

# A Study of the Categories used in ‘Papers with Code’

Jenifer Tabita Ciuciu-Kiss<sup>1</sup>, Daniel Garijo<sup>1</sup>

<sup>1</sup>Universidad Politécnica de Madrid, Boadilla del Monte, Madrid, Spain

## Abstract

An increasing number of machine learning developers share research software online to support their scientific investigations. In order to improve software findability, the scientific community has developed domain-specific taxonomies. However, are these taxonomies appropriate for software classification? This paper explores this question through a case study on Papers with Code, a popular platform where authors share their publications together with their software implementations. We define and apply a comparative framework with state-of-the-art text similarity techniques (TF-IDF, Sentence-BERT, CLIP), and we assess the level of overlap between different software categories defined in the platform, based on the methods descriptions contained in them. Our results show significant category overlap, which may limit the effectiveness of classification algorithms. While community-defined categories provide a useful foundation, they may require refinement, such as subcategories or refined definitions, to better capture interdisciplinary methods and improve classification accuracy.

## Keywords

Research Software Classification, Clustering Quality Analysis, FAIR

## 1. Introduction

In parallel with the adoption of the Findable, Accessible, Interoperable and Reusable (FAIR) guiding principles [1], research software [2] has gained increasing recognition as a first-class research output [3]. Classification of research software is key for supporting findability, improving the discoverability of software tools in scientific research, and promoting the reuse of existing solutions. With the exponential growth in the number of software tools available, the process of finding the most appropriate and relevant software has become more challenging for researchers [4].

A well-structured taxonomy is essential to ease software findability, as it provides an agreed framework that organizes software into distinct common categories based on functionality, domain, or other relevant characteristics. This ensures that both researchers and automated systems can effectively filter and compare tools, making valuable research software easier to locate and apply in diverse contexts. To this end, different communities have proposed various taxonomies [5, 6, 7] of their own for manual or curated research artifact classification. However, it is often unclear whether the choice of selected categories is appropriate for research software classification (i.e., are two categories too similar or redundant?).

In this paper, we examine this issue through a case study on Papers with Code [7], a popular platform designed to capture scientific articles and their corresponding implementations in the Machine Learning domain. Papers with Code contains a crowdsourced software taxonomy with hundreds of different software categories, which have been used to feed several existing methods for research software classification [8, 9, 10, 11]. Our contributions include: 1) a methodology for evaluating the coherence and separability of research software categories, including how the research software categories were analyzed using text embeddings and clustering techniques; 2) the results of the case study, which aims to address whether the level of noise in the categories affects their suitability for classification, highlighting the extent of category overlap and its impact on clustering performance. The implementation of our

*2nd International Workshop on Natural Scientific Language Processing and Research Knowledge Graphs (NSLP 2025), co-located with ESWC 2025, June 01–02, 2025, Portorož, Slovenia*

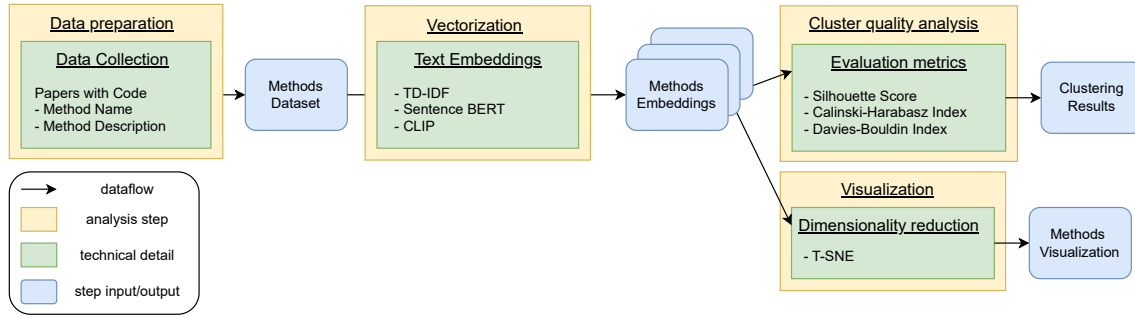
✉ jenifer.ciuciu-kiss@alumnos.upm.es (J. T. Ciuciu-Kiss); daniel.garijo@upm.es (D. Garijo)

🌐 <https://jeniferciuciu-kiss.com/> (J. T. Ciuciu-Kiss); <https://dgarijo.com/> (D. Garijo)

🆔 0000-0002-3170-6730 (J. T. Ciuciu-Kiss); 0000-0003-0454-7145 (D. Garijo)



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



**Figure 1:** Overview of our methodology, including steps for data preparation, vectorization, clustering quality analysis with evaluation metrics and visualization. For each step we depict its corresponding input and output.

analysis and dataset is available on GitHub [12].<sup>1</sup>

The remainder of the paper is structured as follows. Section 2 reviews existing approaches for assessing category similarity and taxonomies for research software classification. Section 3 describes the dataset, text embedding techniques, and clustering quality analysis metrics used to evaluate category coherence. Section 4 presents the clustering quality analysis and visualization, highlighting both the separability and overlap of community-defined categories. Finally, Section 5 summarizes our key findings.

## 2. Related Work

Various tools have been proposed to assess category similarity in research software classification and knowledge organization. Ontology alignment tools compare structured taxonomies through category labels, descriptions, and hierarchical structures to compute similarity scores [13, 14]. Knowledge graph-based approaches leverage structured data and embeddings to identify conceptual relationships between research topics and software categories [5, 15].

Research software taxonomies help structure classification systems for retrieval and organization. For example, the Computer Science Ontology (CSO) [16] has been proposed to structure scientific publications and overlaps with software-related topics. The Software Ontology (SWO) [6] and Bio.tools [17] provide domain-specific software categorization, particularly for biomedical applications using the EDAM ontology [17].

More general taxonomies include Papers with Code [7], which categorizes software implementations in Machine Learning and explicitly links research papers to their implementations [8]. Additionally, Science Knowledge Graphs, such as the AI Knowledge Graph (AI-KG) [18] and OpenAIRE [19], contain software entities but focus primarily on documenting relationships between scientific concepts, publications, and datasets.

Despite these efforts, little work has systematically evaluated whether community-defined research software categories align with natural groupings in category definitions. Existing taxonomies are often created based on expert knowledge rather than empirical validation, raising questions about their effectiveness for classification. This study aims to bridge this gap by assessing the coherence of research software categories using text embeddings and clustering techniques.

## 3. Methodology

Figure 1 outlines the methodology by breaking down the different steps into data preparation, vectorization, clustering quality analysis and visualization. The goal of the methodology is to assess whether

<sup>1</sup>[https://github.com/kuefmz/pow\\_categories/tree/main](https://github.com/kuefmz/pow_categories/tree/main)



**Figure 2:** Data model and taxonomy of research areas and methods from Papers with Code.

a given set of research software categories (e.g., community-defined) provide a foundation for software classification.

### 3.1. Data sources

We adopt the Papers with Code platform, which has emerged as a key resource within the research community, particularly in machine learning and artificial intelligence. This platform integrates research publications with their respective software implementations, offering a holistic approach that bridges the gap between research and application. Their mission is: "to create a free and open resource with Machine Learning papers, code, datasets, methods and evaluation tables."<sup>2</sup>

In addition, Papers with Code categorizes paper-code links manually into different categories. Due to its popularity and widespread use, it provides access to a manually curated set of categories and their descriptions, which is the focus of our study. This manual curation ensures the high quality and relevance of categories, which significantly aids in exploring the research landscape.

### 3.2. Dataset

As shown in Figure 1, the dataset was collected following the data model shown in Figure 2 to create the ‘Methods Dataset’. The dataset is organized around methods associated with specific research areas (e.g., Computer vision, Natural language processing). Each entry in this dataset includes a method name and a detailed description, annotated under a particular research area by the community. This structured organization provides a taxonomy of methods, allowing us to examine whether these community-defined categories naturally form coherent clusters when represented through textual attributes.

The dataset contains **1,064** methods sourced from the Papers with Code platform. Each method is categorized into a specific research area: **Computer vision** (665 methods), **Natural language processing** (119 methods), **Graphs** (104 methods), **Reinforcement learning** (88 methods), **Sequential** (53 methods), and **Audio** (35 methods). Each entry includes the name, description, and associated research area of a method. The dataset was retrieved from Papers with Code on October 12, 2024, using a publicly available JSON file.<sup>3</sup>

### 3.3. Vectorization

To convert textual attributes into numerical representations suitable for machine learning models, we employ and compare three types of text embeddings:

- **TF-IDF [20]:** A lightweight approach for representing text by assigning weights to terms based on their frequency within a document and across the dataset. Despite its simplicity, TF-IDF highlights the most relevant terms within each document, which can help in identifying keywords.
- **Sentence-BERT (SBERT) [21]:** approach for generating dense, context-aware embeddings for sentence-level text, such as abstracts and descriptions. By capturing semantic relationships within sentences, SBERT provides a deeper understanding of the context and meaning of words relative to each other, rather than treating them as isolated terms.

<sup>2</sup><https://paperswithcode.com/about>

<sup>3</sup><https://production-media.paperswithcode.com/about/methods.json.gz>

- **CLIP [22]:** Originally designed for multimodal learning, CLIP’s text encoder can still generate meaningful, contextually rich embeddings for textual tasks. By training on a vast array of web data, CLIP has developed the ability to recognize complex language patterns and associations, which is useful for handling diverse text data.

Comparing different embedding techniques is important because each technique captures semantic information differently, which can significantly impact the performance and interpretability of the clustering and classification tasks. These three techniques were chosen specifically because they each address different aspects of textual representation: TF-IDF for term frequency-based keyword extraction, SBERT for semantic understanding at the sentence level, and CLIP for capturing broader, high-level contextual relationships. We used the following software versions in our experiments to ensure reproducibility: `sentence-transformers==3.1.1`, `transformers==4.45.1`, `scikit-learn==1.4.1.post1`. All experiments were conducted using Python 3.10.

### 3.4. Cluster quality analysis

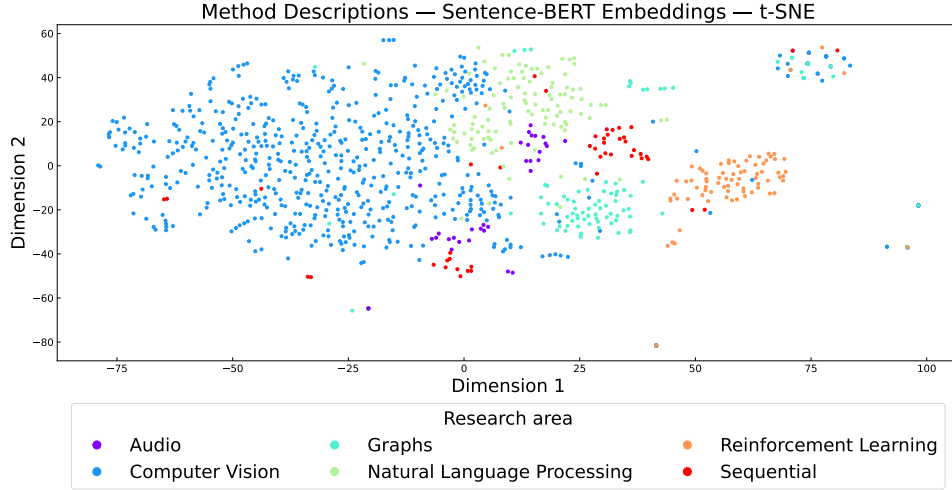
Cluster quality analysis is conducted to assess the natural grouping of research software categories based on their textual embeddings on the analyzed method names and descriptions. The goal is to examine whether community-defined categories form distinct clusters when represented by textual attributes such as method names or descriptions. Clustering quality is evaluated using the following metrics:

- **Silhouette Score (SS) [23]:** Measures cluster separation based on the average distance between clusters. The score ranges from  $-1$  to  $+1$ , where higher scores indicate more distinct clusters. A value close to  $+1$  suggests that samples are well-matched to their own cluster and poorly matched to neighboring clusters, while a value near  $0$  implies overlapping clusters. Negative values indicate that samples may have been assigned to the wrong cluster.
- **Calinski-Harabasz Index (CHI) [24]:** Reflects the ratio of the sum of between-cluster dispersion to within-cluster dispersion, with higher values indicating better-defined and more compact clusters. The CHI is nonnegative and increases as the clusters become more compact and better separated. Higher values generally imply that clusters are dense and well-separated, which is ideal for clustering performance.
- **Davies-Bouldin Index (DBI) [25]:** Evaluates the average similarity ratio of each cluster with the cluster most similar to it. The score ranges from  $0$  upwards, where lower values indicate better separation and more distinct clusters. A DBI score closer to  $0$  implies low similarity between clusters, suggesting effective clustering, while higher values indicate clusters that overlap or are poorly separated.

These three metrics were selected for their complementary strengths in assessing clustering quality. The Silhouette Score evaluates how well each sample matches its own cluster versus neighboring ones, providing insight into cluster separation. The Calinski-Harabasz Index measures cluster compactness and separation, indicating well-defined clusters with higher values. Finally, the Davies-Bouldin Index assesses distinctness by evaluating similarity between clusters, with lower values reflecting minimal overlap. Together, these metrics offer a balanced view of clustering performance by capturing separation, cohesion, and distinctness.

### 3.5. Visualization

T-SNE [26], a dimensionality reduction technique, is used to visualize the embeddings and assess whether the research software attributes align with the community-defined categories. These visualizations provide qualitative support for the quantitative evaluation of clustering and classification by illustrating the distinctiveness of each attribute in capturing category differences. By examining the visual clustering patterns, we gain insights into how well the embeddings represent natural groupings, complementing the quantitative metrics with a visual assessment of the category separability.



**Figure 3:** t-SNE visualization of Sentence-BERT embeddings (Method descriptions), colored by research area. Each method description is represented with a dot. The plot shows that some categories, such as Audio and Sequential, form relatively distinct clusters in the t-SNE projection space. In contrast, categories like Graphs and Natural Language Processing show significant overlap. This visual pattern aligns with our quantitative findings and illustrates that certain method descriptions use similar terminology or concepts across categories.

## 4. Results

In this study, we define a coherent cluster as a group of method descriptions that are closely grouped in the embedding space and belong to the same category, with minimal overlap with other categories. This coherence is indicative of a well-separated and semantically meaningful category.

We used the dataset presented in Section 3.2 to determine whether community-defined categories, represented by method names and descriptions, form distinct clusters that may serve as a solid foundation for future classification tasks.

Table 1 shows the clustering quality analysis over metric names and definitions, using different embedding techniques. Method descriptions, particularly when embedded using Sentence-BERT, provide better clustering of the community-defined research software categories than method names. The Sentence-BERT embeddings for descriptions achieved the highest Calinski-Harabasz Index (67.37) and the lowest Davies-Bouldin Index (3.56). However, the low Silhouette Scores across all embeddings indicate weak separation between categories, suggesting that category boundaries may not be clearly defined. This overlap likely reduces the classification signal, especially for categories such as Graphs and Sequential, which show low inter-category distinction. These findings point to potential redundancy or ambiguity in the current taxonomy. For example, a method used in Natural language processing (NLP) may also be applicable in Computer vision (CV) when dealing with multimodal data that combines text and images. Such examples highlight the challenge of achieving clear-cut clusters, as certain methods are inherently versatile and cross-disciplinary.

**Table 1**

Clustering performance metrics for method names and descriptions using different embedding types. Metrics are computed over six manually assigned research areas: Computer Vision, Natural Language Processing, Graphs, Reinforcement Learning, Sequential, and Audio.

| Embedding | SS    |       | CHI  |              | DBI   |             |
|-----------|-------|-------|------|--------------|-------|-------------|
|           | Name  | Desc. | Name | Desc.        | Name  | Desc.       |
| TF-IDF    | -0.10 | -0.13 | 2.20 | 18.93        | 25.11 | 9.40        |
| S-BERT    | -0.11 | -0.01 | 6.96 | <b>67.37</b> | 21.54 | <b>3.56</b> |
| CLIP      | -0.10 | -0.08 | 6.31 | 19.18        | 10.66 | 6.15        |

The results indicate that while the current categories provide a starting point for classification tasks, their effectiveness is limited due to significant overlap, which introduces noise and reduces their reliability. This suggests that classifications in this space should be interpreted with caution, as the boundaries between categories are not well-defined. Rather than relying solely on the existing taxonomy, further efforts are needed to improve category definitions by incorporating clearer textual descriptions and more representative examples. Such refinements may help mitigate ambiguity and better capture the nuances of methods that span multiple areas, potentially enhancing classification accuracy while acknowledging the inherent limitations of the current structure.

To further explore how well the categories are visually distinct, we applied t-SNE to the Sentence-BERT embeddings of the method descriptions, as these achieved the best clustering performance based on the Calinski-Harabasz and Davies-Bouldin indexes. The t-SNE visualization in Figure 3 shows that there is some overlap between categories in certain areas, such as Computer Vision, which tend to form dense clusters, suggesting that some categories are more distinguishable than others. Natural language processing and Reinforcement learning show more dispersion, reflecting the challenge in categorizing methods that may span multiple domains. Overall, the visualization provides additional insight into the structure of the embeddings, illustrating both the strengths and limitations of the current category definitions.

## 5. Conclusions and Future Work

Automated classification of research software is essential for improving findability and supporting reuse, particularly as research outputs become increasingly available on the Web. In this study, we examined whether community-defined categories in Papers with Code form distinct clusters that align with natural groupings in the data. Our results indicate that some categories, such as Computer Vision and Natural Language Processing, exhibit clear separation, while others, including Graphs and Reinforcement Learning, show substantial overlap. This overlap suggests that classification based on these categories may introduce noise, even when state-of-the-art methods achieve high performance.

Our clustering results indicate that while the existing taxonomy provides a useful foundation, its effectiveness is hindered by ambiguous category boundaries. The presence of overlapping categories suggests that certain research methods span multiple fields, making strict classification challenging. Rather than relying solely on predefined categories, future classification efforts may explore refining taxonomies by introducing additional subcategories or restructuring category definitions based on empirical clustering results. Additionally, incorporating richer metadata, such as method usage context and domain-specific relationships, may enhance classification accuracy. Furthermore, category refinement may directly improve the usability of platforms like Papers with Code by offering more precise filtering options for users. Incorporating subcategories or supporting multi-label assignments would accommodate interdisciplinary methods, reducing misclassification and improving discoverability.

Our future work will evaluate the impact of category refinement on classification performance by testing machine learning models under different category structures. Another direction is to investigate methods for systematically identifying and resolving overlapping categories, such as hierarchical clustering approaches or semi-supervised learning techniques that integrate expert feedback. More broadly, improving research software classification aligns with the FAIR principles, particularly Findability, by ensuring that software tools are categorized in a way that accurately reflects their purpose and functionality. Addressing these challenges will contribute to more reliable and interpretable research software classification, supporting both automated discovery systems and the broader scientific community.

## Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT for grammar checks and rewording. After utilizing these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication’s content.



## References

- [1] 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, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. G. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. C. 't Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, B. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, A. Mons, The fair guiding principles for scientific data management and stewardship, *Scientific Data* 3 (2016) 160018. doi:10.1038/sdata.2016.18.
- [2] 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.
- [3] N. P. Chue Hong, D. S. Katz, M. Barker, A.-L. Lamprecht, C. Martinez, F. E. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, T. Honeyman, A. Struck, A. Lee, A. Loewe, B. van Werkhoven, C. Jones, D. Garijo, E. Plomp, F. Genova, H. Shanahan, J. Leng, M. Hellström, M. Sandström, M. Sinha, M. Kuzak, P. Herterich, Q. Zhang, S. Islam, S.-A. Sansone, T. Pollard, U. D. Atmojo, A. Williams, A. Czerniak, A. Niehues, A. C. Fouilloux, B. Desinghu, C. Goble, C. Richard, C. Gray, C. Erdmann, D. Nüst, D. Tartarini, E. Rangelova, H. Anzt, I. Todorov, J. McNally, J. Moldon, J. Burnett, J. Garrido-Sánchez, K. Belhajjame, L. Sesink, L. Hwang, M. R. Tovani-Palone, M. D. Wilkinson, M. Servillat, M. Liffers, M. Fox, N. Miljković, N. Lynch, P. Martinez Lavanchy, S. Gesing, S. Stevens, S. Martinez Cuesta, S. Peroni, S. Soiland-Reyes, T. Bakker, T. Rabemanantsoa, V. Sochat, Y. Yehudi, R. F. WG, FAIR Principles for Research Software (FAIR4RS Principles), 2022. doi:10.15497/RDA00068.
- [4] 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) 171–191.
- [5] D. Dessì, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, H. Sack, Ai-kg: an automatically generated knowledge graph of artificial intelligence, in: *International Semantic Web Conference*, Springer, 2020, pp. 127–143.
- [6] J. Malone, A. Brown, A. L. Lister, J. C. Ison, D. Hull, H. E. Parkinson, R. Stevens, The software ontology (swo): a resource for reproducibility in biomedical data analysis, curation and digital preservation., *J. Biomed. Semant.* 5 (2014) 25. URL: <http://dblp.uni-trier.de/db/journals/biomedsem/biomedsem5.html#MaloneBLIHPS14>.
- [7] M. AI, Papers with code, 2024. URL: <https://paperswithcode.com>.
- [8] 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.
- [9] 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.
- [10] M. Färber, D. Lamprecht, Linked papers with code: the latest in machine learning as an rdf knowledge graph, *arXiv preprint arXiv:2310.20475* (2023).
- [11] A. Salatino, F. Osborne, E. Motta, Cso classifier 3.0: a scalable unsupervised method for classifying documents in terms of research topics, *International Journal on Digital Libraries* (2022) 1–20.
- [12] J. T. Ciuciu-Kiss, D. Garijo, Implementation of the analysis and dataset for research software classification, [https://github.com/kuefmz/pow\\_categories](https://github.com/kuefmz/pow_categories), 2025. doi:h10.5281/zenodo.15230833, accessed: April 17, 2025.
- [13] S. M. Costa, C. Pesquita, Agreementmaker: A flexible and efficient ontology matching system, *Journal of Web Semantics* (2012).
- [14] D. Faria, C. Pesquita, I. F. Cruz, Agreementmakerlight: Boosting ontology alignment through

hybrid matching strategies, in: *Lecture Notes in Computer Science*, 2013.

- [15] A. A. Salatino, F. Osborne, E. Motta, The cso classifier: Ontology-driven detection of research topics in scholarly articles, *International Journal on Digital Libraries* (2020).
- [16] A. A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne, E. Motta, The computer science ontology: a large-scale taxonomy of research areas, in: *International Semantic Web Conference*, Springer, 2018, pp. 187–205.
- [17] J. Ison, M. Kalas, I. Jonassen, D. Bolser, M. Uludag, H. McWilliam, J. Malone, R. Lopez, S. Pettifer, P. Rice, Tools and data services registry: A community effort to document and share bioinformatics resources, *Nucleic Acids Research* 44 (2016) D38–D47. doi:10.1093/nar/gkv1116.
- [18] M. Al-Ahmad, et al., Ai knowledge graph: Large-scale knowledge graph for ai research, *Journal of Web Semantics* (2021). URL: <https://link.springer.com/article/10.1007/s10586-021-03211-4>.
- [19] P. Manghi, C. Atzori, A. Bardi, M. Baglioni, H. Dimitropoulos, S. La Bruzzo, I. Foufoulas, A. Mannocci, M. Horst, K. Iatropoulou, A. Kokogiannaki, M. De Bonis, M. Artini, A. Lempesis, A. Ioannidis, N. Manola, P. Principe, T. Vergoulis, S. Chatzopoulos, Openaire graph dataset, 2024. doi:10.5281/zenodo.12819872.
- [20] A. Rajaraman, J. D. Ullman, *Mining of Massive Datasets*, Cambridge University Press, 2011. URL: <https://infolab.stanford.edu/~ullman/mmds/book.pdf>.
- [21] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019, pp. 3982–3992.
- [22] 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.
- [23] 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.
- [24] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, *Communications in Statistics* 3 (1974) 1–27.
- [25] D. L. Davies, D. W. Bouldin, A cluster separation measure, *IEEE transactions on pattern analysis and machine intelligence* (1979) 224–227.
- [26] L. Maaten, G. Hinton, Visualizing data using t-sne, *Journal of Machine Learning Research* (2008) 2579–2605.