Reproducibility and Analysis of Scientific Dataset Recommendation Methods

Ornella Irrera
ornella.irrera@unipd.it
University of Padua
Padua, Italy

Matteo Lissandrini
matteo.lissandrini@univr.it
University of Verona
Verona, Italy

Daniele Dell'Aglio
dade@cs.aau.dk
University of Aalborg
Aalborg, Denmark

Gianmaria Silvello
gianmaria.silvello@unipd.it
University of Padua
Padua, Italy

Abstract

Datasets play a central role in scholarly communications. However, scholarly graphs are often incomplete, particularly due to the lack of connections between publications and datasets. Therefore, the importance of dataset recommendation—identifying relevant datasets for a scientific paper, an author, or a textual query—is increasing. Although various methods have been proposed for this task, their reproducibility remains unexplored, making it difficult to compare them with new approaches. We reviewed current recommendation methods for scientific datasets, focusing on the most recent and competitive approaches, including an SVM-based model, a bi-encoder retriever, a method leveraging co-authors and citation network embeddings, and a heterogeneous variational graph autoencoder. These approaches underwent a comprehensive analysis under consistent experimental conditions. Our reproducibility efforts show that three methods can be reproduced, while the graph variational autoencoder is challenging due to unavailable code and test datasets. Hence, we re-implemented this method and performed a component-based analysis to examine its strengths and limitations. Furthermore, our study indicated that three out of four considered methods produce subpar results when applied to real-world data instead of specialized datasets with ad-hoc features.

CCS Concepts

• **Information systems** → **Recommender systems**; Retrieval models and ranking.

Keywords

Dataset Recommendations, Recommender Systems, Reproducibility

ACM Reference Format:
Ornella Irrera, Matteo Lissandrini, Daniele Dell'Aglio, and Gianmaria Silvello. 2024. Reproducibility and Analysis of Scientific Dataset Recommendation Methods. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3640457.3688071

RecSys '24, October 14–18, 2024, Bari, Italy

ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688071

1 Introduction

With the growing abundance of open datasets [5], the task of dataset recommendation, wherein relevant datasets are retrieved based on a paper, author, or textual query, is gaining momentum and the demand for dataset recommendation methods is increasingly evident, especially in the context of scholarly graphs [2, 4, 8, 12, 21, 28, 30–32]. The challenge in recommending datasets stems from their diverse nature, encompassing various formats such as images, tables, CSV, XML, and RDF files. Compounding this difficulty is the absence or low quality of metadata containing a dataset description, creator, and related publications describing their content.

The dataset recommendation methods proposed so far in the scientific domain are heterogeneous and challenging to compare. In particular, the input queries can take on diverse forms, including natural language sentences [14, 28], keywords [28], publications [2, 30], and authors [8]. The outputs vary, with some dataset recommendation systems generating a ranking of datasets [2, 8, 28], while others generate a set of datasets [14, 30, 32]. Further, the methodologies employed for producing dataset recommendations may leverage the similarity between the query's and datasets' embedding representations [21, 31], the exploration of scholarly graphs [2, 8], or the combination of graph-based and textual-based approaches [30].

In this work, we examine methods designed for the dataset recommendation task, selecting the most recent, pertinent, and high-performing approaches published in relevant venues. We considered methods where the query is represented by one or more publications or a textual description. These approaches are:

- a *LinearSVM*-based approach presented at the International Conference on Information and Knowledge Management (CIKM) in 2022 [14] (ICORE[1] rank A);
- a *bi-encoder retriever* outlined in the proc. of the Association for Computational Linguistics (ACL) in 2023 [28] (rank A*);
- *Ensemble_CN*, a model leveraging co-authors and citation network introduced in the Data Science Journal (Scimago Q1 [Computer Science Applications]) [32] and in the proc. of KSEC 2022 [30] (rank C);

[1] https://www.core.edu.au/icore-portal

- the *Heterogeneous Variational Graph AutoEncoder (HVGAE)* [2] published in the IEEE International Conference on Data Mining (ICDM, rank A*) in 2019;

Research gap. We highlight that none of these approaches directly compares with each other due to differences in input, the data used for training, and/or the output they generate. Each method is typically assessed against baseline measurements not explicitly designed for the dataset recommendation task. Consequently, establishing the current state-of-the-art dataset recommendation task is challenging unless we adapt existing methods to consider the same input, training data, and output. This requires the reproducibility of the methods and the definition of shared datasets for comparison. In this work, we lay the groundwork to make the methods comparable and establish a competitive baseline for dataset recommendation.

In this regard, the main **contributions** are:

(1) A replication study (i.e., same setup, different team) of the LinearSVM, the bi-encoder, and the Ensemble_CN methods. The study shows that the LinearSVM and the bi-encoder are fully replicable, while Ensemble_CN is only partially replicable.
(2) A reproducibility study (i.e., different setup, different team). We re-implemented HVGAE and conducted an in-depth components-based study.
(3) An evaluation of the four recommendation methods on three newly created test datasets representing real-world data. We observed a performance drop in three out of four methods when tested on the new datasets rather than on the ad-hoc datasets employed in the original papers.
(4) A generalization study (i.e., different setup, team, and context) based on three baseline recommendation methods – TopPop, BPR, and LightGCN – not designed for the dataset recommendation task. We adapted these methods to recommend datasets and tested them on the new datasets. We demonstrate that, under specific experimental conditions, LightGCN outperforms the analogous reproduced methods, highlighting its untapped potential for this type of architecture.

The rest of the paper is organized as follows. Section 2 describes the relevant works on the dataset recommendation task. Section 3 presents the recommendation methods evaluated in this work [2]. In Sections 4 and 5, we describe the datasets employed in the analyses and the implementation details. We present and discuss the results in Section 6. Finally, Section 7 draws some final remarks.

2 Related Work

Many studies have proposed methods targeting the dataset recommendation task in the last decade. In this context, "*dataset*" refers to a set of related observations or results derived from research experiments [7]. The typical setup consists of a scientist who needs to identify existing datasets often adopted or presented in the available scientific literature. Hence, datasets comprise traditional databases and data files, e.g., in the CSV format, as well as scientific artefacts like tables, figures, and archives.

Most methods [14, 21, 28, 31] treat dataset recommendation as a text-centric problem modeling papers and datasets as collections of documents, with datasets presumed to have accompanying titles or short descriptive texts. Usually, queries are expressed in some textual form, such as a keyword query. Consequently, the available methods usually capitalize on the similarity between the dataset's textual metadata and a given query to formulate a ranking of recommended datasets. However, it is noteworthy that more recent scholarly databases adopt a more sophisticated representation of the scientific literature and related artefacts through a richer semantic graph structure known as scholarly knowledge graphs [3, 6, 19, 24]. Yet, some methods exploit networks interconnecting scholarly products to perform recommendations. Finally, we identify an essential common aspect: no solution directly considers the content of the datasets because their heterogeneity poses relevant challenges for recommendation [31]. Thus, the existing methods are classified as *metadata-based* or *graph-based* solutions.

Metadata-driven solutions utilize the textual metadata associated with a dataset for recommendation. Most of these approaches involve encoding the dataset metadata and queries into a shared vector space, enabling the computation of similarity to generate a ranking of recommended datasets. For example, the technique proposed by Viswanathan et al. [28] employs a text bi-encoder retriever, demonstrating its effectiveness compared to BM25 and nearest neighbors retrieval. Another approach, presented by Färber and Leisinger [14], takes a classification approach where the query consists of multiple sentences describing the user's needs. We included these two approaches in our study due to their methodological nuances and significance in the scientific community.

We excluded from the study other methodologies that were outperformed by the selected methods or focused solely on a straightforward application of BM25 or the calculation of similarity between embeddings using BERT or Doc2Vec [4, 12, 21, 28, 31]. Among these, Patra et al. [21] employs metadata-based filtering to suggest datasets to researchers based on their previous publications. In contrast, Wang et al. [31] employs ranking functions such as BM25 or relies on the similarity between vector representations obtained with Doc2Vec or BERT for the recommendation process. [31] also proposed a second method based on extracting one or more ontology concepts from the dataset metadata and determining its relevance to the query by using ontology-based similarity approaches. Lastly, Ben Ellefi et al. [4] represents a dataset with schema concept labels and ranks the datasets by computing the cosine similarity between the vector representations of the concepts.

Graph-based approaches leverage scholarly knowledge graphs interconnecting research products to perform dataset recommendations. Among the methods in the literature adopting this approach, the approach by Altaf et al. [2] recommends the most relevant datasets for a set of papers, adopting a heterogeneous variational graphs autoencoder (HVGAE) that learns the representation of papers and datasets. Chen et al. [8] proposed AMENDER, a multilayered model to recommend a set of datasets leveraging the information of a three-layers network interconnecting authors, papers, and datasets. Further methods, such as the one by Wang et al. [32], use a co-author network to recommend a set of datasets similar to the one used as a query. This work has been used in authors' subsequent publication [30], where the recommendation combines link prediction-based and ranking-based approaches.

[2]Experimental suite available at: https://github.com/HeterogeneousGL-SAN/dataset_recomm_repro

3 Dataset Recommendation Methods

Table 1 reports the reproduced methods by outlining the approach type, the dataset employed in the original papers (detailed in Section 4), the query type, the output, and the code availability.

We can see that the metadata-based methods [14, 28] take in input a free text query representing the user information need. Thus, these methods allow also to use a paper abstract to retrieve relevant datasets. The graph-based methods take as input one or more network nodes representing publications, datasets, or authors. The nodes can be accompanied by textual descriptions (e.g., title and description). As outputs, the bi-encoder, and HVGAE return a ranked list of datasets sorted by their relevance to the query. In contrast, LinearSVM and Ensemble_CN return a set of datasets relevant to the user's query without ranking them.

LinearSVM. In [14], the dataset recommendation task is treated as a supervised multiclass, multilabel text classification problem. The idea is that the more detailed the research problem description (i.e., the user's query) is, the higher the quality of the recommended datasets will be. The approach relies on different text representation methods to encode the user's query, specifically tf-idf (the most effective), doc2vec, and sciBERT. Then it applies a linear SVM to recommend a set of relevant datasets. Two types of inputs are considered to train and test the models: the papers' abstracts mentioning one or more datasets and the datasets' citation contexts in the abstracts.

Bi-Encoder Retriever. Viswanathan et al. [28] consider short textual descriptions or a set of keywords as queries. The authors performed dataset recommendation using BM25, *k*-nearest neighbors retrieval, and a bi-encoder retriever. In the present work, we consider only the bi-encoder retriever as it has been proven to be the most effective. In the proposed implementation, the bi-encoder is initialized with sciBERT and is fine-tuned on the training set. Each query and dataset's descriptions are encoded with BERT embeddings; the similarity between the embeddings is their inner product. The output of the bi-encoder is a ranked list of datasets for each query. The authors showed that using keywords or full-sentence queries achieved similar results.

Ensemble_CN. Wang et al. [30] leverage a large-scale graph interconnecting scientific papers, datasets, and authors extracted from the Microsoft Academic Knowledge Graph (MAKG) to recommend one or more scientific items. The queries are composed of scientific items in the graph. This method relies on KGlove [9] and Glove [23] to generate an embedding for each publication and dataset; then, the cosine similarity is used to measure the similarity between the embedding of the query and compute the final prediction. This method combines graph walk on the co-authors network [32] and authors' pre-trained embeddings similarity. The items are ranked according to the BM25 score and BERT-based link prediction; the ranking is based on the cosine similarity between BERT pre-trained embeddings. According to the original paper, combining all the above methods resulted in the best performances, highlighting that relying on a citation network is essential but not enough to produce effective recommendations. The output of this approach is a set of datasets corresponding to the intersection of the datasets returned from each approach. Even though this method targets a more general task involving both publications and dataset recommendations, recommending datasets can be achieved considering only publications as queries.

HVGAE. Altaf et al. [2] address the problem of recommending datasets relevant to one or more publications in a heterogeneous graph. The input of HVGAE is a heterogeneous graph composed of a bipartite network ($\mathbf{H}_{\mathcal{P},\mathcal{D}}$) interconnecting papers and datasets, and an attributed citation network interconnecting papers, where $\mathbf{A}_{\mathcal{P},\mathcal{P}}$ is the adjacency matrix, and $\mathbf{X}_{\mathcal{P}}$ identifies papers' content vector embeddings. The architecture consists of two autoencoders: the first one takes in input the adjacency matrix of the citation network $\mathbf{A}_{\mathcal{P},\mathcal{P}}$, and the papers' embeddings $\mathbf{X}_{\mathcal{P}}$ and outputs $\mathbf{A'}_{\mathcal{P},\mathcal{P}}$, which is the reconstructed adjacency matrix; the other one takes in input the adjacency matrix of the bipartite graph $\mathbf{H}_{\mathcal{P},\mathcal{D}}$, and outputs the reconstructed adjacency matrix $\mathbf{H'}_{\mathcal{P},\mathcal{D}}$. The citation network is encoded using GraphSAGE [17]; the bipartite graph, instead, is encoded relying on a densely connected multi-layer perceptron (MLP). Each encoder produces mean and variance vectors of a Gaussian distribution from which paper and dataset representations $\vec{\mathbf{z}}_p$, $\vec{\mathbf{z}}_d$ are sampled. The decoding stage includes two inner product decoders that decode the latent variables $\vec{\mathbf{z}}_p$ and $\vec{\mathbf{z}}_d$ into probabilistic dense adjacency matrices that reconstruct $\mathbf{A}_{\mathcal{P},\mathcal{P}}$, and $\mathbf{H}_{\mathcal{P},\mathcal{D}}$. For the objective function, the authors maximize the Evidence Lower Bound (ELBO) on the marginal likelihood of the observed variables $\mathbf{A}_{\mathcal{P},\mathcal{P}}$ and $\mathbf{H}_{\mathcal{P},\mathcal{D}}$ (cfr. Eq.13 in the reference paper [2].) The representation $\vec{\mathbf{z}}_Q$ of a query Q is considered as a Gaussian distribution whose mean μ_Q and variance σ_Q^2 correspond to the element-wise mean of paper vectors in $\{\vec{\mathbf{z}}_p \sim \mathcal{N}(\mu_p, \sigma_p^2), \forall p \in Q\}$. The relevance of a dataset to a query is computed via the KL-divergence between the learned representations of the query and of the datasets and used for top-k retrieval.

4 Datasets

In this section we present the datasets adopted by the original papers and the newly introduced shared datasets describing their features and code and data availability.

The **LinearSVM_Dataset** includes 1,691 dataset titles in the field of computer science extracted from the DSKG [13] and 88,000 abstracts extracted from the Microsoft Academic Graph [29]. Each of the selected abstracts referenced at least one dataset. According to the authors' analyses, most datasets are mentioned up to 100 times in the abstracts. 70% of the datasets has been left for training, 20% for testing, and 10% for validation.

The **DataFinder Dataset** contains a set of *(q, R)* pairs, where *q* is the query, either a textual description or a set of keywords, and *R* is a set of datasets relevant for that query. DataFinder contains 17,5K queries used for training the models and 392 for testing. The datasets in DataFinder are collected from *Papers With Code*, [4] a large index of papers including metadata. The queries in the training set are automatically formulated using Galactica [27] and are based on paper abstracts; the relevant datasets have been extracted with a rule-based procedure. For the test set, pairs of queries and relevant datasets are manually annotated.

MAKG_CN is an RDF graph of interconnected publications (3M), datasets (1.5K), and authors (5M) extracted from the MAKG and ScholeXplorer. Analyzing the MAKG_CN dataset, we verified

[4] https://paperswithcode.com/

Table 1: Overview of type, dataset, query, output, and availability of all the methods examined.

Method	Type	Dataset	Query	Output	Availability
LinearSVM [14]	metadata-based	LinearSVM_Dataset: MAG [29] + DSKG [13, 16]	Textual description	Set of datasets	✓
Bi-Encoder Retriever [28]	metadata-based	DataFinder Dataset [28]	keywords, textual description	Datasets ranking	✓
Ensemble_CN [30]	graph-based	MAKG_CN: MAKG [15, 29] + ScholeXplorer[3]	Publications, datasets	Set of datasets	✓
HVGAE [2]	graph-based	Delve [1]	Publications	Datasets ranking	✗

Table 2: Overview of employed datasets. We report the number of publications (P), datasets (D), authors (A), publication-dataset edges (p → d), publication-publication (p → p), and dataset-dataset (d → d) edges.

Dataset	P	D	A	p→d	p→p	d→d
MES	2,181	2,949	9,408	3,098K	500	1,182
PubMed_KCore2	2,563	3,291	19,655	5,503	1,055	1,504
PubMed	33,797	42,635	330,000	37,462	18,917	8,078

that only 1,500 triples involve publications connected to datasets. In particular, 1,400 distinct publications are connected to at least one dataset, and 257 datasets to at least one publication. Only 11,600 publications and 326 datasets have a textual description.

MES [19] is a curated scholarly graph representing the European Marine Science Community of OpenAIRE [5]. It interconnects publications (4K), datasets (5,5K), authors (21,6K), and software (not considered in this work). All the publication and dataset nodes have a set of textual attributes that include the abstract, the title, the URL (or the DOI), and the publication date; the author's nodes contain names, surnames, and ORCID (if present). We processed and enriched the original MES dataset [19] with a citation network interconnecting publications to publications required by Ensemble_CN, and HVGAE.

PubMed has been extracted from the OpenAIRE Graph and interconnects a set of publications available on PubMed (33,8K), the related datasets (42,6K), and their authors (330K). Publications, datasets, and authors have the same attributes as the nodes in the aforementioned MES dataset. Authors have been disambiguated with the FDup framework [11].

PubMed_KCore2 represents a subset of the aforementioned PubMed dataset (2,5K publications, 3,3K datasets and 19,6K authors). It has been created by selecting all the publications (datasets) with a minimum of two interconnected datasets (publications) and includes all the associated authors.

A more comprehensive description of nodes and edges count in MES, PubMed_KCore2 and PubMed datasets used in the present paper is provided in Table 2. In these datasets, each publication is connected to at least one dataset. We analyzed the impact of authors' disambiguation on MES, PubMed, and PubMed_KCore2. We detected that in MES, on average, each author contributed to 3.2 research outcomes (i.e., publications, datasets); conversely, in PubMed_KCore2 and PubMed, we detected that each author contributed to 1 and 1.2 outcomes, respectively. In addition to this, in MES there are $9K$ authors connected both to at least one publication and a dataset, while in PubMed_KCore2 and PubMed only 866 and $72K$ respectively. These characteristics, typical of real-world datasets, can significantly impact the final performance of methods as they contribute to the sparsity of the graph.

[5]https://mes.openaire.eu/

5 Implementation Details

We utilized the training-test splits provided in the original papers to perform replicability experiments. In MES, PubMed_KCore2 and PubMed datasets, we left 80% of the edges for training and used the remaining 20% for validation and test.

Replicability. Replicability involves LinearSVM, bi-encoder, and Ensemble_CN methods. To replicate the results of the linearSVM-based approach, we relied on the code and data provided in the GitHub repository; [6] the authors relied on the LinearSVM implementation provided by the `scikit-learn` python library [22]. Specifically, among the solutions proposed, we replicated the most effective one: tf-idf for text representation and LinearSVM as selected model.

The bi-encoder retriever relies on the Tevatron[7] python package. To replicate its evaluation on the full-sentence and keywords queries experiments, we contacted the authors of the paper, who fixed the code repository. [8]

While replicating the Ensemble_CN method we noticed that one of the files on GitHub[9] was corrupted, and the code could not run properly. We contacted the authors of the paper and we managed to run the code even though not in all the settings presented in the original paper as detailed in the results section.

Reproducibility. The original code and data for the HVGAE approach are inaccessible. Despite contacting the authors, we could not obtain the code nor the training and test data used in the original experiments. Consequently, we re-implemented this approach from scratch relying on the `PyTorch`[10] and `PyTorch Geometric`[11] python libraries.

The original paper for HVGAE does not provide details about how the initial metadata representations were obtained. Consequently, we computed the representations using `all-MiniLM-L6-v2`, a pre-trained language model available on Hugging Face[12] that generates sentence and paragraphs embeddings with 384 features. We first ran the re-implemented version of HVGAE on MES, PubMed, and PubMed_KCore2 relying on the initial combination of parameters provided in the original paper –i.e., 110 epochs, learning rate equal to 10^{-4}, L2 regularization equal to

[6]https://github.com/michaelfaerber/datarec
[7]https://github.com/texttron/tevatron
[8]https://github.com/viswavi/datafinder/tree/main
[9]https://github.com/xuwang0010/datarecommend
[10]https://pytorch.org/
[11]https://pytorch-geometric.readthedocs.io/en/latest/
[12]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

10^{-3}. Running randomized search, we found that the most effective combination of hyperparameters was instead 500 epochs, learning rate 10^{-2}, L2 regularization 10^{-5}. While in the original implementation, a query could include more than one paper, in our evaluation setting, each query corresponds to precisely one paper. To test the system, we first obtained the query papers representations by leveraging the GraphSAGE encoder, and then, we computed the KL-divergence between each query and dataset representations.

Generalization. We selected three general-purpose recommendation methods – TopPop [10], BPR [25], and LightGCN [18] – not targeting the dataset recommendation task and analyzed how they perform on the shared datasets in our settings. We relied on the implementation of these methods provided by Cornac [26], a framework for multimodal recommender systems. Given that BPR and LightGCN methods distinguish between users and items, to run these models, we considered publications as users and datasets as items.

BPR [25] (Bayesian Personalized Ranking) is a collaborative filtering model designed for recommendation systems that work in implicit feedback settings. It takes in input the adjacency matrix describing the interaction between the publications (i.e., the users) and datasets (i.e., the items). Rather than predicting a precise rating for each dataset, the method predicts the relative publication preferences for all publication-dataset pairs. The model learns to predict a higher ranking for datasets with implicit positive feedback compared to non-interacted datasets, thus providing personalized recommendations based on implicit publication preferences. It is commonly used as a strong baseline in similar settings: it has been used as baseline also in HVGAE original evaluations [2].

LightGCN [18] is a transductive approach that learns publication-dataset–i-e., user-item–embeddings by linearly propagating them on the publication-dataset interaction graph. The final embedding for prediction is derived by combining the embeddings from the propagation layers through a weighted sum, capturing information from different layers in the process. The prediction is based on the inner product between the publication and the dataset.

We relied on grid search for hyperparameters tuning. We found that for BPR the best results are achieved with a learning rate of 0.1, a L2 regularization parameter of 0.001, and at most 300 iterations. For LightGCN instead, the highest effectiveness is reached with a learning rate of $1e-3$, 2 layers, 500 iterations, and a batch size of 1024.

Additionally, we evaluated BM25 and cosine similarity-based rankings to assess the performance of these baselines across the three shared datasets. These baselines, mentioned in the original papers of the methods under study, are points of comparison for the respective approaches.

For the BM25 implementation, we relied on the one provided by the `rank_bm25`[13] python library. For the ranking based on the cosine similarity instead, we first computed the embeddings using the `all-MiniLM-L6-v2`[14] sentence-transformer model, and then we computed the cosine-similarity between the query and the datasets embeddings.

[13]https://github.com/dorianbrown/rank_bm25

[14]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Table 3: Replicability results. We provide the results obtained replicating each experiment (*Replicated*), the results reported in the reference paper for that experiment (*Original*), and the difference between these values (*Difference*).

Method [dataset]	Experiment	Precision	Recall	F1
LinearSVM [LinearSVM_Dataset]	Abstract - Original	0.3900	0.1700	0.2200
	Abstract - Replicated	0.3700	0.1900	0.2300
	Abstract - Difference	-0.0200	0.0200	0.0100
	Citation - Original	0.6200	0.5500	0.5700
	Citation - Replicated	0.6000	0.5500	0.5600
	Citation - Difference	-0.0200	0.0000	-0.0100
Bi-encoder [DataFinder Dataset]	Sentences - Original	0.1600	0.3120	0.2115
	Sentences - Replicated	0.1705	0.3355	0.2262
	Sentences - Difference	0.0105	0.0235	0.0147
	Keyphrase - Original	0.1650	0.3240	0.2186
	Keyphrase - Replicated	0.1649	0.3260	0.2184
	Keyphrase - Difference	-0.0001	0.0020	-0.0002
Ensemble_CN[MAKG_CN]	Exp 1 - Original	0.7600	0.6400	0.1180
	Exp 1 - Replicated	0.7600	0.6390	0.1180
	Exp 1 - Difference	0.0000	0.0010	0.0000
	Exp 2 - Original	0.6950	0.0176	0.0343
	Exp 2 - Replicated	0.2662	0.0007	0.0015
	Exp 2 - Difference	-0.4228	-0.0169	-0.0328
	Exp 3 - Original	0.6404	0.0315	0.0600
	Exp 3 - Replicated	0.2456	0.0010	0.0021
	Exp 3 - Difference	-0.3948	-0.0315	-0.0579

6 Experimental Results and Discussion

In this section we discuss the replicability, reproducibility, generalizability results.

Replicability results. Table 3 reports the results of the replicability analyses. Each method has been evaluated on its original datasets when possible using the original evaluation metrics.

Both LinearSVM and bi-encoder retriever are entirely replicable, following the guidelines provided on the original repositories. Only minor variations were observed in comparison to the original results.

On the contrary, the Ensemble_CN approach is partially replicable. Indeed, in the Ensemble_CN paper, the authors presented three experiments, each one characterized by two sets of seeds, i.e., publications and datasets nodes – one for queries and one for candidates; we have observed that the seeds and candidate sizes of the second and third experiments were more than ten times greater than those mentioned in the paper. Even with the help of the authors (contacted via email), it was impossible to solve the data discrepancy. For the cases where the experiment seeds and candidate sizes did not match, the code did not terminate the execution. To overcome this issue, we ran the experiments using the numbers indicated in the paper rather than the repository. Moreover, the original paper does not provide details on how the reported results were calculated, and the authors could not provide this information. We could replicate the first experiment, finding out the authors used an unorthodox definition of mean precision and recall in the original paper. Indeed, precision and recall were calculated first by counting all the relevant datasets retrieved in each query and dividing them by all retrieved datasets. On the other hand, we could

Table 4: LinearSVM Reproducibility. For each dataset, we report Precision, Recall and F1 score.

Dataset	P	R	F1
MES	0.030	0.020	0.020
PubMed_KCore2	0.080	0.080	0.080
PubMed	-	-	-
LinearSVM_Dataset	0.390	0.170	0.230

Table 5: Bi-encoder retriever reproducibility results. For each test data, we report nDCG@5, P@5, R@5.

Dataset	nDCG@5	P@5	R@5
MES	0.380	0.084	0.363
PubMed_KCore2	0.301	0.098	0.401
PubMed	0.377	0.097	0.469
DataFinder Dataset	0.160	0.211	0.320

not replicate the second and third experiments in the original paper (we obtained a precision of approximately 0.4 lower).

Reproducibility results.

LinearSVM. To test the LinearSVM on the shared datasets, we extracted from MES, PubMed _KCore2, and PubMed the interconnected publications and datasets along their textual attributes. In our experiments, we relied on tf-idf text representation. The textual descriptions of publications were treated as queries for our analyses. The results of LinearSVM on MES, PubMed_KCore2, and PubMed data are reported in Table 4. The last row contains the results reported in the original paper and obtained running LinearSVM on the LinearSVM_Dataset. On MES and PubMed_KCore2, LinearSVM achieved the lowest performance – precision and recall equal to 0.03 and 0.02 on MES and 0.08 and 0.08 on PubMed_KCore2, respectively. LinearSVM treats the task as a multi-class, multi-label text classification problem. Yet, in our three test datasets, the number of labels (i.e., the datasets) is larger than the number of classes (i.e., papers' abstracts). We postulate that this might be the most impacting challenge to the performance of the classification models. This condition is never met in the original data, where the number of classes ($88K$) largely exceeded the number of labels ($1, 6K$). Running the experiment on PubMed data instead did not produce results as LinearSVM does not handle large amounts of datasets and abstracts.

Bi-encoder retriever. Table 5 reports the experimental results for the bi-encoder, where, similarly to the LinearSVM experimental setup, we considered the publications' abstracts as queries. In the last row, we report the results obtained running the bi-encoder on full-sentence queries on the DataFinder Dataset. The bi-encoder performs well across MES, PubMed, and PubMed_KCore2, showing higher nDCG and recall than the DataFinder dataset. However, its precision remains consistently below 0.1 in all tested datasets, lower than reported in the original paper. We also tested the effectiveness of the bi-encoder providing shorter queries, i.e., the title in place of the description, but we did not detect significant differences.

Table 6: Ensemble_CN reproducibility results. For each test data, we report Precision, Recall and F1 score.

Dataset	P	R	F1
MAKG_CN-datarec	0.222	0.020	0.020
MES	0.652	0.258	0.370
PubMed_KCore2	0.035	0.015	0.021
PubMed	0.010	0.005	0.007
MAKG_CN	0.640	0.031	0.060

Table 7: HVGAE reproducibility results. For each test data, we report nDCG@5, P@5, R@5.

Dataset	nDCG@5	P@5	R@5
MES	0.014	0.008	0.041
PubMed_KCore2	0.023	0.014	0.067
PubMed	0.006	0.004	0.027
Delve	0.715	0.184	0.922

Ensemble_CN. Table 6 reports the results of the Ensemble_CN method; the last row presents the results reported in the original paper for the third experiment (cfr. Table 7 in [30]) evaluated on the MAKG_CN dataset. We first analyzed how this method performs when the queries are publications and the recommendation contains only datasets (MAKG_CN-datarec row in Table 6). MAKG_CN-datarec is obtained by extracting from the MAKG_CN dataset a new set of seeds containing 129 publications randomly selected from the publications connected to at least one dataset – i.e., the queries, and a set of 257 datasets used as candidates for the recommendation.

When restricting recommendations solely to datasets, there is a noticeable decrease in effectiveness, with precision at 0.222. In contrast, the original paper, recommending both publications and datasets, reported higher precision across all experiments. This drop in effectiveness can be attributed to several factors. Firstly, a limited number of datasets (only 326 out of $1.5K$) have associated descriptions, negatively impacting text-based recommendation models like BERT and BM25. Additionally, datasets often exhibit lower connectivity than publications, typically part of a dense citation network. As a result, this method struggles to effectively leverage the citation network for dataset recommendations, particularly when compared to more interconnected and extensively described items. Author connectivity is a crucial aspect. A prior study on dataset mentions and citations in scientific publications [20] found that most connected publications and datasets share at least one author. Consequently, exploring author embeddings and the co-author network becomes essential for effective recommendations. However, for MAKG_CN-datarec, the absence of dense co-author networks, attributed to either the lack of author information linked to datasets or issues with author name disambiguation, further contributes to the method's failure in recommending datasets. As a result, optimal performances are observed with the MES dataset, while the PubMed_KCore2 and PubMed datasets exhibit lower performance. This discrepancy is attributed to the increased sparsity in PubMed_KCore2 and PubMed datasets. Furthermore, most authors remain undistinguished, restricting the effectiveness of citation and co-author network exploration.

Table 8: HVGAE component-based analyses results.

	VAE Cit. Net.			VAE Bip. Net (mlp)			VAE Bip. Net (gcn)			VAE Bip. Net (sage)		
Dataset	nDCG@5	P@5	R@5	nDCG@5	P@5	R@5	nDCG@5	P@5	R@5	nDCG@5	P@5	R@5
MES	0.093	0.054	0.262	0.100	0.070	0.219	0.095	0.066	0.222	0.077	0.054	0.187
PubMed_KCore2	0.015	0.010	0.045	0.079	0.050	0.212	0.065	0.041	0.175	0.071	0.045	0.189
PubMed	0.013	0.008	0.036	0.026	0.016	0.070	0.036	0.022	0.103	0.020	0.012	0.053
$MES_{enriched}$	0.119	0.073	0.326	0.084	0.062	0.195	0.095	0.066	0.225	0.072	0.045	0.194
$PubMed\_KCore2_{enriched}$	0.017	0.010	0.045	0.080	0.050	0.212	0.111	0.070	0.294	0.100	0.062	0.265
$PubMed_{enriched}$	0.010	0.006	0.029	0.032	0.019	0.085	0.039	0.025	0.101	0.056	0.035	0.145
$MES_{reduced}$	0.039	0.025	0.100	0.215	0.141	0.513	0.196	0.133	0.476	0.068	0.049	0.143
$PubMed\_KCore2_{reduced}$	0.013	0.008	0.040	0.095	0.061	0.251	0.074	0.047	0.194	0.058	0.035	0.160
$PubMed_{reduced}$	0.002	0.001	0.007	0.049	0.032	0.115	0.069	0.044	0.175	0.003	0.002	0.008

Table 9: HVGAE results considering different bipartite networks autoencoders.

	HVGAE (mlp)			HVGAE (gcn)			HVGAE (sage)		
Dataset	nDCG@5	P@5	R@5	nDCG@5	P@5	R@5	nDCG@5	P@5	R@5
MES	0.014	0.008	0.041	0.035	0.020	0.104	0.026	0.016	0.070
PubMed_KCore2	0.023	0.014	0.067	0.080	0.050	0.218	0.051	0.032	0.139
PubMed	0.006	0.004	0.027	0.007	0.004	0.021	0.009	0.005	0.023
$MES_{enriched}$	0.014	0.008	0.041	0.039	0.025	0.104	0.011	0.008	0.031
$PubMed\_KCore2_{enriched}$	0.016	0.010	0.042	0.067	0.042	0.183	0.038	0.024	0.100
$PubMed_{enriched}$	0.006	0.005	0.012	0.006	0.003	0.016	0.011	0.007	0.027
$MES_{reduced}$	0.054	0.033	0.145	0.060	0.037	0.173	0.010	0.062	0.260
$PubMed\_KCore2_{reduced}$	0.015	0.009	0.042	0.060	0.038	0.163	0.052	0.033	0.135
$PubMed_{reduced}$	0.006	0.004	0.018	0.004	0.002	0.012	0.003	0.003	0.009

HVGAE. Table 7 presents the result of the re-implemented HVGAE on MES, PubMed, and PubMed_KCore2. Notably, the method exhibits non-reproducibility, with all the datasets achieving nDCG and recall lower than 0.1. A key factor contributing to these outcomes is the dataset definition. The original HVGAE paper utilized a training dataset comprising $8K$ publications, $5K$ datasets, and network edges totaling $360K$ in the citation network and $14K$ in the bipartite graph. This graph's density, at 0.0044, exceeds that of our test datasets. Another potential factor impacting our implementation's overall performance relates to queries. In the original paper, authors extracted 426 queries, each represented by a set of papers, varying in size from one to more than six papers. The authors noted that recommendations for queries with multiple papers are more effective due to the strengthened representation of a topic across multiple papers. In our case, having queries composed of a single paper may affect the system's effectiveness.

We conducted a components-based analysis on HVGAE to assess the efficacy of two autoencoders and their influence on reproducibility. Each autoencoder underwent separate training to evaluate its ability to reconstruct the adjacency matrices of the citation and bipartite networks. Results of this analysis are presented in Table 8. The first autoencoder was focused on reconstructing the publications' citation network (VAE Cit. Net. in Table 8), while the second autoencoder was focused on reconstructing the bipartite network of publications and datasets (VAE Bip. Net. in Table 8). We first tested the performances of each autoencoder on the three shared datasets (rows 1-3 in Table 8). This is the *standard* setup. Notably, the publications' citation network autoencoder (VAE Cit. Net. in Table 8) achieved the highest performance on the MES dataset with recall exceeding 0.25. However, its performance on PubMed and PubMed_KCore2 was consistently below 0.1. This is likely due to the sparse citation networks in these datasets, which makes it challenging for GraphSAGE to perform effectively. The bipartite network autoencoder (VAE Bip. Net. (mlp) in Table 8), used also in the original implementation, showed higher performances on MES and PubMed_KCore2 with a recall exceeding 0.2. However, it achieved low performances on PubMed. The sparse nature of shared datasets significantly impacted the second autoencoder, which processed a sparse adjacency matrix representing connections between publications and datasets. This sparsity posed challenges during training,

making capturing meaningful patterns in the sparse data difficult. To address this, we explored whether incorporating a set of features extracted from the textual attributes of datasets could enhance the second autoencoder's performance. We replaced the multi-layer perceptron encoder with two layers of graph convolutional neural networks (VAE Bip. Net (gcn) in Table 8) and a GraphSAGE encoder (VAE Bip. Net (sage) in Table 8). VAE Bip. Net. (gcn) approximated the results of VAE Bip. Net. (mlp) across all datasets, whereas VAE Bip. Net. (sage) showed decreased performance. This is because the GCN considers all neighbors of a node, while GraphSAGE samples a subset of neighbors, thus reducing the representative sets of nodes for each publication/dataset.

We tested the performances of each of the presented autoencoders on two alternative setups: *enriched* and *reduced*. In the *enriched* setup (rows 4-6 in Table 8), we enriched the citation network in input to the VAE Cit. Net with $p \rightarrow d$ edges, and the bipartite network in input to the VAE Bip. Net. with $p \rightarrow p$, and $d \rightarrow d$ edges. These connections serve to expand the neighborhood of each node, thereby improving the representations of the nodes employed in the autoencoders, while mitigating graph sparsity. In the *reduced* setup (rows 7-9 in Table 8), we excluded from the training set of each dataset all $p \rightarrow d$ edges where neither p nor d appeared in the validation and test sets. This setup gives us information about how noisy individual $p \rightarrow d$ connections are. To construct the reduced setup, we removed 2,358, 1,986, 29,076 $p \rightarrow d$ edges from MES, PubMed_KCore2 and PubMed datasets, respectively.

In the *enriched* setup, both the VAE Cit. Net. and VAE Bip. Net. (sage) autoencoders achieve the highest performance compared to the standard and *reduced* setups (rows 1-3 and 7-9 in Table 8). This suggests that denser neighborhoods can enhance the effectiveness of these models. However, the performances of VAE Bip. Net. (mlp) and VAE Bip. Net. (gcn) did not significantly improve compared to the standard setup.

In the *reduced* setup instead, the VAE Cit. Net. achieved the lowest performances on all the datasets: this is due to the fact that the number of considered publications decreased and the models had not enough data to learn. The VAE Bip. Net. (mlp) achieved the highest performances across all the datasets, and the VAE Bip. Net. (gcn) achieved the highest performances on MES and PubMed data. In VAE Bip. Net (sage) instead, there is a performance decrease.

Table 9 presents the analysis of HVGAE across the three setups, combining the original VAE Cit. Net. with three different VAE Bip. Net. models defined above. Specifically, HVGAE (mlp) utilizes the VAE Bip. Net. (mlp) from the original paper, HVGAE (gcn) employs the VAE Bip. Net. (gcn), and HVGAE (sage) utilizes the VAE Bip. Net. (sage). We can see that HVGAE (gcn) always performs better than HVGAE (mlp) and HVGAE (sage). In contrast, HVGAE (mlp) achieves the worst performance. HVGAE (sage) performances are between the HVGAE (mlp) and the HVGAE (gcn) in all the setups, except for the $PubMed_{reduced}$ where it performed worse than HVGAE (mlp). These results show that relying on dense attributed networks can improve the overall performance of the HVGAE model. However, we can see that the performance decreases across all the datasets when two autoencoders (one for the publications' citation network and one for the bipartite one) are combined (Table 9) and, recommending datasets relying exclusively on one VAE Bip. Net. autoencoder (Table 8) is always more effective. This inefficacy is rooted in our training and test sets. According to our interpretation of HVGAE, each query is constructed by exploring the references list of a paper in the test set and selecting papers already present in the list of 8,503 preprocessed publications. Subsequently, the authors compute the mean of all representations of the papers in the references list to generate the final query representation. However, this implementation generates the query representation by aggregating vectors of papers that were created during the training phase. In our implementation, the representation of each query paper was generated using the trained model, specifically by feeding the citation network, composed of query papers and their connections, into the GraphSAGE encoder. This underscores that the method struggles to generalize to new, unseen data, as performance decreases when utilizing vector representations of papers that have never been encountered.

Table 10: Generalization. For each dataset, we report nDCG@5, P@5, R@5 for three recommendation methods.

Method	Dataset	nDCG@5	P@5	R@5
TopPop	MES	0.060	0.020	0.083
	PubMed_KCore2	0.002	0.001	0.004
	PubMed	0.007	0.003	0.011
BPR	MES	0.042	0.008	0.042
	PubMed_KCore2	0.239	0.070	0.315
	PubMed	0.000	0.000	0.000
LightGCN	MES	0.042	0.008	0.042
	PubMed_KCore2	0.434	0.110	0.472
	PubMed	0.006	0.001	0.006

Generalization results. In Table 10, we present the results obtained evaluating LightGCN, BPR and TopPop methods on our test datasets.

TopPop, unsurprisingly, exhibits lower effectiveness. This is because it does not perform a personalized recommendation (i.e., results are query-independent), while the ground truth ratings showcase slight popularity bias.

For BPR, we observe that it achieves the highest performance on the PubMed_KCore2 dataset (recall > 0.3); on the contrary, on MES dataset, the performance on nDCG, precision and recall are always lower than 0.05, while on PubMed dataset they are equal to 0.

Similar considerations can be drwan for LightGCN. LightGCN shows good results when applied to PubMed_KCore2 dataset, but there is a performance decrease with MES and PubMed. This could be attributed to the denser bipartite network present in PubMed_KCore2. Conversely, in MES and PubMed, a significant portion of publications is connected to at most one dataset. These relationships may introduce noise, consequently leading to a decrease in performance.

One of the main reasons why standard recommendation methods usually do not work well for the dataset recommendation task is that, typically, a dataset has been utilized by a tiny number of publications. This makes it difficult to rely on models based on user–i.e., publication–interaction with multiple items–i.e., datasets.

We conducted a further analysis on two standard search baselines: BM25 and cosine similarity-based. The results are reported in Table 11. We see that both methods outperform all the methods selected in the present paper, confirming that relying solely on textual

Table 11: BM25 and cosine similarity-based results. For each test data, we report nDCG@5, P@5, R@5.

	BM25			cosine-based similarity		
Dataset	nDCG@5	P@5	R@5	nDCG@5	P@5	R@5
MES	0.166	0.070	0.227	0.459	0.245	0.629
PubMed_KCore2	0.262	0.080	0.356	0.437	0.260	0.532
PubMed	0.209	0.077	0.283	0.316	0.176	0.421

content, when available, can be more effective than those exploiting the topology of the scholarly graph. This effectiveness is attributed to the fact that scholarly graphs are usually very sparse, and there are few datasets that are truly relevant to a given publication.

7 Final Remarks

In this study, we considered four metadata-based and graph-based approaches for dataset recommendation analyzed for the first time under the same experimental conditions. Our replicability analysis showed that only the LinearSVM-based approach and the bi-encoder are fully replicable, while the Ensemble_CN is only partially replicable. In this respect, contacting the original authors of the original papers was crucial for ensuring replicability.

Our extended analyses on reproducibility uncovered some important observations. First, the currently available methods do not fully capture the complexity of real-world data. The most relevant examples are the HVGAE and Ensemble_CN methods, whose performances depend on the availability of large and dense citation and co-author networks, rarely available in real scenarios.

The component-based analyses conducted on HVGAE demonstrated that, on real-world data, relying on a bipartite network variational autoencoder (VAE) is more effective than the proposed HVGAE, pointing to the fact that less complex approaches, even when not targeting the dataset recommendation task, can be more effective. The analyses conducted on three recommendation-based approaches – LightGCN, BPR, and TopPop – showed that LightGCN outperforms the reproduced baselines when applied to dense bipartite networks as PubMed_KCore2.

Acknowledgments

This work is partially supported by the HEREDITARY Project, as part of the European Union's Horizon Europe research and innovation programme under Grant Agreement No GA 101137074 and funded by the EC H2020 project OpenAIRE-Nexus (Grant Agreement No. 101017452).

References

[1] U. Akujuobi and X. Zhang. 2017. Delve: a dataset-driven scholarly search and analysis system. *ACM SIGKDD Explorations Newsletter* 19, 2 (2017), 36–46.

[2] B. Altaf, U. Akujuobi, L. Yu, and X. Zhang. 2019. Dataset Recommendation via Variational Graph Autoencoder. In *2019 IEEE International Conference on Data Mining, ICDM 2019, Beijing, China, November 8-11, 2019*. IEEE, 11–20. https://doi.org/10.1109/ICDM.2019.00011

[3] S. Auer, A. Oelen, M. Haris, M. Stocker, J. D'Souza, K. E. Farfar, L. Vogt, M. Prinz, V. Wiens, and M. Y. Jaradeh. 2020. Improving access to scientific literature with knowledge graphs. *Bibliothek Forschung und Praxis* 44, 3 (2020), 516–529.

[4] M. Ben Ellefi, Z. Bellahsene, S. Dietze, and K. Todorov. 2016. Dataset recommendation for data linking: An intensional approach. In *European semantic Web conference*. Springer, 36–51.

[5] D. Brickley, M. Burgess, and N. Noy. 2019. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In *The World Wide Web Conference*. 1365–1375.

[6] P. Buneman, D. Dosso, M. Lissandrini, and G. Silvello. 2021. Data citation and the citation graph. *Quantitative Science Studies* 2, 4 (2021), 1399–1422.

[7] A. Chapman, E. Simperl, L. Koesten, G. Konstantinidis, L. D. Ibáñez, E. Kacprzak, and P. Groth. 2020. Dataset search: a survey. *The VLDB Journal* 29, 1 (2020), 251–272.

[8] Y. Chen, Y. Wang, Y. Zhang, J. Pu, and X. Zhang. 2019. Amender: an attentive and aggregate multi-layered network for dataset recommendation. In *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 988–993.

[9] M. Cochez, P. Ristoski, S. P. Ponzetto, and H. Paulheim. 2017. Global RDF vector space embeddings. In *The Semantic Web–ISWC 2017: 16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I 16*. Springer, 190–207.

[10] P. Cremonesi, Y. Koren, and R. Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*. 39–46.

[11] M. De Bonis, P. Manghi, and C. Atzori. 2022. FDup: a framework for general-purpose and efficient entity deduplication of record collections. *PeerJ Computer Science* 8 (2022), e1058.

[12] M. Färber and L. Ao. 2022. The Microsoft Academic Knowledge Graph enhanced: Author name disambiguation, publication classification, and embeddings. *Quantitative Science Studies* 3, 1 (2022), 51–98. https://doi.org/10.1162/qss_a_00183

[13] M. Färber and L. Lamprecht. 2021. The data set knowledge graph: Creating a linked open data source for data sets. *Quantitative Science Studies* 2, 4 (2021), 1324–1355. https://doi.org/10.1162/qss_a_00161

[14] M. Färber and A. K. Leisinger. 2021. Recommending Datasets for Scientific Problem Descriptions. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 3014–3018. https://doi.org/10.1145/3459637.3482166

[15] M. Färber. 2021. *The Microsoft Academic Graph in RDF: A Linked Data Source with 8 Billion Triples of Scholarly Data*. https://doi.org/10.5281/zenodo.4617285

[16] M. Färber and D. Lamprecht. [n. d.]. *Data Set Knowledge Graph (DSKG)*. https://doi.org/10.5281/zenodo.4478921

[17] W. Hamilton, Z. Ying, and J. Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[18] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[19] O. Irrera, A. Mannocci, P. Manghi, and G. Silvello. 2023. A Novel Curated Scholarly Graph Connecting Textual and Data Publications. *ACM Journal of Data and Information Quality* (2023).

[20] O. Irrera, A. Mannocci, P. Manghi, and G. Silvello. 2023. Tracing Data Footprints: Formal and Informal Data Citations in the Scientific Literature. In *International Conference on Theory and Practice of Digital Libraries*. Springer, 79–92.

[21] B. G. Patra, K. Roberts, and H. Wu. 2020. A content-based dataset recommendation system for researchers—a case study on Gene Expression Omnibus (GEO) repository. *Database* 2020 (2020), baaa064.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[23] J. Pennington, R. Socher, and C. D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[24] S. Pestryakova, D. Vollmers, M. A. Sherif, S. Heindorf, N. Saleem, D. Moussallem, and A-D N. Ngomo. 2022. Covidpubgraph: A fair knowledge graph of covid-19 publications. *Scientific Data* 9, 1 (2022), 389.

[25] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[26] A. Salah, Q.T. Truong, and H. W. Lauw. 2020. Cornac: A comparative framework for multimodal recommender systems. *Journal of Machine Learning Research* 21, 95 (2020), 1–5.

[27] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, and R. Stojnic. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085* (2022).

[28] V. Viswanathan, L. Gao, T. Wu, P. Liu, and G. Neubig. 2023. DataFinder: Scientific Dataset Recommendation from Natural Language Descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*. Association for Computational Linguistics, 10288–10303. https://doi.org/10.18653/v1/2023.acl-long.573

[29] K. Wang, Z. Shen, C. Huang, C. Wu, Y. Dong, and A. Kanakia. 2020. Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies* 1, 1 (2020), 396–413. https://doi.org/10.1162/qss_a_00021

[30] X. Wang, F. Van Harmelen, M. Cochez, and Z. Huang. 2022. Scientific Item Recommendation Using a Citation Network. In *Knowledge Science, Engineering and Management - 15th International Conference, KSEM 2022, Singapore, August 6-8, 2022, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 13369)*. Springer, 469–484. https://doi.org/10.1007/978-3-031-10986-7_38

[31] X. Wang, F. van Harmelen, and Z. Huang. 2021. Biomedical Dataset Recommendation. In *DATA*. 192–199.

[32] X. Wang, F. Van Harmelen, and Z. Huang. 2022. Recommending scientific datasets using author networks in ensemble methods. *Data Science* 5, 2 (2022), 167–193. https://doi.org/10.3233/ds-220056