

# Attributes, Taxonomies and Semantic alignment for Automated Research Software Classification

Jenifer Tabita Ciuciu-Kiss<sup>1</sup>

<sup>1</sup>*Ontology Engineering Group (OEG), Universidad Politécnica de Madrid, Madrid, Spain*

## Abstract

Research software (RS) plays a critical role in computational science, yet remains poorly categorized and difficult to discover or reuse. This research explores RS classification by investigating how textual and metadata attributes can be leveraged to develop scalable, interpretable classification methodologies. Existing taxonomies are evaluated through alignment with scientific knowledge graphs to identify redundancies and structural gaps. Labeled datasets are constructed by linking publications to software repositories, and RS attributes, such as README files, abstracts, and source code features are benchmarked using multiple machine learning models and embedding strategies. A methodology that integrates semantic enrichment and transformer-based models is proposed for robust RS classification. Preliminary findings highlight the informativeness of publication abstracts for classification tasks and expose limitations in current community-defined taxonomies.

## Keywords

Research Software Classification, FAIR principles, Software metadata, Scientific Knowledge Graphs

## 1. Problem Statement

Research software (RS) [1] plays a fundamental role in the computational results of scientific publications. However, the classification and integration of RS into scholarly infrastructures remains underdeveloped. Unlike scientific publications, which benefit from standardized indexing practices, taxonomies, and incorporation into scientific knowledge graphs (SKGs), RS is typically shared via code repositories without consistent metadata, formal categories, or semantic links to other research outputs [2].

This lack of structured classification restricts the discoverability, reuse, and proper attribution of RS. It also limits the potential for automation as automated tools rely on machine-readable and semantically consistent categories. Alignment with broader scientific ontologies is important to enable cross-domain reasoning, metadata enrichment, and consistent representation within larger knowledge infrastructures.. Addressing this gap requires a systematic approach to understanding how RS can be categorized, what types of information are useful for classification, and how existing semantic technologies can support this process.

This research adopts a data-driven and semantic-aware methodology to explore RS classification, combining techniques from machine learning, knowledge graph alignment, and natural language processing. The goal is to improve the organization, discoverability, and reuse of RS

---

*ISWC 2025 Companion Volume, November 2–6, 2025, Nara, Japan*

✉ [jenifer.ciuciu-kiss@alumnos.upm.es](mailto:jenifer.ciuciu-kiss@alumnos.upm.es) (J. T. Ciuciu-Kiss)

🆔 0000-0002-3170-6730 (J. T. Ciuciu-Kiss)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

across disciplines by developing scalable and interpretable classification pipelines.

## 2. Importance

The classification of RS is a foundational step toward improving its role as a first-class research output. As scientific inquiry increasingly relies on computational methods, software is not just a tool but a key artifact of knowledge production. Yet, without structured categorization, RS remains difficult to discover, reuse, cite, or integrate into broader research workflows [3, 4, 5, 6].

A classification system addresses several critical challenges and brings benefits to a wide range of stakeholders:

- **Researchers** gain better access to relevant tools and methodologies, reducing duplication of effort and enabling more rapid experimentation and collaboration [3, 6].
- **Repository maintainers and curators** improve the quality and completeness of metadata, enabling more sophisticated filtering, recommendation, and search mechanisms [5].
- **Knowledge graph builders and digital libraries** benefit from richer, semantical connections between software, publications, datasets, and research areas [4, 5].
- **Funding agencies and institutions** can better assess the impact and reuse of software products, supporting open science policies and data management requirements [7, 5].

Scalable classification of RS is critical to managing the growing volume and heterogeneity of repositories on platforms such as GitHub, Zenodo, and Software Heritage. It enables automation in organization, supports semantic search, and improves tools for software discovery, citation tracking, and recommendation.

This work also advances the FAIR principles [8], particularly Findability and Reusability, by enriching metadata and enabling integration with scientific ontologies and knowledge graphs. By exploring categorization through metadata, publication links, and semantic technologies, it contributes to establishing software as a first-class, citable element of scientific output and supports a more connected research ecosystem.

## 3. Related Work

RS classification draws from work on taxonomies and ontologies, dataset construction, software attribute modeling, and classification methodologies. Below we briefly review related literature grouped along these key dimensions.

### 3.1. Taxonomies and ontologies for RS and publications

Numerous efforts have proposed taxonomies to structure scientific knowledge. The Computer Science Ontology (CSO) provides a hierarchical taxonomy of research areas in computer science [9], and the Software Ontology (SWO) focuses on biomedical software types and versions [10]. Papers with Code (PwC) is widely used in machine learning research to assign task- and

method-level categories to software [11]. Other domain-specific resources include EDAM [12], used by bio.tools [13], and the AI Knowledge Graph (AI-KG) [14], which encodes AI concepts in a semantic graph.

More general scientific knowledge graphs such as OpenAIRE [15] and the Open Research Knowledge Graph (ORKG) [16] contain information about publications and, occasionally, their associated software artifacts. While some of these efforts provide strong hierarchical or semantic structures, others remain flat or sparse, with inconsistencies in labeling or concept coverage across domains.

Many existing taxonomies suffer from semantic redundancy or domain specificity, and few are designed to support classification tasks across disciplines. While ontology alignment has been extensively studied [17, 18], our focus extends beyond alignment to include identifying conceptual overlaps, redundancies, and opportunities for simplification across classification schemes. Existing evaluation techniques rarely address these goals directly, highlighting the need for scalable methods to assess and refine taxonomies based on their structure, coherence, and practical utility for classification [19, 20].

### 3.2. Data sources and strategies for constructing software-publication datasets

Public resources such as PwC [11] and OpenAlex [21], link research papers to software implementations, often using GitHub <sup>1</sup> or arXiv <sup>2</sup> metadata. Software citation initiatives like FORCE11 [7] promote structured citation formats for software as first-class research outputs. Efforts like SoftCite [22] focus on extracting unstructured mentions of software in publications to build retrospective links. Repository-level metadata extraction tools such as SoMEF [23] can automate the retrieval of README content and descriptions.

Efforts like RepoFromPaper [24] and citation intent classifiers [25] help identify and validate links between papers and software. Surveyed techniques include DOI matching, co-author heuristics, and bibliographic coupling [26]. However, most large-scale resources are biased towards the machine learning domain or biomedicine, and rarely support multi-label classification out-of-the-box.

While existing pipelines enable partial automation, there is little agreement on what constitutes a robust software-publication link, and many datasets remain noisy, domain-specific, or lack comprehensive label structures. Few works address the challenge of balancing or validating multi-label datasets for classification.

### 3.3. Attributes and embedding strategies for RS classification

A wide range of RS features have been considered for classification: README files [27], abstracts [26], GitHub metadata like stars and forks [28], repository descriptions [29], and code-level features. Works like HiGitClass [30], LabelGit [31], and ClassifyHub [32] benchmark different metadata properties, with varying success across datasets and embedding methods.

In terms of representation, TF-IDF remains a baseline technique for capturing term salience [33], while contextual models such as Sentence-BERT [34] and CLIP [35] offer semantic em-

---

<sup>1</sup><https://github.com/>

<sup>2</sup><https://arxiv.org/>

beddings for unstructured text. These embeddings are often evaluated using classification or clustering metrics such as silhouette score [36] or F1-score to assess discriminative power.

There is no consensus on which software attributes generalize best across domains, and most existing studies only benchmark combinations of attributes on a small subset of models or datasets. Moreover, interpretability and attribute redundancy are rarely examined in-depth.

### 3.4. Classification methodologies and semantic enrichment

Several recent works explore enriched classification pipelines that incorporate ontological information, transformer models, or model interpretability techniques. SKG-based classifiers, such as the CSO Classifier [37], leverage semantic types and hierarchical context. Other methods integrate zero-shot or few-shot LLMs [38], or apply SHAP values [39] to understand feature contributions. Hybrid approaches like AIMMX [29] and SoftCite+NER [22] show that combining ML with domain knowledge improves classification quality in scientific contexts.

Despite this progress, methodological reproducibility remains a challenge, as pipelines are often highly domain-specific and rely on partially annotated or curated inputs. Most pipelines are not reusable outside their source datasets or domains, and they rarely compare multiple models under a shared experimental framework. Furthermore, the cost and scalability of semantic enrichment, particularly when integrating LLMs or large SKGs, are underexplored.

## 4. Research Questions

This doctoral research addresses the overarching question: **How can we classify research software (RS) into categories that support findability, reuse, and analysis?** To investigate this problem, the work is structured around the following research questions:

**RQ1:** How can we identify suitable categories to classify RS?

This involves analyzing and aligning existing taxonomies (e.g., PwC, CSO, SWO) and evaluating their coherence through clustering and semantic similarity.

*Contribution:* A comparative analysis of RS taxonomies, highlighting overlaps and gaps, inconsistencies, clustering quality and proposing an evaluation framework to assess their conceptual clarity. This is combined with a methodology for evaluating taxonomies and exploring automated techniques to enrich them with additional software-related terms.

**RQ2:** How can we (semi)automatically build a labeled dataset of RS and categories that is reliable, reusable, and representative?

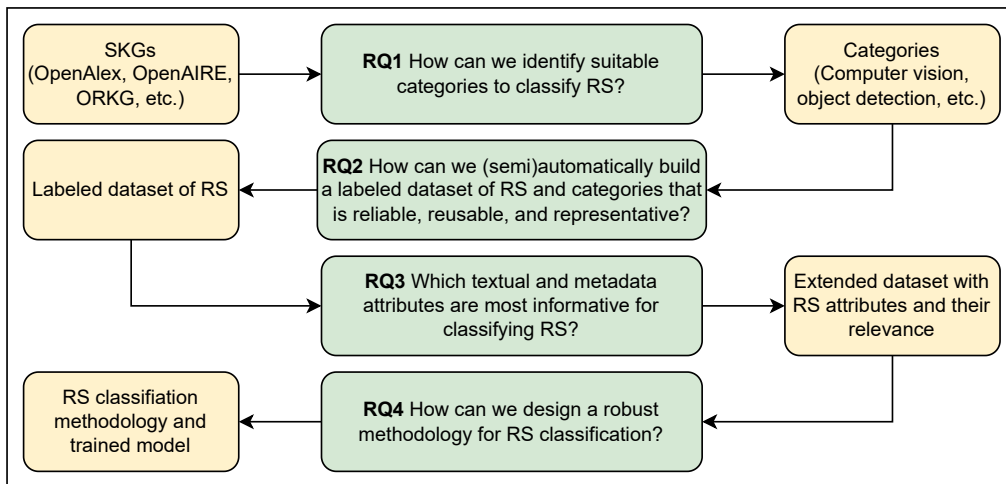
This includes linking software repositories to publications, handling multi-label assignments, and mitigating class imbalance.

*Contribution:* A curated dataset linking RS to publication-derived categories, along with a reproducible methodology for extending this dataset to new domains or knowledge graph structures.

**RQ3:** Which textual and metadata attributes are most informative for classifying RS?

Attributes such as README content, abstracts of linked publications, metadata, and source code features are benchmarked for classification effectiveness.

*Contribution:* A benchmarking framework that evaluates different RS attributes and their



**Figure 1:** Overview of the RS classification workflow and dataflow across research questions. Each component feeds into the next, from taxonomy design to dataset construction, attribute analysis, and methodology integration.

combinations for classification across multiple models and metrics, and a method for identifying the most informative and complementary features.

**RQ4:** How can we design a methodology for RS classification?

The methodology should combine traditional machine learning and transformer-based models with semantic enrichment techniques, including knowledge graph alignment and potentially large language models (LLMs).

*Contribution:* A methodology that integrates semantic knowledge graphs and machine learning for scalable and interpretable RS classification.

## 5. Preliminary Results

The preliminary results of this research provide early validation for each of the four RQs, which are also visually connected in Figure 1. This figure illustrates the logical flow between research questions: RQ1 defines the classification targets, RQ2 constructs the labeled dataset, RQ3 assesses the predictive value of different attributes, and RQ4 integrates these components into a coherent classification methodology. Each RQs were explored at a high level in earlier work during my Master’s thesis [40], which motivated the structure of the present thesis. We detail the research questions with preliminary work below. The methodology design for RQ1, RQ2 and RQ3 has been implemented and partially tested, although the full analysis is still in progress. RQ4 remains part of future work.

## 5.1. RQ1: Taxonomy Evaluation and Alignment

Preliminary work on RQ1 resulted in the alignment of 71 Papers with Code (PwC) categories [11] to concepts in the Computer Science Ontology (CSO) [9] and OpenAlex [21] using cosine similarity between Sentence-BERT (SBERT) [34] embeddings and manually curated mappings. This work, published in our NSLP 2024 paper [41], revealed conceptual redundancies in community-defined labels and helped identify opportunities for consolidation and enrichment through scientific knowledge graph (SKG) alignment. Further evaluation using clustering metrics—such as silhouette score [36], Davies-Bouldin index [42], and Dunn index [43], on category embeddings confirmed low intra-class cohesion for several overlapping categories, particularly in NLP-related tasks like “text-classification” and “sentiment-analysis.” This supports RQ1 by showing that some categories lack semantic distinctiveness, highlighting the need to refine or merge them for effective classification.

One example from our SKG alignment analysis involves the category “lasso (programming language)” in OpenAlex, which was frequently associated with papers that, in fact, addressed “lasso (statistics),” a regression technique. Although the paper content clearly referred to statistical methods, the label misalignment led to incorrect categorization across knowledge graphs. This example underscores the importance of validating category assignments through semantic similarity and expert review, especially in ambiguous or overloaded terms. It also illustrates the potential of embedding-based methods to detect such inconsistencies and guide taxonomy refinement.

## 5.2. RQ2: Dataset Construction from RS–Publication Links

To address RQ2, a preliminary labeled dataset was constructed using paper–code links from PwC [11]. From these links, publication abstracts, GitHub metadata (e.g., repository titles, stars, forks), and README files were extracted using the GitHub API and the SOMEF tool [23]. Community-curated sources such as Awesome Lists and partially annotated links from knowledge graphs were also considered to expand coverage. Instances with multiple labels were filtered to simplify the task to a multi-class setting, as a first step toward building a reliable multi-label dataset. This approach draws on techniques validated in my master’s thesis [40], where semi-automated linking workflows were manually evaluated for precision and coverage. These early results confirm the feasibility of paper–code dataset construction but also highlight the need for more scalable, automated methods for multi-label assignment and balancing for the categories selected in the previous step. The next step involves extending the dataset to more domains and integrating confidence estimation mechanisms for noisy links.

In this context, a multi-label assignment refers to cases where a software repository is associated with several scientific categories—for example, a tool for generating datasets using synthetic data may fall under both Data Augmentation and Simulation. Such overlap reflects real-world interdisciplinarity but introduces modeling complexity. Additionally, the dataset is affected by class imbalance: high-frequency categories like Computer Vision dominate the corpus, while others such as Anomaly Detection or Graph Learning are underrepresented. This imbalance can lead to biased models that generalize poorly to rare classes. We plan to mitigate this through undersampling of dominant categories during training and by exploring

reweighting and augmentation strategies to achieve a more balanced representation across labels.

### **5.3. RQ3: Attribute Benchmarking for Classification**

. Initial experiments compared RS attributes using a stratified One-vs-Rest classification pipeline [44] with undersampling. Features included README content, publication abstracts, GitHub titles, and keyword metadata. Textual data was embedded using TF-IDF [33], Sentence-BERT [34], and CLIP [35]. Ongoing work suggests that publication abstracts consistently yield the highest macro-F1 scores across all embedding methods, indicating they are the most informative individual attribute. These experiments provide initial evidence to guide the prioritization of textual features in classification pipelines. Next steps include extending the evaluation to combinations of attributes, introducing code-level features, and analyzing feature contributions using interpretability techniques such as SHAP [39].

## **6. Evaluation Plan**

The evaluation follows the four RQs, each targeting a core challenge in RS classification. We use a mix of quantitative and qualitative methods, with metrics tailored to relevant areas such as ontology alignment, clustering, dataset construction, and classification. The following subsections outline the strategy for each RQ.

### **6.1. RQ1: Evaluating RS Categories with Taxonomy Alignment and Clustering**

To assess the quality of existing taxonomies for classifying RS, we will apply a quantitative ontology alignment methodology introduced in our NSLP 2024 paper [41]. This method evaluates alignments based on structural coverage, term overlap, and semantic coherence, providing a scalable and interpretable way to compare taxonomies and knowledge graphs such as OpenAlex [21], OpenAIRE [45], the Open Research Knowledge Graph (ORKG) [16], Papers with Code (PwC) [11], the Computer Science Ontology (CSO) [9], and the Software Ontology (SWO) [10]. Unlike manual or gold-standard comparisons, this approach enables a data-driven assessment of taxonomy compatibility and conceptual clarity.

To assess the intrinsic coherence of the categories themselves, we will analyze how well they form clusters based on their semantic representations. Specifically, we compute clustering quality metrics such as the Silhouette Coefficient [36], the Davies-Bouldin Index [42], and the Dunn Index [43] on embeddings of category descriptions. High scores on these metrics indicate that the categories capture semantically distinct and internally coherent groupings, which is a desirable property for any classification scheme.

### **6.2. RQ2: Evaluating the Construction of a Labeled RS Dataset**

To evaluate the linking between software repositories and scientific publications, we will assess existing techniques that use metadata matching based on DOIs, author names, repository URLs, and citation contexts [46]. Methods such as heuristic-based string matching and bibliographic

coupling [26] will be considered. In addition to automated techniques, we will review existing community-curated resources such as Awesome lists [47], and note that some scientific knowledge graphs, such as OpenAIRE [45] and ORKG [16], store software-publication links. However, coverage is limited and biased toward recent publications. We will evaluate the precision, recall, and F1-score of these mappings against a manually curated standard to assess their reliability.

To ensure that the dataset supports multi-label tasks, we will analyze the co-occurrence of labels per software instance. We will compute statistics such as label cardinality and label density [48] to confirm the multi-label nature of the data. To assess label balance, we will examine the frequency distribution of individual and co-occurring labels. Skewness and long-tail effects will be quantified using metrics like the imbalance ratio and entropy across label frequencies [49]. If imbalance is found, we will consider balancing strategies during dataset construction.

### **6.3. RQ3: Evaluating Informative Attributes for RS Classification**

RS can be represented using textual and metadata attributes. Common textual sources include README files [27] and publication abstracts [26], while GitHub metadata, like repository title and descriptions reflect community contributions [28]. Source code features such as function names and comments may provide technical insight but are more complex to extract.

We will benchmark each attribute, individually and in combination, across various machine learning and transformer-based models to assess their contribution to classification performance. For textual features, we will use embedding strategies from TF-IDF [33] to Sentence-BERT [34] and CLIP [35]. Evaluation will include precision, recall, F1-score, and SHAP-based attribution [39]. We aim to identify compact yet informative feature sets and assess trade-offs between informativeness and extractability. We will also compare these approaches with large language models to understand their added value under resource constraints.

### **6.4. RQ4: Evaluating the RS Classification Methodology**

To evaluate the proposed classification methodology, we will measure both model performance and its ability to generalize across different datasets. Performance will be assessed using macro-averaged F1-score, accuracy, and AUC-ROC across different classifiers, including traditional machine learning models and transformer-based approaches such as BERT [50]. We will evaluate the generalizability of the method by training on one labeled dataset and testing on another (cross-dataset transfer).

The added value of semantic enrichment through knowledge graphs (e.g., CSO, AI-KG) will be evaluated by comparing classification performance before and after enrichment. If LLM-based embeddings or entity linking are used, their contribution will also be isolated and measured. Finally, scalability and efficiency will be evaluated using time and memory usage benchmarks during training and inference, ensuring the methodology is applicable to real-world, large-scale scenarios, as suggested in prior work on semantic web systems [51].

## 7. Reflection and Future Work

This work provides a foundation for RS classification by combining metadata benchmarking, taxonomy evaluation, and semantic alignment through knowledge graphs. Preliminary results demonstrate the feasibility of using abstracts for classification, highlight structural inconsistencies in current taxonomies, and confirm the potential of SKG alignment to enhance interpretability.

However, several challenges and limitations remain. Repository metadata is often sparse or inconsistently maintained, limiting its reliability as an input source. Many existing taxonomies lack clear definitions or consistent granularity, complicating automated classification. A key assumption is that textual representations like READMEs or abstracts reflect a software’s actual functionality and domain—yet this is not always the case, as documentation can be outdated or incomplete. The evolving nature of software also introduces subjectivity: a repository may span multiple domains over time, or its README may no longer correspond to its latest capabilities. Additionally, while large language models offer promising results, their performance is sensitive to domain shifts and their outputs can be difficult to interpret. Semantic alignment using SKGs, though valuable, still requires manual validation to ensure accurate mappings.

Future work includes scaling the dataset to support multi-label classification, refining the proposed taxonomy based on clustering and alignment results [36, 51], and further exploring LLMs in few-shot settings [38]. Additional integration with platforms like OpenAIRE [45], Software Heritage [52], and Wikidata [53] is planned to promote interoperability and reuse.

## Acknowledgments

The author would like to thank Daniel Garijo for supervising this work.

## Declaration on Generative AI

All the research content and ideas are original to the author. We acknowledge the use of ChatGPT for supervised grammar checks and minor paragraph rewording.

## References

- [1] M. Gruenpeter, D. S. Katz, A.-L. Lamprecht, T. Honeyman, D. Garijo, A. Struck, A. Niehues, P. A. Martinez, L. J. Castro, T. Rabemanantsoa, N. P. Chue Hong, C. Martinez-Ortiz, L. Sesink, M. Liffers, A. C. Fouilloux, C. Erdmann, S. Peroni, P. Martinez Lavanchy, I. Todorov, M. Sinha, *Defining Research Software: a controversial discussion*, 2021. doi:10.5281/zenodo.5504016.
- [2] M. Hucka, M. J. Graham, *Software search is not a science, even among scientists*, arXiv preprint arXiv:1605.02265 (2016).
- [3] M. Hucka, M. J. Graham, *Software search is not a science, even among scientists: A survey of how scientists and engineers find software*, *Journal of Systems and Software* 141 (2018) 102–114. doi:10.1016/j.jss.2018.03.043.
- [4] S. Hettrick, M. Antonioletti, L. Carr, N. Chue Hong, S. Crouch, D. C. De Roure, I. Emsley, C. Goble, A. Hay, D. Inupakutika, et al., *Uk research software survey 2014* (2014).
- [5] M. A. Hossain, Y. K. Dwivedi, N. P. Rana, *State-of-the-art in open data research: Insights from existing literature and a research agenda*, *Journal of organizational computing and electronic commerce* 26 (2016) 14–40.
- [6] W. Maalej, R. Tiarks, T. Roehm, R. Koschke, *On the comprehension of program comprehension*, *ACM Transactions on Software Engineering and Methodology (TOSEM)* 23 (2014) 1–37.
- [7] A. M. Smith, D. S. Katz, K. E. Niemeyer, *Software citation principles*, *PeerJ Computer Science* 2 (2016) e86.
- [8] 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) 1–9.
- [9] A. A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, E. Motta, *The computer science ontology: a large-scale taxonomy of research areas*, in: *The Semantic Web–ISWC 2018: 17th International Semantic Web Conference, Monterey, CA, USA, October 8–12, 2018, Proceedings, Part II* 17, Springer, 2018, pp. 187–205.
- [10] J. Malone, A. Brown, A. L. Lister, J. Ison, D. Hull, H. Parkinson, R. Stevens, *The software ontology (swo): a resource for reproducibility in biomedical data analysis, curation and digital preservation*, *Journal of biomedical semantics* 5 (2014) 1–13.
- [11] M. AI, *Papers with code*, 2025. URL: <https://paperswithcode.com>.
- [12] J. Ison, M. Kalaš, I. Jonassen, D. Bolser, M. Uludag, H. McWilliam, J. Malone, R. Lopez, S. Pettifer, P. Rice, *Edam: an ontology of bioinformatics operations, types of data and identifiers, topics and formats*, *Bioinformatics* 29 (2013) 1325–1332.
- [13] J. Ison, H. Ienasescu, P. Chmura, E. Rydza, H. Ménager, M. Kalaš, V. Schwämmle, B. Grüning, N. Beard, R. Lopez, et al., *The bio. tools registry of software tools and data resources for the life sciences*, *Genome biology* 20 (2019) 1–4.
- [14] 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, Athens, Greece, November 2–6, 2020, Proceedings, Part II* 19, Springer, 2020, pp. 127–143.
- [15] P. Manghi, M. Artini, C. Atzori, M. Baglioni, A. Bardi, S. La Bruzzo, M. De Bonis, H. Dim-

- itropoulos, I. Foufloulas, K. Iatropoulou, et al., Openaire: Advancing open science, in: Proceedings of the Nineteenth International Conference on Grey Literature, Rome, Italy, 2017, pp. 23–24.
- [16] 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.
- [17] P. Shvaiko, J. Euzenat, Ontology matching: state of the art and future challenges, IEEE Transactions on knowledge and data engineering 25 (2011) 158–176.
- [18] F. Ardjani, D. Bouchiha, M. Malki, Ontology-alignment techniques: survey and analysis, International Journal of Modern Education and Computer Science 7 (2015) 67.
- [19] J. Brank, M. Grobelnik, D. Mladenic, A survey of ontology evaluation techniques, in: Proceedings of the conference on data mining and data warehouses (SIKDD 2005), Citeseer, 2005, pp. 166–170.
- [20] D. Lavbič, M. Krisper, Facilitating ontology development with continuous evaluation, arXiv preprint arXiv:1807.04090 (2018).
- [21] J. Priem, H. Piwowar, R. Orr, Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts, arXiv preprint arXiv:2205.01833 (2022).
- [22] C. Du, J. Cohoon, P. Lopez, J. Howison, Softcite dataset: A dataset of software mentions in biomedical and economic research publications, Journal of the Association for Information Science and Technology 72 (2021) 870–884.
- [23] A. Mao, D. Garijo, S. Fakhraei, Somef: A framework for capturing scientific software metadata from its documentation, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 3032–3037.
- [24] A. Stankovski, D. Garijo, Repofrompaper: An approach to extract software code implementations from scientific publications, in: International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs, Springer, 2024, pp. 100–113.
- [25] P. Koloveas, S. Chatzopoulos, T. Vergoulis, C. Tryfonopoulos, Can llms predict citation intent? an experimental analysis of in-context learning and fine-tuning on open llms, arXiv preprint arXiv:2502.14561 (2025).
- [26] E. Fregnan, T. Baum, F. Palomba, A. Bacchelli, A survey on software coupling relations and tools, Information and Software Technology 107 (2019) 159–178.
- [27] G. A. A. Prana, C. Treude, F. Thung, T. Atapattu, D. Lo, Categorizing the content of github readme files, Empirical Software Engineering 24 (2019) 1296–1327.
- [28] H. Borges, M. T. Valente, What’s in a github star? understanding repository starring practices in a social coding platform, Journal of Systems and Software 146 (2018) 112–129.
- [29] J. Tsay, A. Braz, M. Hirzel, A. Shinnar, T. Mummert, Aimmx: Artificial intelligence model metadata extractor, in: Proceedings of the 17th international conference on mining software repositories, 2020, pp. 81–92.
- [30] Y. Zhang, F. F. Xu, S. Li, Y. Meng, X. Wang, Q. Li, J. Han, Higitclass: Keyword-driven hierarchical classification of github repositories, in: 2019 IEEE International Conference on Data Mining (ICDM), IEEE, 2019, pp. 876–885.
- [31] C. Sas, A. Capiluppi, Labelgit: A dataset for software repositories classification using attributed dependency graphs, arXiv preprint arXiv:2103.08890 (2021).

- [32] M. Soll, M. Vosgerau, Classifyhub: an algorithm to classify github repositories, in: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz), Springer, 2017, pp. 373–379.
- [33] J. Ramos, et al., Using tf-idf to determine word relevance in document queries, in: Proceedings of the first instructional conference on machine learning, volume 242, Citeseer, 2003, pp. 29–48.
- [34] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, arXiv preprint arXiv:1908.10084 (2019).
- [35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International conference on machine learning, PmLR, 2021, pp. 8748–8763.
- [36] P. J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of computational and applied mathematics* 20 (1987) 53–65.
- [37] A. A. Salatino, F. Osborne, T. Thanapalasingam, E. Motta, The cso classifier: Ontology-driven detection of research topics in scholarly articles, in: Digital Libraries for Open Knowledge: 23rd International Conference on Theory and Practice of Digital Libraries, TPDL 2019, Oslo, Norway, September 9-12, 2019, Proceedings 23, Springer, 2019, pp. 296–311.
- [38] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, *Advances in neural information processing systems* 33 (2020) 1877–1901.
- [39] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, *Advances in neural information processing systems* 30 (2017).
- [40] J. T. Ciuciu-Kiss, A methodology for research software classification, Master’s thesis, UPM, 2022.
- [41] J. T. Ciuciu-Kiss, D. Garijo, Assessing the overlap of science knowledge graphs: A quantitative analysis, in: International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs, Springer, 2024, pp. 171–185.
- [42] D. L. Davies, D. W. Bouldin, A cluster separation measure, *IEEE transactions on pattern analysis and machine intelligence* (2009) 224–227.
- [43] J. C. Dunn, Well-separated clusters and optimal fuzzy partitions, *Journal of cybernetics* 4 (1974) 95–104.
- [44] R. Rifkin, A. Klautau, In defense of one-vs-all classification, *Journal of machine learning research* 5 (2004) 101–141.
- [45] P. Manghi, A. Bardi, C. Atzori, M. Baglioni, N. Manola, J. Schirrwagen, P. Principe, M. Artini, A. Becker, M. De Bonis, et al., The openaire research graph data model, Zenodo (2019).
- [46] H. Hata, J. L. Guo, R. G. Kula, C. Treude, Science-software linkage: the challenges of traceability between scientific knowledge and software artifacts, arXiv preprint arXiv:2104.05891 (2021).
- [47] R. V. Small, M. Arnone, Awesome web sites: How to find and use them, *Knowledge Quest* 30 (2001) 38.
- [48] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Džeroski, An extensive experimental comparison of methods for multi-label learning, *Pattern recognition* 45 (2012) 3084–3104.
- [49] F. Charte, A. J. Rivera, M. J. Del Jesus, F. Herrera, Mlsmote: Approaching imbalanced

multilabel learning through synthetic instance generation, *Knowledge-Based Systems* 89 (2015) 385–397.

- [50] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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)*, 2019, pp. 4171–4186.
- [51] H. Paulheim, Knowledge graph refinement: A survey of approaches and evaluation methods, *Semantic web* 8 (2016) 489–508.
- [52] A. Pietri, D. Spinellis, S. Zacchiroli, The software heritage graph dataset: public software development under one roof, in: *2019 IEEE/ACM 16th International Conference on Mining Software Repositories (MSR)*, IEEE, 2019, pp. 138–142.
- [53] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, *Communications of the ACM* 57 (2014) 78–85.