ModelCard: A framework for the automated generation of model cards from research papers

Mateo Campaya-Pinto^{1,2}, Elvira Amador-Domínguez^{1,2,*}, Erick Cedeño¹, Daniel Garijo¹ and Sergio Zaballos-Herrera¹

¹*Ontology Engineering Group, Universidad Politécnica de Madrid*

²*Departamento de Sistemas Informáticos, ETSISI, Universidad Politécnica de Madrid*

Abstract

Model Cards were introduced in 2019 as a means to provide a human-readable, clear summarization of machine learning models, including key metadata such as license, biases or evaluation results. While the adoption of these cards has grown by the scientific community (being adopted by open platforms like HuggingFace) model cards are usually manually filled by authors when uploading their models, thus leading to missing or inaccurate information. Moreover, model cards are not semantically interoperable, since they are not compliant with any ontology or data model. This paper presents Paper2ModelCard (P2MC), a framework for end-to-end generation of model cards from research papers. P2MC integrates processing tools for correctly parsing the input PDF into a structured format (using existing tools like SciPDF and LightOnOCR-1B), which is then processed by an ensemble of models, each targeted towards efficiently extracting specific information from the input. The FAIR4ML ontology is used as the backbone to generate the model cards, in order to ensure their interoperability. Moreover, P2MC links the generated model cards to existing scientific knowledge graphs, such as SemOpenAlex or LinkedPapersWithCode.

Demo: <https://w3id.org/p2mc>

Code: <https://github.com/oeg-upm/p2mc>

Keywords

model cards, semantic interoperability, FAIR principles, information extraction, knowledge graphs

1. Introduction

In recent years, the number of new machine learning (ML) models has increased significantly. The volume of ML models available have underscored the need for transparency, reproducibility, and traceability. One of the first attempts at systematically profiling and describing ML models was proposed by Mitchell et al. [1], where the authors introduced the notion of model cards. Model cards provide key information on the purpose, training, and use of the model in a textual format. However, these model cards present two main drawbacks. First, they are manually created and maintained by their own creators, and therefore they are prone to contain errors or being incomplete. Secondly, they follow a plain text format, thus hindering their querying, interoperability, and reusability. Previous approaches have addressed related aspects of extracting and organizing information from scientific publications. CardGen [2] uses a retrieval-and-generation pipeline to produce model and data cards following HuggingFace documentation templates, but it does not provide ontology-aligned JSON-LD cards or links to external knowledge graphs. Dessì et al. [3] proposed an approach that uses natural language processing and machine learning techniques to extract entities and relationships from scientific abstracts and integrate them into a scholarly knowledge graph, but it does not generate model cards. AutoLLM-Card [4] extracts model names, licences and applications, but it is specifically designed for Large Language Models. Consequently, these approaches do not directly provide the combination

6th International Workshop on Scientific Knowledge: Representation, Discovery, and Assessment, Oct 2026, Bari, Italy

*Corresponding author.

†These authors contributed equally.

✉ mateo.campaya@upm.es (M. Campaya-Pinto); elvira.amador@upm.es (E. Amador-Domínguez); erick.cedeno@upm.es (E. Cedeño); daniel.garijo@upm.es (D. Garijo); s.zaballos@alumnos.upm.es (S. Zaballos-Herrera)

ORCID 0009-0001-0901-5562 (M. Campaya-Pinto); 0000-0001-6838-1266 (E. Amador-Domínguez); 0009-0004-4478-6890 (E. Cedeño); 0000-0003-0454-7145 (D. Garijo); 0009-0000-3795-2767 (S. Zaballos-Herrera)



© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

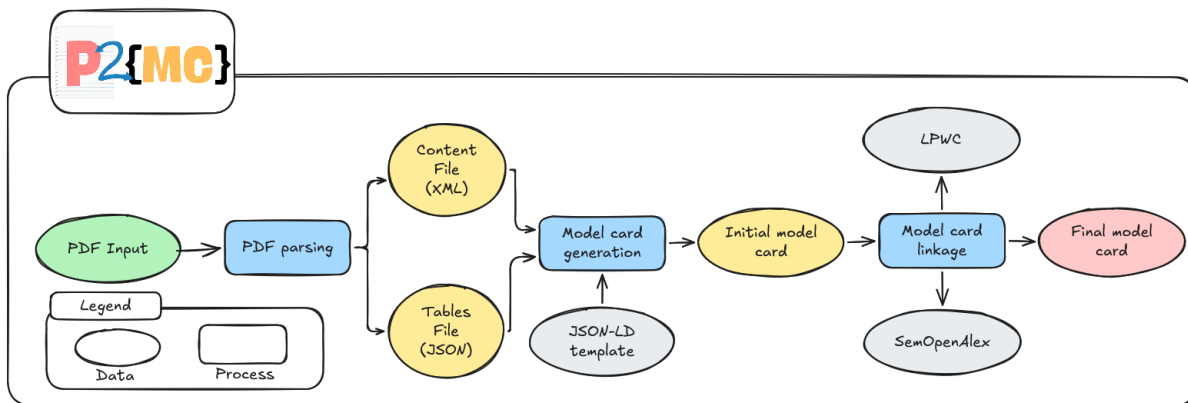


Figure 1: Overview of the P2MC framework. Green and red ellipses denote the input and output of the system, respectively. Yellow ellipses denote intermediate system results, while blue boxes depict the three main modules of the system.

of ontology-aligned model cards and links to external SKGs offered by P2MC. Moreover, the lack of alignment of metadata with existing ontologies significantly limits the compliance with respect to the FAIR principles [5]. Some recent efforts have been made in this direction, such as MLentory [6], a FAIR Digital Objects (FDO) register for ML models and their workflows.

This paper presents P2MC, a framework devised to automatically generate model cards from scientific publications. P2MC converts research articles in PDF format into normalized model cards in JSON-LD format following the FAIR4ML¹ ontology, thus ensuring their interoperability. Other alternatives for model card description exist, such as the work by Amith et al. [7]. However, this approach is devised specifically for biomedical research, and thus is not aligned with the application domain of P2MC. Additionally, P2MC enriches the generated model cards by automatically linking the detected entities within the paper to existing SKGs, such as SemOpenAlex [8] and LinkedPapersWithCode [9].

2. P2MC Overview and Architecture

Paper2ModelCard (or P2MC), is a web service that takes either a PDF document or its public URL as input, and generates the corresponding model card in JSON-LD format, following the FAIR4ML ontology². The system comprises three differentiated modules: PDF parsing, model card generation, and model card linkage. Figure 1 outlines the workflow of P2MC. The steps of the workflow are further detailed below.

PDF parsing P2MC receives either the content or the URL of a research paper in PDF format. Once the PDF file is downloaded, the system first parses the document into a machine-readable format. The input must constitute a scientific paper in which a ML model is presented. Two separate machine-readable files are generated, since research papers contain heterogeneous data, such as tables and/or figures, and therefore require tailored processing. In order to extract text-only content, a research-oriented version of GROBID [10] named SciPdf³ is used, which generates an XML file with all textual information contained within the paper. For table data, LightOnOCR-1B [11] is used, which generates a JSON file with all extracted tables in a Markdown format.

Model card generation Once the input is correctly parsed into machine-readable documents, the system automatically extracts the information required to generate a base model card. The pipeline uses template in JSON-LD format (see Appendix A) in order to organize the required information in a

¹<https://w3id.org/fair4ml/>

²<https://w3id.org/fair4ml/>

³https://github.com/titipata/scipdf_parser

structured, interoperable format. This template is based on the FAIR4ML ontology [12], a Schema.org[13] extension that systematically describes machine learning model metadata. A total of six attributes are automatically mined from the input paper, namely: experimental dataset, experimental metrics, model name, model category, evaluation tasks, and implementation URL. In order to extract the information required to fill each field, an ensemble of different ML models is used, each targeted at extracting a specific field from a different input combination. A preliminary benchmark on the fitness of different content and model approaches for model card generation was conducted as part of a previous research, whose results are reported in Appendix A. Table 1 outlines the pairing of input content and the model employed for extracting each of the fields considered in the template.

Field	Input content	Selected Model
<i>Dataset</i>	Tables Only	Qwen3:1.7B
<i>Metrics</i>	Tables Only	Qwen3:1.7B
<i>Model name</i>	Full Text	Llama3:8B
<i>Model category</i>	Paper Abstract	TF-IDF+SVC
<i>Tasks</i>	Paper Abstract	Llama3:8B
<i>Implementation URL</i>	Full Text	Llama3:8B

Table 1

Model and content selection per P2MC field. Small models are prioritized, due to low resource consumption.

After this process, an automatically extracted, initial version of the model card is generated in the JSON-LD serialization [14]. In this initial version of the model card, no URIs are included, only the information that can be automatically mined from the input source, either as part of the metadata extracted by SciPDF (for example, the authors and keywords) or as part of the paper content.

Model card linkage Once the base model card has been generated, the next steps of the pipeline link the model card to external Scientific Knowledge Graphs (SKGs). In order to adopt the FAIR principles, existing knowledge graphs are used for interoperability purposes. Three different URI spaces are used to identify each model card element:

- **LPWC:** Linked Papers with Code, or LPWC knowledge graph [9] is used to assign URIs to the datasets, metrics and tasks, if available.
- **SemOpenAlex:** For authors and publications, the SemOpenAlex knowledge graph is used [8] to assign a URI to these elements, if found.
- **P2MC:** This URI space is specifically devised for the P2MC tool, and is used for the identification of each generated model card, as well as to provide a URI to those elements that do not have any correspondence either in LPWC or SemOpenAlex. These include examples such as model categories, as well as upcoming datasets or tasks that are not contained within LPWC.

Besides assigning a URI to each model card entity, this module conducts an additional set of postprocessing steps, such as entity disambiguation (which is especially noticeable for task detection), reference publication linkage, or description generation. Then, each clean entity is searched using SPARQL queries on the corresponding SKG, retrieving the corresponding URIs when existing. If no valid URIs are found, a new URI in the P2MC space is created.

3. Demonstration

The P2MC framework is available as a web service at <https://p2mc.linkeddata.es/>, and can be also found at <https://github.com/oeg-upm/p2mc>. The P2MC web service runs on a 16GB RAM 4-core server, which implements a queuing service on RabbitMQ [15]. Since the server does not possess a GPU, the processing may be slow when the input paper comprises several pages with tables, as PDF and OCR parsing may take a significant amount of time. Nonetheless, the framework is devised to be easily adaptable and deployed into other dedicated machines.

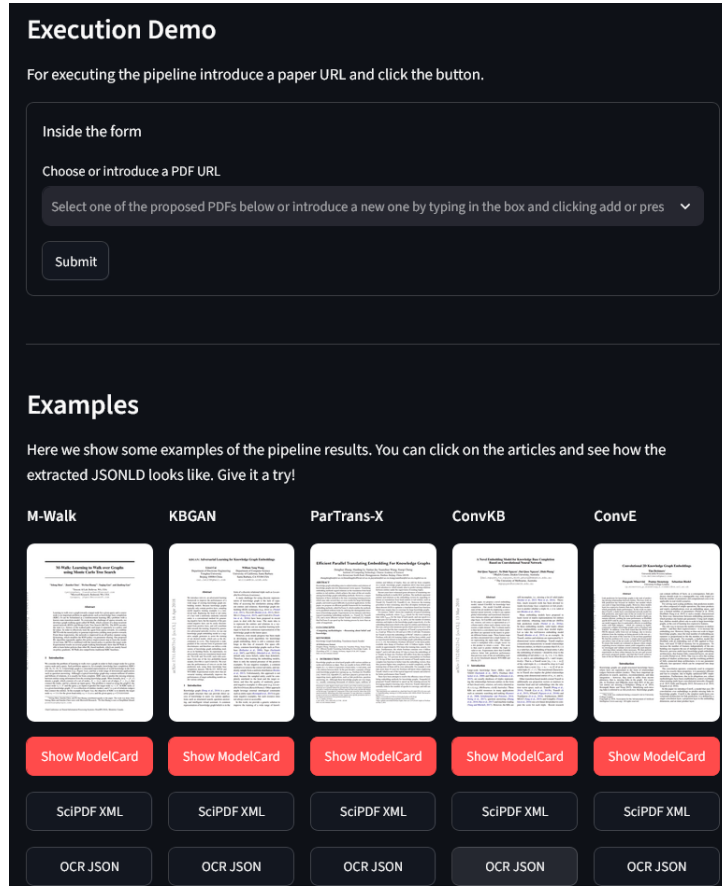


Figure 2: Overview of the main website of the P2MC demo. On the top part of the screen, users may request a model card generation from their own pdf files. On the lower part of the screen, several examples of pre-computed model cards are provided

In order to illustrate the reliability and usage of P2MC, several examples of model cards extracted from a set of sample papers are provided, as shown in Figure 2.

For each sample paper, a link to its original PDF file is provided on each paper image. Under each sample, three separate files are provided: the final model card, and both the processed textual input and table data. These intermediate files are included for further transparency on the behavior of the tool. Users can also select one of these sample papers and conduct its end-to-end processing, by choosing its URL from the dropdown selector. Moreover, users can either introduce a custom URL pointing to a PDF file, given that the PDF file is publicly published; or upload the PDF file as itself for processing.

Once the input data is submitted to the server, the background process queues the requests, notifying users when the final model card has been generated. After going through the parsing, generation and linking stages, a model card in JSON-LD format is provided. Figure 3 shows one of the sample model cards, corresponding to the KBGAN model, extracted from [16].

The generated model card contains the names of each element extracted for each field, along with the corresponding URI for each field. As shown in Figure 3, P2MC successfully links each detected entity to the corresponding SKG entity. For example, authors are correctly linked to their corresponding entity URIs in SemOpenAlex, while tasks are linked to the LPWC SKG.

4. Discussion and Limitations

While P2MC provides a fully working tool for the generation of base model cards in JSON-LD format, it still faces some challenges and limitations, which are described as follows:

Generated ModelCard for KBGAN:

```
"author": [
  {
    "@id": "https://semopenalex.org/author/A5088291592",
    "name": "Liwei Cai"
  },
  {
    "@id": "https://semopenalex.org/author/A5100702488",
    "name": "William Yang Wang"
  }
],
"dateCreated": "2026-07-17T10:19:43",
"mlTask": [
  {
    "@id": "https://linkedpaperswithcode.com/task/knowledge-graph-embedding",
    "name": "Knowledge Graph Embedding"
  },
  {
    "@id": "https://linkedpaperswithcode.com/task/link-prediction",
    "name": "Link Prediction"
  }
]
```

Figure 3: Sample model card for the KBGAN model, as described in its corresponding paper [16]. Image has been converted to high-contrasting mode to improve readability.

Data linkage As mentioned in Paragraph 2, different SKGs URI spaces are used to identify several model card elements such as datasets, tasks, metrics, authors and publications. Sometimes, the linking process tries to find a valid URI for one of these elements in an external SKG and fails. Several reasons can cause this incident, but it mostly relates to the absence of a valid URI for the input element in the queried SKG. In such cases, a custom URI in our own dataspace (<https://w3id.org/p2mc>) is assigned to the element. Since the SKG ecosystem is everchanging, the SKGs currently used by P2MC may become unavailable in the future. In order to deal with this potential issue, P2MC includes a fallback mechanism to ensure that every element within the model card includes a valid URI, replacing the obsolete URIs with new ones in our dataspace when needed. Table 2 reports the distribution of linked and unlinked entities obtained after processing a sample of 156 papers.

Field	Linked entities	Unlinked entities	Total entities
<i>Author IDs</i>	583	64	647
<i>Task IDs</i>	153	0	153
<i>Dataset IDs</i>	349	181	530

Table 2

Distribution of linked and unlinked entities after model card extraction of 156 papers.

Enriching model cards While P2MC generates a simple, yet descriptive, model card for the input source, there are still some relevant information that is still missing. Such is the case of the *bias* and *limitations* fields, which are often included in HuggingFace model cards. Similarly, the *intended use* and the *experimental setting* would further enrich the information available about the model.

Limitations P2MC, in its current state, presents some limitations that must be acknowledged. First, the extraction pipeline is devised to work with papers that introduce a model in the Machine Learning and, more specifically, for Knowledge Graph Embedding models. This implies that the behavior of the model in input papers from different resource areas has not been tested, and therefore the generated model cards may be prone to errors. In addition, the generated model cards are not stored nor published, and therefore cannot be queried by the users. Finally, it is worth noting that the current pipeline faces

some efficiency bottlenecks, specially when the input paper contains several tables.

5. Conclusions and Future Work

This paper describes P2MC, a demonstration service for converting research articles into structured model cards, following the FAIR4ML ontology and linking its results to existing SKGs such as LinkedPapersWithCode and SemOpenAlex.

Our work is tailored to ease the adoption of Model Cards, hence improving model comparison, findability and reusability. Future work will address the creation of domain-specific KGs, tailored towards improving the description of models in public platforms (e.g., models with low coverage but abundant in existing publications).

Acknowledgments

This work is part of the GAP-KGE project (M230020126A-EAD), which has been funded by the Comunidad de Madrid, as part of the grant call Línea de Actuación A, Doctores Emergentes. The work was also supported by the Predoctoral Grant PIPF-2024/COM-35132 of the Consejería de Educación, Ciencia y Universidades de la Comunidad de Madrid, Spain and the project FAIR2Adapt, under European Union's Horizon Europe framework programme with grant agreement No. 101188256 and DOI: 10.3030/101188256.

Declaration on Generative AI

The authors have not employed any Generative AI tools for the preparation of this manuscript.

References

- [1] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, T. Gebru, Model cards for model reporting, in: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* '19, Association for Computing Machinery, New York, NY, USA, 2019, p. 220–229. URL: <https://doi.org/10.1145/3287560.3287596>. doi:10.1145/3287560.3287596.
- [2] J. Liu, W. Li, Z. Jin, M. Diab, Automatic generation of model and data cards: A step towards responsible ai, 2024. URL: <https://arxiv.org/abs/2405.06258>. arXiv:2405.06258.
- [3] D. Dessì, F. Osborne, D. Reforgiato 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. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X2033003X>. doi:<https://doi.org/10.1016/j.future.2020.10.026>.
- [4] S. Tian, L. Han, G. Nenadic, Autollm-card: Towards a description and landscape of large language models, 2024. URL: <https://arxiv.org/abs/2409.17011>. arXiv:2409.17011.
- [5] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al., The fair guiding principles for scientific data management and stewardship, Scientific data 3 (2016).
- [6] D. Solanki, N. Quiñones, D. Rebholz-Schuhmann, L. J. Castro, Mlentry, an fdo registry for machine learning models, in: SWAT4HCLS, 2024. URL: <https://api.semanticscholar.org/CorpusID:275583310>.
- [7] M. Amith, L. Cui, K. Roberts, C. Tao, Application of an ontology for model cards to generate computable artifacts for linking machine learning information from biomedical research, in: Companion Proceedings of the ACM Web Conference 2023, WWW '23 Companion, Association

- for Computing Machinery, New York, NY, USA, 2023, p. 820–825. URL: <https://doi.org/10.1145/3543873.3587601>. doi:10.1145/3543873.3587601.
- [8] M. Färber, D. Lamprecht, J. Krause, L. Aung, P. Haase, Semopenalex: The scientific landscape in 26 billion rdf triples, 2023. URL: <https://arxiv.org/abs/2308.03671>. arXiv:2308.03671.
 - [9] M. Färber, D. Lamprecht, Linked papers with code: The latest in machine learning as an rdf knowledge graph, 2023. URL: <https://arxiv.org/abs/2310.20475>. arXiv:2310.20475.
 - [10] Grobid, <https://github.com/grobidOrg/grobid>, 2008–2025. arXiv:1:dir:dab86b296e3c3216e2241968f0d63b68e8
 - [11] S. Taghadouini, B. Aubertin, A. Cavallès, Lightonocr-1b: End-to-end and efficient domain-specific vision-language models for ocr, <https://huggingface.co/blog/lightonai/lightonocr>, 2025.
 - [12] L. J. Castro, D. Garijo, D. Rebholz-Schuhmann, D. Solanki, J. T. Ciuciu-Kiss, D. Katz, L. Eklund, G. Bharathy, R. D. A. F. T. Force, Fair4ml-schema, 2024. URL: <https://doi.org/10.5281/zenodo.14002310>. doi:10.5281/zenodo.14002310.
 - [13] Schema.org: The schema.org types and properties, <https://schema.org/>, ??? Last accessed: 2026-07-24.
 - [14] World Wide Web Consortium, JSON-LD 1.1: A JSON-based Serialization for Linked Data, W3C Recommendation, W3C, 2020. URL: <https://www.w3.org/TR/json-ld11/>.
 - [15] Broadcom, Inc., Rabbitmq documentation, 2026. URL: <https://rabbitmq.com>, accessed: July 21, 2026.
 - [16] L. Cai, W. Y. Wang, Kbgan: Adversarial learning for knowledge graph embeddings, 2018. URL: <https://arxiv.org/abs/1711.04071>. arXiv:1711.04071.

A. JSON-LD template

Listing 1: P2MC model template

```
{
  "@context": "https://w3id.org/fair4ml",
  "@id": "",
  "@type": "MLModel",
  "name": "",
  "description": "",
  "keywords": [],
  "version": "1.0.0",
  "author": [],
  "dateCreated": "",
  "mlTask": [],
  "modelCategory": { "@id": "", "name": "" },
  "evaluatedOn": [],
  "referencePublication": {
    "@type": "ScholarlyArticle",
    "@id": "",
    "author": [{}],
    "datePublished": "",
    "url": "",
    "schema:sameAs": ""
  },
  "codeRepository": ""
}
```

A. Experimentation results for optimal pipeline detection

In order to assess the optimal model choice for accurately detecting each element of the model card, a preliminary study was conducted where three different approaches were considered: a simple NER

Field	Model	F1-Score ↑			Precision ↑			Recall ↑			Time (secs) ↓		
		FP	A	OS	FP	A	OS	FP	A	OS	FP	A	OS
Tasks	<i>GliNER</i>	0.546	0.566	0.345	0.554	0.575	0.350	0.539	0.557	0.339	0.308	0.268	0.190
	<i>qwen3:1.7b</i>	0.810	0.830	0.674	0.802	0.830	0.678	0.820	0.831	0.671	2.161	0.781	0.467
	<i>llama3:8b</i>	0.815	0.857	0.808	0.808	0.867	0.812	0.823	0.849	0.805	2.852	0.761	0.850
Model name	<i>GliNER</i>	0.605	0.599	0.600	0.638	0.635	0.619	0.576	0.570	0.584	1.460	0.162	1.002
	<i>qwen3:1.7b</i>	0.733	0.494	0.348	0.773	0.521	0.367	0.698	0.470	0.331	0.959	0.196	0.212
	<i>llama3:8b</i>	0.740	0.710	0.692	0.765	0.751	0.720	0.718	0.675	0.668	5.752	1.034	1.726
Implementation	<i>GliNER</i>	0.114	0.249	0.147	0.117	0.264	0.152	0.111	0.237	0.143	0.262	0.151	0.196
	<i>qwen3:1.7b</i>	0.452	0.252	0.088	0.452	0.255	0.089	0.453	0.250	0.086	1.348	0.249	0.114
	<i>llama3:8b</i>	0.839	0.322	0.354	0.842	0.327	0.359	0.836	0.319	0.349	4.129	0.642	0.694
Model type	<i>TF-IDF+SVC</i>	0.720	0.720	0.721	0.817	0.818	0.819	0.644	0.643	0.643	0.013	0.001	0.002
	<i>qwen3:1.7b</i>	0.715	0.715	0.715	0.811	0.811	0.811	0.639	0.639	0.639	0.836	0.315	0.388
	<i>llama3:8b</i>	0.704	0.634	0.426	0.797	0.719	0.481	0.631	0.567	0.383	3.122	1.164	1.603

Table 3

Average F1-score, precision, recall and execution time per model, field and content selection (FP= full paper, A= Abstract, OS = Only sections). Best results per field and metric are depicted in bold.

approach, a Small Language Model (SLM) and a Large Language Model (LLM). The study was conducted on a corpus created from the July 2025 PapersWithCode available at <https://huggingface.co/pwc-archive>, retaining articles related to knowledge graph embedding, knowledge graph completion, and link prediction tasks. Reference labels were extracted from these dumps (tasks, authors, implementation links, model names, datasets, and metrics); model categories were assigned manually. The number of labeled instances varies by field: 2,611 for tasks, 172 for datasets, metrics, and model names, 988 for implementation URLs, and 72 for the model category. The results are summarized in Table 3.

A similar study was conducted for those fields where the input data can be either text data or table data. Results for this study are drawn in Table 4.

Field	Model	F1-Score ↑			Precision ↑			Recall ↑			Time (secs) ↓		
		TaO	TeO	TaO+TeO	TaO	TeO	TaO+TeO	TaO	TeO	TaO+TeO	Ta	TeO	TaO+TeO
Datasets	<i>GliNER</i>	0.836	0.810	0.827	0.823	0.783	0.796	0.851	0.838	0.861	0.777	0.514	1.291
	<i>qwen3:1.7b</i>	0.842	0.841	0.836	0.809	0.820	0.801	0.880	0.865	0.876	20.569	0.965	21.536
	<i>llama3:8b</i>	0.832	0.793	0.832	0.808	0.779	0.803	0.858	0.809	0.866	29.008	1.538	30.546
Metrics	<i>GliNER</i>	0.819	0.823	0.841	0.807	0.797	0.812	0.832	0.852	0.874	0.781	0.511	1.291
	<i>qwen3:1.7b</i>	0.858	0.837	0.846	0.829	0.809	0.810	0.890	0.868	0.886	28.614	8.219	36.833
	<i>llama3:8b</i>	0.831	0.839	0.826	0.793	0.811	0.783	0.874	0.869	0.875	39.175	12.682	51.857

Table 4

Average F1-score, precision, recall and execution time per model, field and content selection (TaO= tables only, TeO= full text only, TeO + TaO = tables and full text). Best results per field and metric are depicted in bold.

Once the most reliable approach for each model card field has been detected, a cost analysis study was conducted to determine the most balanced, optimal approach for each field. Efficiency is measured as follows:

$$E_{log} = \frac{F1_{score}^{\alpha}}{\log_{10}(1 + T)} \quad (1)$$

According to the results reported for this metric, the optimal model selection per field was established.