ISSN 2007-9737

Knowledge Injection for Field of Research Classification and Scholarly Information Processing

Raia Abu Ahmad*, Georg Rehm

Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, Berlin, Germany

{raia.abu_ahmad, georg.rehm}@dfki.de

Abstract. This paper aims to address the need for efficient classification of scholarly articles into their respective fields considering the growing volume of scientific research. We address this by exploring the application of Semantic Web resources, such as DBpedia, to represent classes in deep learning models using knowledge graph embeddings. We construct a dataset comprising publications from 123 fields of research using openly available resources. Models are then trained using different ways to semantically represent class labels via an automatic entity linking approach to DBpedia. We assess the impact of different publication metadata combinations, including titles, abstracts, authors and publishers, on the performance of these models. We find that general pre-trained knowledge graph embeddings suffer from a noise problem when applied to research classification tasks, with textual descriptions of DBpedia entities emerging as a more effective means of representing classes. Notably, we notice that titles and abstracts alone, without additional metadata features such as authors and publishers, provide the best-performing representation for publication metadata. This study fills a gap in the literature by demonstrating the efficacy of deep learning methods and Semantic Web resources like DBpedia in the classification of scholarly articles across various fields of research. Our findings underscore the importance of considering the impact of different metadata features on model performance, as well as using textual descriptions of DBpedia entities as class representations.

Keywords. Field of research classification, knowledge injection, scholarly article processing, scientific text processing, knowledge graphs, semantic web.

1 Introduction

We have been experiencing an exponential surge in scientific research recently, as the volume of academic publications is doubling every 15-17 years [12, 4]. Consequently, there has been a concerted effort to develop repositories and digital libraries to capture and organise scientific contributions. Examples include the Semantic Scholar Academic Graph (S2AG) [20], Crossref [14], and the Open Research Knowledge Graph (ORKG) [17].

An essential task for such repositories is classifying scientific artefacts into their respective fields of research (FoR), which can be then utilised in downstream applications such as bibliometric analyses and scientific search engines. Classification is usually done using supervised or unsupervised models.

The first relies on annotated articles or journals, while the second clusters (meta-)data of publications according to pre-defined criteria without needing labelled datasets. Many current FoR classification systems are limited.

In terms of supervised methods, limitations are usually manifested in the used FoR vocabulary, which does not cover fine-grained hierarchical fields, while unsupervised methods often exhibit noise problems and fail to capture the correct labels [9]. For example, S2AG classifies articles into ca. 25 FoR in a general, flat list and bases its method on classifying the publication venue, rather than the article itself [20]. The Microsoft Academic

doi: 10.13053/CyS-28-4-5301

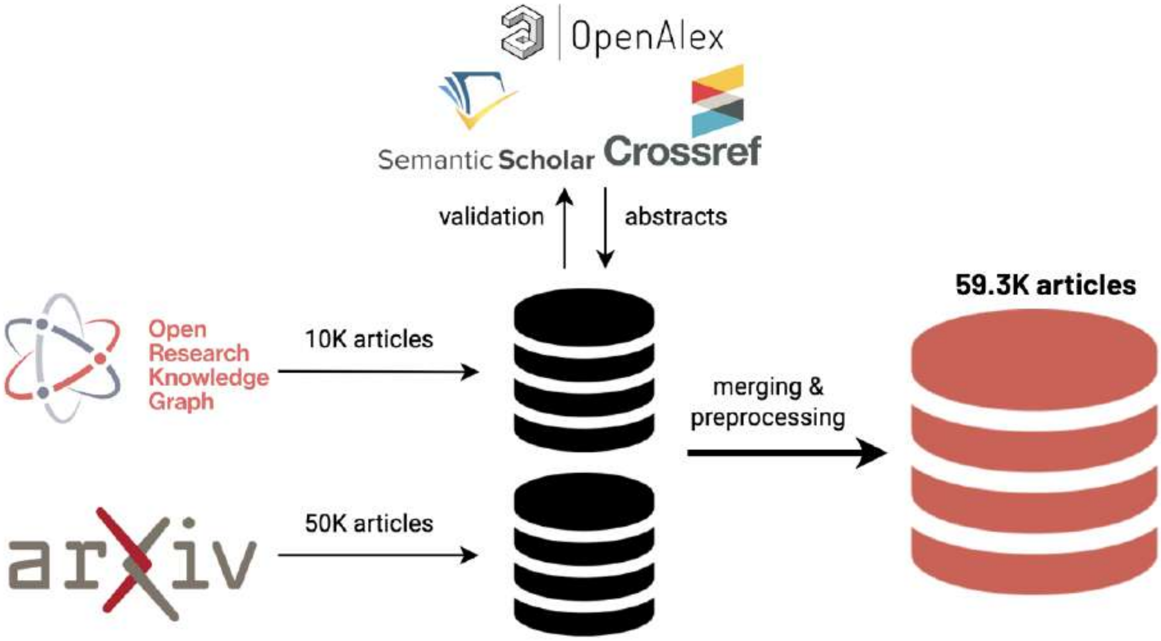

Fig. 1. Dataset construction pipeline

Graph [37], now discontinued, based its system on semantic clustering of scholarly papers and Wikipedia to get human-readable labels without any human intervention, and the ORKG does not have a FoR classification system at the time of writing. Recent research has been exploring methods of knowledge injection by using Semantic Web resources to augment the representation of text in machine learning (ML) and deep learning (DL) [27, 15, 23, 16, 5].

These methods make use of knowledge resources, e. g., knowledge graphs (KGs) like DBpedia [2], that describe millions of entities. Such rich symbolic resources can be transformed into a uniform low-dimensional vector space using advances in KG embedding (KGE) methods.

These embeddings represent KG entities along with their relations and can be directly used in ML/DL pipelines. Recent FoR classification efforts have attempted to use such methods for representing classes, providing richer embeddings than simply using the FoR label that consists of one or two words without context [5, 16].

This paper explores FoR classification by focusing on the semantic enrichment of classes. First, an extensive dataset of 59,344 instances is constructed from two resources with human-labelled scholarly articles: the ORKG and the arXiv repository[1].

Each article in the dataset is labelled with one FoR from a subset of the ORKG taxonomy of research fields[2] consisting of 123 FoR covering four hierarchical levels. To semantically enrich FoR classes, taxonomy labels are automatically linked to scientific entities on DBpedia, which are embedded using RDF2Vec [26].

Several models with various settings of FoR class features are trained, aiming to observe whether injecting external knowledge from DBpedia into the pipeline yields better classification results. Additionally, we experiment with different configurations of publication features using titles, abstracts, authors, and publishers. Our work contributes in three key ways:

[1] arxiv.org

[2] orkg.org/fields

Table 1. Sample of three instances from the final dataset

Title	Author(s)	DOI	Label
Sea–air CO2 fluxes in the Indian Ocean between 1990 and 2009	[’V. Valsala’, ’M. Ramonet’, ’E. Dlugokencky’, ’I. D. Lima’, ’S. Doney’, ’P. K. Patra’,’N. Metzl’, ’R. M. Law’, ’A. Lenton’, ’V. V. S. S. Sarma’]	bg.copernicus.org/ articles/10/7035/2013/	Oceanography
Modeling the Energy Evaluation for an Electric Machine	[’Valerian Croitorescu’]	link.springer.com/chapter/ 10.1007/978-3-319-45447-4_33	Mechanical Engineering

1. The introduced dataset can serve as a benchmark for future research employing different classification methods, and has been used for training and testing a shared task on FoR classification [1].
2. We present a method for automatically linking FoR to DBpedia entities using existing resources.
3. We expand research on knowledge injection methods for FoR classification by leveraging a large dataset, a more comprehensive set of FoR labels, and experimenting with additional metadata combinations for classification.

The rest of the paper is structured as follows. Section 2 explores related work in this domain. Section 3 delves into the taxonomy and specifies how the dataset was constructed and validated. Section 4 explains the methodology for representing publications and FoR classes.

In Section 5, we introduce the model architecture and present our results, which are then discussed in Section 6. Afterwards, Section 7 details the limitations of our research, and Section 8 concludes the article.

2 Related Work

Previous efforts have tackled the problem of classifying scholarly publications using different techniques. With the continuous expansion of research and the dynamic emergence of new FoR, there is a growing need for scalable systems capable of handling the increasing number of daily publications and classification labels. Therefore, some researchers argue that unsupervised classification systems that do not require manually curated and expensive training data are ideal in this scenario [30, 32, 33].

However, relying on such error-prone methods is not enough and typically requires manual validation [9]. Thus, we follow other research efforts that work with existing datasets of research publications labelled with FoR based on existing taxonomies, preferring a supervised learning approach that trains a model on more accurate data [40, 39, 10]. Supervised learning approaches can be performed using different methods and model architectures.

Research that mainly focuses on the representation of publication (meta-)data (i. e., without focusing on enriching FoR class representations) varies from using convolutional neural networks [28, 10] to deep neural networks [18]. Alternatively, other research explores representing FoR classes along with publication (meta-)data representations by jointly learning both in the same latent space [40, 6, 39] or by maximising mutual text-class information [11].

Additionally, some research has been done to explore the impact of incorporating external knowledge sources into class representations. [16] link 21 FoR classes to external KG entities on DBpedia, employing pre-trained KGEs for representation. They use computer science publications from arXiv, creating representations by Word2Vec embeddings of abstracts.

Their binary classification model achieves a micro F-measure of 30.7%. [5] compare knowledge injection approaches on a dataset constructed using AIDA KG.[3] The dataset is balanced and includes 12K articles for three FoR labels: artificial intelligence, software engineering, and human-computer interaction. Different models are trained, each employing a distinct knowledge injection approach.

[3] w3id.org/aida

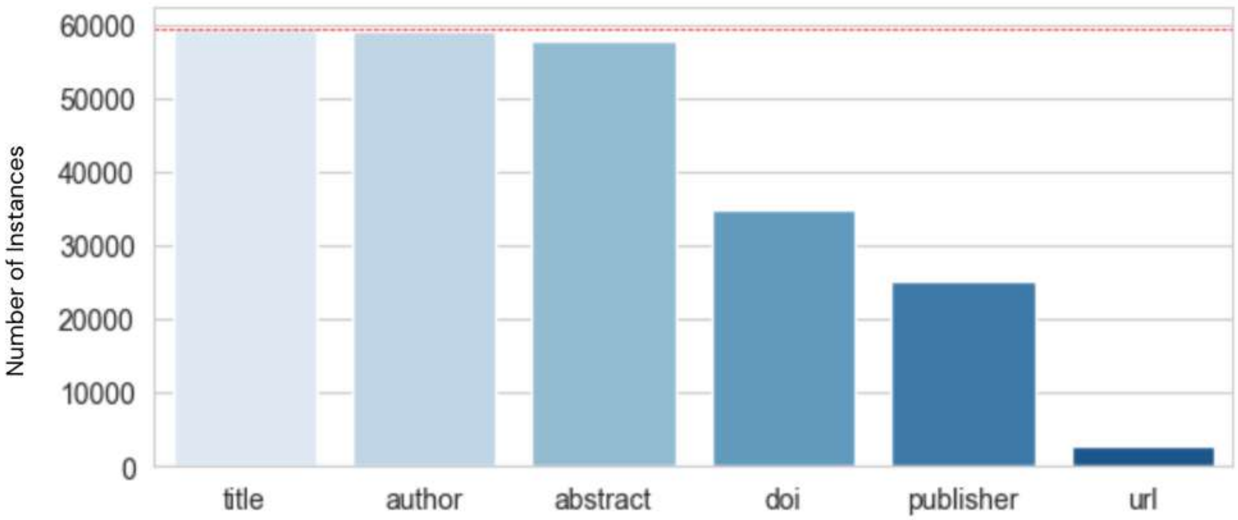

Fig. 2. Available metadata in the final dataset; the red line denotes the number of scholarly articles ($n = 59,344$)

They explore direct text injection, appending text from the computer science ontology [30] to input text, as well as models like K-BERT [22] and BERT followed by a multilayer perceptron (MLP). The best-performing method with a large training size was BERT-MLP, which concatenates BERT embeddings of articles with KGEs of labels, achieving an F1 score of 88%.

However, the aforementioned research on knowledge injection approaches applied to FoR classification has limitations. First, the manual linking of FoR to KGs like DBpedia is not scalable given the expanding nature of scientific FoR.

Additionally, both studies use a limited set of FoR labels within specific domains, hindering a comprehensive representation of research diversity and limiting conclusions on knowledge injection scalability and applicability to larger scientific repositories.

Moreover, in terms of textual embeddings, [16] rely on fixed embeddings (Word2Vec), while [5] use general-purpose BERT embeddings. However, studies on scientific text processing suggest that contextualised embeddings from science-specific language models, like SciBERT [3], often yield superior results [3, 24, 8].

3 Taxonomy and Dataset

3.1 Taxonomy

The existing taxonomy of research fields provided by the ORKG[4] is used in this study. At the time of writing, this taxonomy consists of more than 700 hierarchical FoR with up to five levels. The ORKG research fields taxonomy describes general scientific fields and was created based on:

1. The national academies of science, engineering, and medicine[5].
2. The German research foundation [6].
3. The arXiv category taxonomy [7].

We decided to use the ORKG research fields taxonomy because each paper is uploaded manually to the platform, and the uploader chooses the relevant FoR by selecting one from the available labels or suggesting a new one. Additionally, the taxonomy is large, hierarchical, and has a high level of granularity.

[4] orkg.org/fields shows the complete taxonomy.
[5] www.nationalacademies.org/home
[6] www.dfg.de/en/dfg_profile/statutory_bodies/review_boards/subject_areas/index.jsp
[7] arxiv.org/category_taxonomy

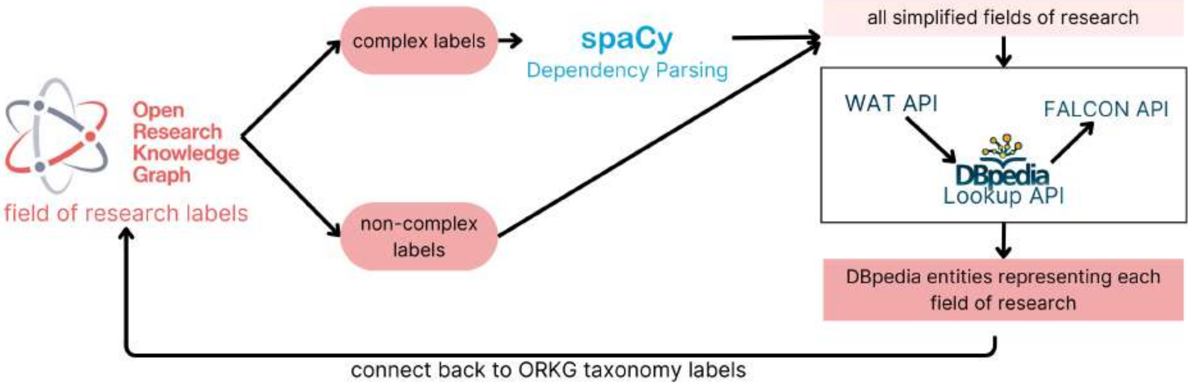
Open Research Knowledge Graph
field of research labels
complex labels
spaCy
Dependency Parsing
non-complex labels
all simplified fields of research
WAT API
FALCON API
DBpedia
Lookup API
DBpedia entities representing each field of research
connect back to ORKG taxonomy labels

Fig. 3. Entity linking pipeline

However, the taxonomy subset used in this study does not encompass all labels in the ORKG. When retrieving data, approximately 320 unique FoR were assigned to papers, but around 50% of labels had only a single associated paper. Those labels were merged with their parent nodes, and labels without assigned papers but with papers linked to their child nodes were merged with them. In total, the final taxonomy comprises 123 labels organised in up to four hierarchical levels[8].

3.2 Dataset Construction

The dataset used for training and testing is constructed from several open-source repositories. First, data from the ORKG was gathered by accessing their rdfDump and extracting papers that link to a specific FoR, which resulted in (meta-)data of ca. 10,000 papers. However, since the ORKG does not store abstracts for scholarly papers, three services connecting to general data repositories were used:

1. Crossref API [14], available via a CC BY 4.0 license.
2. S2AG API,[9] available via an ODC-BY-1.0 license.
3. OpenAlex [25], available via a CC0 license.

To augment the dataset, scholarly publications from arXiv[10] were included, which was chosen due to its manually curated FoR labels by volunteer moderators and its open accessibility under a CC0 1.0 license. Various preprocessing steps were applied to merge publications from both resources, detailed below. Figure 1 illustrates the overall pipeline[11]. Importantly, FoR labels were intentionally sourced exclusively from the ORKG and arXiv, as these repositories manually upload papers and curate FoR from their respective taxonomies. Unlike other repositories, they do not rely on automatic classification systems for labelling scholarly papers, aligning with our objective of avoiding duplication of previous classifiers in this work.

3.2.1 The ORKG Dataset

The ORKG (meta-)data was fetched by extracting all triples that include the property research field with the uniform resource identifier (URI) orkg.org/property/P30. For each resulting paper ID, a call to the API was performed to gather the following metadata fields: DOI, author, publication month, publication year, title, publisher, and URL. In order to fetch abstracts, we queried the DOI of each publication using Crossref API, S2AG, and PyAlex.[12]

[8] huggingface.co/spaces/rabuahmad/forcI-taxonomy/blob/main/taxonomy.json

[9] www.semanticscholar.org/product/api

[10] arxiv.org

[11] The dataset construction code is available at github.com/DFKI-NLP/nfdi4ds-forc

[12] pypi.org/project/pyalex

```
"Cosmology, Relativity, and Gravity": {
"http://dbpedia.org/resource/Cosmology":1,
"http://dbpedia.org/resource/General_relativity":1,
"http://dbpedia.org/resource/Theory_of_relativity":2,
"http://dbpedia.org/resource/Special_relativity":1,
"http://dbpedia.org/resource/Gravity":1
}
```

Fig. 4. Sample of an ORKG taxonomy label linked to DBpedia entities

This resulted in completing the abstracts of 83% of the publications retrieved from the ORKG (7,719 publications). In addition, the Crossref API and S2AG were used to validate metadata information obtained from the ORKG. More specifically, we validated and updated the publication year because a default date of January 2000 was used for many publications.

Several cleaning steps were performed to reduce the amount of noise in the data as much as possible. Publications with title lengths of less than 20 characters were removed, as were publications with null values for all of the following metadata fields: title, URL, DOI, abstract, and author.

Further cleaning steps included removing extra spaces and code snippets, standardising DOI formats to not include the prefix doi.org/, and standardising formats of author names. Finally, a step of deduplication according to titles and DOIs was performed. If two publications with the same title and DOI were found, the one with fewer null values in its metadata was kept.

The ORKG taxonomy includes the broad label Science, which lacks specificity for accurate publication classification. To address this, ca. 1,300 publications in the dataset originally labelled as Science were reassigned labels based on metadata from the Crossref API and S2AG.

FoR provided by these resources were employed by using manual mappings of the Crossref API and S2AG fields to the ORKG taxonomy.[13] This process replaced the labels for 830 publications. For the remaining publications, manual annotation by the first author was conducted to assign appropriate labels.

3.2.2 The arXiv Dataset

In order to obtain publications along with their metadata from arXiv, a snapshot was downloaded in November 2022 that contains ca. 2,000,000 publications with the following metadata fields: arXiv ID, submitter, authors, title, comments, journal-ref, DOI, abstract, versions, and categories.

Publications in arXiv that already exist in the ORKG dataset were removed based on their DOI. Since arXiv publications are labelled according to the arXiv Category Taxonomy,[14] they were mapped to the ORKG taxonomy in order to match the ORKG dataset. This was done by extracting all unique labels present in the arXiv dataset and manually mapping them to a corresponding label in the ORKG research fields taxonomy.[15]

Although the arXiv dataset contained more than 2,000,000 publications along with their metadata, 50,000 publications were sampled in order to keep a logical proportion with the ORKG dataset.

Only publications with a single FoR label were considered since this study deals with single-label classification according to the ORKG taxonomy and dataset. The distribution of labels was calculated, and random publications were sampled from the dataset in a manner that kept the original distribution of each label.

3.3 The Final Dataset

The two datasets were concatenated, and as a final preprocessing step, publications with non-English titles and abstracts were dropped by utilising the fastText language identification model.[16] The workflow explained above resulted in a dataset consisting of 59,344 instances of scholarly articles with their metadata. 9,331 of which originate from the ORKG, and the remaining 49,929 from the arXiv repository. The available metadata fields are title, abstracts, author, DOI, URL, publication month, publication year, and publisher. Table 1 displays a sample of two data instances with partial metadata fields, and Figure 2

[13] github.com/DFKI-NLP/nfdi4ds-forc/tree/main/data_processing/data/mappings

[14] arxiv.org/category_taxonomy

[15] github.com/DFKI-NLP/nfdi4ds-forc/blob/main/data_processing/data/mappings/arxiv_to_orkg_fields.json

[16] fasttext.cc/docs/en/language-identification.html

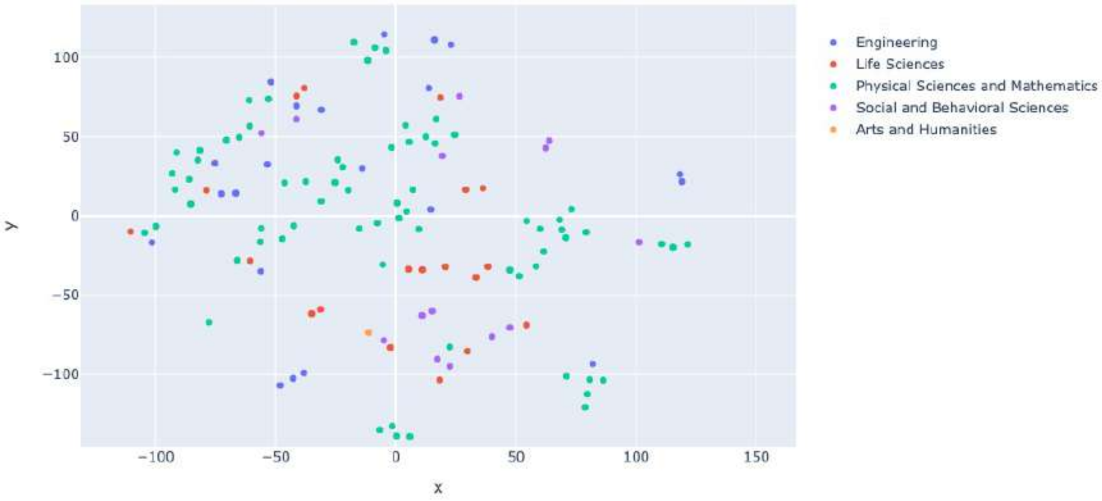

Fig. 5. t-SNE representation of the 123 ORKG taxonomy labels' embeddings; different colours denote the high level parent in the taxonomy

depicts the availability of each (meta-)data field[17]. Upon inspecting the distribution of FoR in the dataset, it becomes evident that the data is heavily imbalanced. Since most of the articles were taken from arXiv, the high-level label "Physical Sciences and Mathematics" possesses the most amount of articles. The three labels, "Physics", "Quantum Physics", and "Astrophysics and Astronomy", are the most frequent, with 6,610, 5,209, and 3,716 scholarly articles, respectively. On the other end of the spectrum, the label "Molecular, cellular, and tissue engineering" is the least frequent, with eight scholarly articles. The average number of scholarly articles per field is 482.5, and the median is 175.

4 Metadata and Class Representations

4.1 Metadata Features of Publications

In the constructed dataset, every paper instance has a title, and most (ca. 97%) have an abstract, which makes them the primary representation of each paper.

We embed them using SciNCL [24], a language model (LM) developed for training representations of scientific documents, outperforming similar models on several scientific text processing tasks. We take the representation of the last hidden state as the contextual embedding for each input. Author names in the dataset exhibit noise and inconsistency due to manual curation from the ORKG and arXiv.

Variations include random use of initials or omitting first names, stating only the last name of the first author followed by "et al.", and diverse affiliation representations. To ensure consistent preprocessing, we employ the Python nameparser module,[18] designed for parsing human names into structured components.

While disambiguating authors with identical names remains challenging, we focus on last names and the first letters of first names, assuming identical names refer to the same person. This approach results in 121,507 distinct authors. To embed them, we link authors to the papers they wrote, representing them as the average of all

[17] The full dataset is available at zenodo.org/records/10777735

[18] pypi.org/project/nameparser

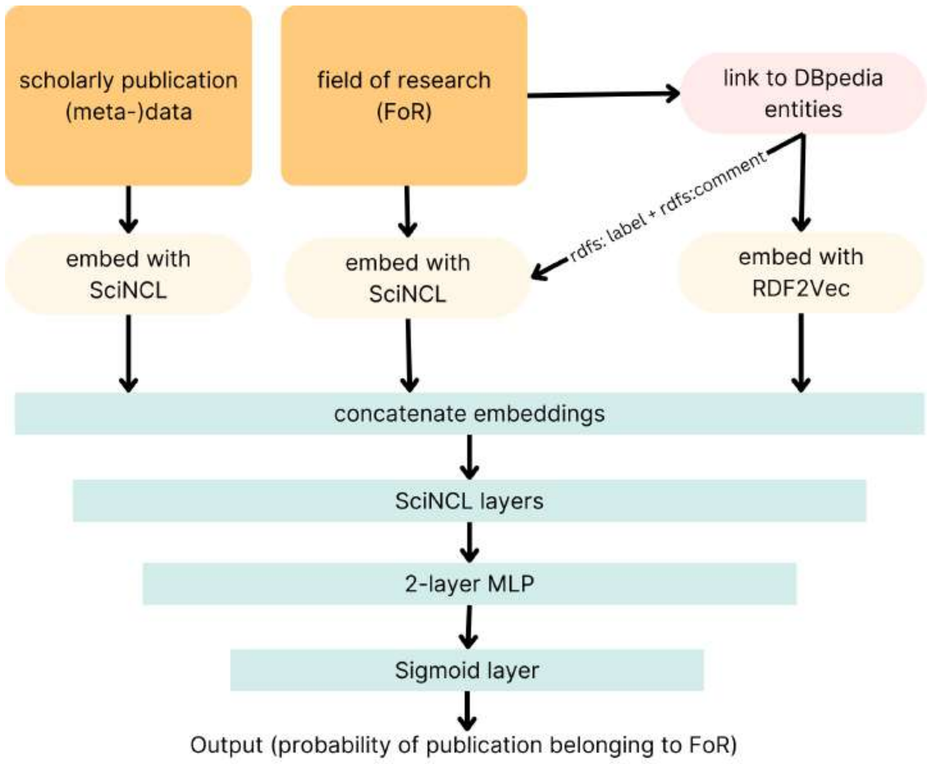

Fig. 6. Model architecture of the proposed approach

paper embedding (i. e., contextual embeddings of titles and abstracts using SciNCL). The author representation for each publication is the average of its listed authors. Publications without affiliated authors are represented by a vector of zeros. The publisher in the constructed dataset denotes the journal, proceedings of a conference, or website that published the associated scholarly paper. Similar to authors, the documentation of publishers in the ORKG and arXiv is inconsistent.

For example, some papers include the full name of the publication (e. g., “Journal of Marine Systems”), while others include abbreviations and a specific publication year and/or volume or pages numbers (e. g., “Mech. Res. Commun. 47 (2013), 69-76”). We apply a similar approach to publishers as to authors. We preprocess the texts by lower-casing and removing digits, punctuation, and white spaces, which results in 7,827 unique publishers.

We link each publisher to all of its affiliated papers and represent it as the average of its paper embeddings, using contextual text embeddings from SciNCL. Since each scholarly paper only has one publisher, it is represented as its created embedding. As with authors, a paper with no affiliated publisher is represented by a vector of zeros.

4.2 Semantic Representation of Classes

Manually linking taxonomy labels to equivalent DBpedia entities is a laborious and time-consuming effort, especially if the taxonomy consists of a large number of labels. Because of this, we implement an automatic linking pipeline that relies on previous entity linking work of short natural text to their corresponding DBpedia entities. Figure 3 displays this pipeline, each step of which will be addressed in detail below.

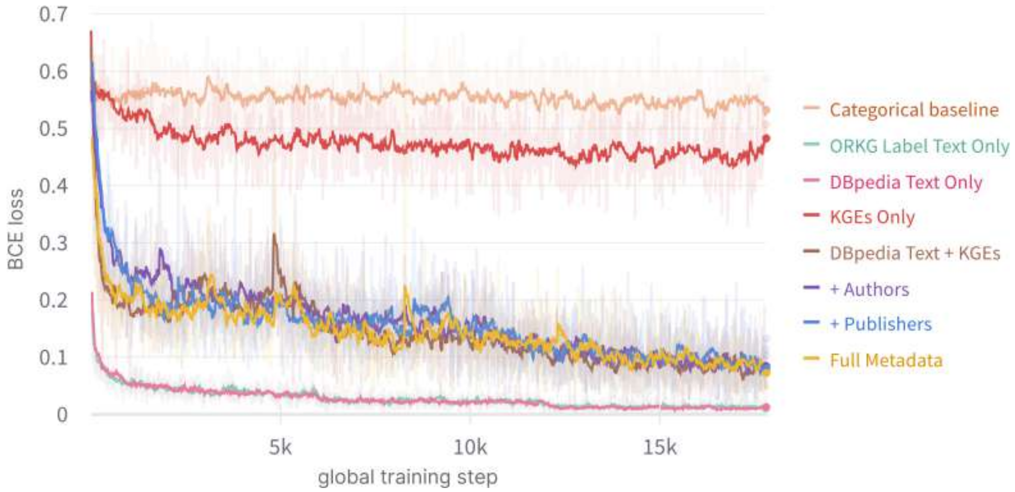

Fig. 7. BCE loss per global training step combined (smoothing of 80% is applied for easier comparison)

Since the labels in the ORKG are inconsistent and can describe multiple FoR (e. g., “Molecular, cellular, and tissue engineering”), we divide them into two categories, complex and non-complex labels. Complex labels are classified by checking if a comma or the token “and” exists in the text. This division results in 51 complex and 72 non-complex labels. Since complex labels have to be linked to all of their relevant DBpedia entities (e. g., “Social and Behavioral Sciences” has to be linked to both “Social Sciences” and “Behavioral Sciences”), we parse them using the dependency parsing module offered by the spaCy library[19].

We iterate over all complex FoR labels and extract their dependency relations, creating new non-complex labels by taking different relations into account (e. g., compound, noun/adjective modifier). This method correctly parses 43 out of the 51 complex labels (84%)[20]. DBpedia is one of the largest KGs containing more than 850 million triples.[21] Because of this, it is sensible to assume that linking the FoR labels to all DBpedia entities will be computationally expensive and result in many false positives.

This is especially problematic for polysemous labels such as “Tissues”, which could be mistakenly linked with the entity http://dbpedia.org/resource/Tissue_paper rather than dbpedia.org/page/Tissue_(biology). In an effort to avoid this, we extract a list of all DBpedia entities that appear as objects to the predicate dbpedia.org/ontology/academicDiscipline.

This list is extracted using a SPARQL query that is executed on the DBpedia SPARQL endpoint. In addition to extracting the entity URLs, we add each entity’s English label (www.w3.org/2000/01/rdf-schema#label) and comment (www.w3.org/2000/01/rdf-schema#comment). At the time of writing, this query results in 5060 entities.

To start linking labels to DBpedia entities from the list of academic disciplines, we gather the results of parsed ORKG labels with the non-complex ORKG labels, encompassing all unique FoR labels. We use existing APIs for linking short text to DBpedia entities, starting by querying WAT API[22] with each one of the labels.

If an exact match exists for a FoR label (i. e., the label of the linked DBpedia entity is an exact match of the FoR text after lower-casing and stemming), and the entity exists in the list of dbo:academicDisciplines, it is used as the linked DBpedia resource of that FoR.

[19] spacy.io/usage/linguistic-features\#dependency-parse

[20] github.com/DFKI-NLP/for-classifier/blob/main/results/dep-parsing.json (full results)

[21] The number is taken from the 2021-06 snapshot release announcement www.dbpedia.org/blog/snapshot-2021-06-release/

[22] sobigdata.d4science.org/web/tagme/wat-api

Table 2. Experimental results

Publication Features	Class Features	Precision	Recall	F1	Accuracy
Baseline					
Titles + Abstracts	Categorical Encoder	0.0	0.0	0.0	74.85
Embedding Class Labels with SciNCL					
Titles + Abstracts	ORKG Labels Text	93.54	93.80	93.67	96.83
Injecting DBpedia Class Features					
Titles + Abstracts	DBpedia Text	**93.55**	**94.11**	**93.83**	**96.91**
Titles + Abstracts	KGEs	75.83	29.39	42.36	80.00
Titles + Abstracts	DBpedia Text + KGEs	93.18	93.19	93.18	96.60
Adding Publication Metadata					
Titles + Abstracts Authors	DBpedia Text + KGEs	93.20	92.02	92.61	96.32
Titles + Abstracts Publishers	DBpedia Text + KGEs	92.25	93.52	92.88	96.43
Titles + Abstracts Authors Publishers	DBpedia Text + KGEs	93.28	92.51	92.90	96.43

The same is done for results from DBpedia Lookup[23] and FALCON API [29]. The exact order of APIs was chosen after testing all three by checking how many exact matches they have with the FoR labels. WAT API resulted in ca. 68%, DBpedia Lookup in ca. 65%, and FALCON API in 53%. It is important to note that WAT API only returns the single top result for a query, while DBpedia Lookup and FALCON API were both set to return the top five results.

For the remaining labels that do not have an exact match, we gather all three API results and only keep the entities that exist in the list of dbo:academicDisciplines. We link the FoR label with all those entities while taking their frequency into account. For example, if a DBpedia entity showed up in the results of all three APIs, it was given a weight of 3, denoting its frequency in the results.

The idea is to use the weights in order to represent the FoR label with the weighted average of all its linked DBpedia entities. Any remaining FoR were linked based on a fuzzy matching algorithm between the FoR label text and the labels of the entities in the dbo:academicDisciplines list.

FoR labels that were linked either by exact or fuzzy matches get a weight of 1. After linking all FoR entities, complex labels were connected to the results of all their parsed labels. A sample of the final linking results is depicted in Figure 4[24].

In order to obtain KGEs of the DBpedia entities that were linked to the taxonomy, we use publicly available RDF2Vec pre-trained embeddings [7].

We link the embeddings of each parsed label back to the original ORKG taxonomy labels. To do that, we use the weights gathered in the linking process to represent each ORKG label as the weighted average of its linked entities' embeddings.

Equation 1 denotes how the representative embedding of each label from the ORKG taxonomy is calculated, where n is the overall number of linked DBpedia entities, i is individual linked entities, e is the embedding of each linked entity obtained from the pre-trained RDF2Vec dataset, and w is its weight. Figure 5 uses t-SNE [36] to display the final representative embeddings of the 123 class labels from the ORKG taxonomy:

[23] github.com/dbpedia/lookup

[24] Full results are available at github.com/DFKI-NLP/for-classifier/blob/main/results/orkg-taxonomy-linking.json

$$\text{ORKG label} = \frac{\sum_{i=1}^{n} w_i e_i}{\sum_{i=1}^{n} w_i}. \qquad (1)$$

Additionally, we use textual representations of each DBpedia entity by extracting the English objects of the label and comment properties. To represent each ORKG taxonomy label, we construct a string of all the labels and comments that represent all DBpedia entities linked to a certain ORKG class. Separate elements are separated by the [SEP] token since the objective is to embed the full textual representation of each ORKG label using SciNCL[25].

5 Implementation and Experimental Results

We extend previous work on knowledge injection applied to FoR classification [16, 5] by employing the rule-based process outlined earlier for linking taxonomy labels to DBpedia. The resulting entity embeddings are then concatenated to augment the semantic representation of classes.

Additionally, we embed the linked DBpedia entity label(s) and comment(s), extracted from rdfs:label and rdfs:comment respectively, using SciNCL. We include representations of titles, authors, and publishers as features for each scholarly article, exploring different combinations for optimal classification results.

The concatenated layer is then processed through SciNCL layers, two MLP layers, and a final Sigmoid layer that outputs the probability of the scholarly article belonging to a specific class. The architecture of the proposed knowledge injection model is illustrated in Figure 6.

To run the classifier, the constructed dataset is processed to suit a binary classification task by iterating over its rows and creating one positive sample and three negative ones. Positive samples consist of the scholarly publication with the correct FoR class attached to it, while negative samples attach a random incorrect FoR class from the taxonomy. This dataset consists of 237,376 data points with 25% positive samples (i. e., 1) and 75% negative samples (i. e., 0). Each instance in the binary dataset has the following format:

(title and abstract, author(s) embedding, publisher embedding, ORKG class textual representation, DBpedia textual representation, DBpedia KGE, binary label)

We present the outcomes of various experiments involving different metadata configurations. All model runs share identical hyperparameters and utilise the NVIDIA RTX A6000 GPU. The data is shuffled with a random seed of 42, split into 80/20 for training/testing, and trained for three epochs with a batch size of 32.

Default hyperparameters from the HuggingFace Trainer class[26] are applied for all other settings.[27] Figure 7 illustrates the cumulative training loss using binary cross-entropy (BCE) over global training steps, and Table 2 consolidates the evaluation results for each model, presenting precision, recall, F1, and accuracy scores.

Categorical Baseline As a baseline for the following experiments, we run a model that receives the title and abstract of each publication as input and encodes the classes categorically using LabelEncoder[28] without any method of semantic representation for the class labels.

ORKG Labels Representation We run a pairwise text classification model that only uses the title and abstract of each paper paired with the ORKG taxonomy label it is tagged as. We do not include any external knowledge from DBpedia in textual format or entity embeddings. Texts are tokenised and embedded using SciNCL through the HuggingFace Transformers library, specifically the BertForSequenceClassification class,[29] with the number of labels assigned as 1.

[25] The full results are available in JSON format at github.com/DFKI-NLP/for-classifier/blob/main/results/orkg-taxonomy-text.json

[26] huggingface.co/docs/transformers/v4.34.1/en/main_classes/trainer

[27] The code for all models is accessible at github.com/DFKI-NLP/for-classifier/tree/main/models

[28] tinyurl.com/nw2hh356

[29] huggingface.co/docs/transformers/model_doc/bert\#transformers.BertForSequenceClassification

DBpedia Textual Representation We run a model of textual pairwise binary classification using the text representation of each publication by fine-tuning SciNCL on the textual representations extracted from DBpedia entities.

Since BERT tokenisers use a maximum length of 512 tokens per text, we use the extracted label and comment for each linked entity, as they tend to be used for short human-readable descriptions of the entity. This setup is identical to the one described for ORKG labels, with the difference being using DBpedia text extracted from the entities instead of the ORKG label itself.

DBpedia KGEs We run the same experiment described above without using the textual data extracted from DBpedia entities' labels and comments. Instead, we only use titles and abstracts of each scholarly publication and the KGEs of their associated DBpedia entities from pre-trained RDF2Vec embeddings.

DBpedia KGEs and Texts We add the representations of KGEs to the model by implementing the architecture depicted previously in Figure 6. To do that, we define a class which builds a layer of the pre-trained SciNCL model using the HuggingFace Transformers AutoModel class.[30] In the first experiment, we only use title and abstract metadata for each publication.

Adding Authors and Publishers We experiment further by running the following combinations of additional metadata: 1. Adding author representations; 2. Adding publisher representations; and 3. Adding both author and publisher representations.

6 Discussion

The categorical baseline results (Figure 7) reveal that the binary classifier fails to learn the effective classification of titles and abstracts into FoR using numerical categories alone. The BCE training loss consistently hovers around 0.6, indicating a lack of learning. Evaluation metrics, including precision, recall, and F1 scores, are all 0.0, suggesting an inability to accurately predict positive instances, with the accuracy aligning with the distribution of positive and negative classes (Table 2). Replacing categorical encoding with KGEs results in a modest improvement (Figure 7). The training loss decreases initially and plateaus around 0.47, showcasing a positive trend. However, KGEs do not offer the optimal semantic representation compared to other models.

This limitation is most likely attributed to the use of pre-trained DBpedia embeddings, introducing knowledge noise due to an abundance of unfiltered triples, which results in a diversion of entity embeddings from their correct meanings [41, 22, 31]. Large-scale KGs, lacking sufficient human supervision and error detection mechanisms, have been shown to amplify this knowledge noise issue [21].

However, the noisy KGEs revealed an intriguing observation in the ORKG taxonomy of research fields. The t-SNE representation of the taxonomy (Figure 5) reveals a sparse and inconsistent distribution across the embedding space that deviates from the hierarchical structure in the ORKG. Ideally, KGE representations of classes should form clusters aligning with their high-level taxonomy labels.

The scattered arrangement highlights the taxonomy's complexity and inconsistency, stemming from the interdisciplinary nature of FoR. Notably, instances like “Computational Linguistics” positioned as a child node of “Linguistics” under “Social and Behavioral Sciences” without a connection to “Computer Sciences” exemplify this interdisciplinary challenge. Additionally, the automatic linking method to DBpedia produced some false positives, which could have also contributed to this representation.

The results clearly demonstrate that the most effective method for representing semantic information of classes is to embed their representative text using a large LM (LLM). Surprisingly, the length of the text does not significantly impact the performance, whether it is a short label from the ORKG or a longer description from DBpedia entities. This is evident in the aligned training loss trajectories of both model variants in Figure 7 and the marginal difference in evaluation scores shown in Table 2.

[30] huggingface.co/docs/transformers/model_doc/auto

The slight superiority of DBpedia textual representations is likely inconsequential for the given task. The similar outcomes of both models can be attributed to the use of SciNCL, a model specifically trained on scientific text, which effectively embeds short and longer texts alike. However, further experimentation with a general LLM, such as BERT, would be beneficial to confirm this hypothesis.

Combining DBpedia textual models with their KGEs results in a slightly inferior performance due to the discussed knowledge noise problem. This setting is employed to compare various metadata combinations for publications. The four models utilising DBpedia text and KGEs exhibit similar training loss trajectories (Figure 7). The best final loss score (0.145) is shared by two models:

1. Using only titles and abstracts.
2. Using full metadata.

When comparing author and publisher embeddings, adding publishers shows a marginal advantage with a final loss score of 0.15 compared to 0.16 when only adding author embeddings. However, this difference may be attributed to noise and does not necessarily indicate the superiority of publisher embeddings.

In terms of evaluation scores, the most effective model appears to be the one solely relying on titles and abstracts to represent publications. These results align with the embedding methods for authors and publishers. The author embedding method, based on the assumption that last name + first letter of first name uniquely identifies individuals, could benefit from experimenting with author disambiguation approaches [19, 34].

Notably, the Open Researcher and Contributor ID[31], designed to provide unique identifiers linked to researchers' publications and affiliations, presents a promising avenue for improving disambiguation. Similar challenges are observed in the publisher embedding method, where ensuring the uniqueness of extracted entities and preventing duplicates remains uncertain.

7 Limitations

The approach detailed in the previous sections demonstrates promising results, yet it has some constraints. First, our reliance on manually uploaded resources during dataset construction introduces limitations. This is because label annotation is conducted by a single individual without assurance of authorship alignment with the uploaded publication, potentially introducing biases in label selection.

The absence of multiple annotators with no inter-annotator agreement score to validate the labels further compounds this issue. Moreover, the language restriction to English, the predominant research publication language [13], limits both our dataset and the developed model's scope, as scientific LLMs like SciNCL predominantly train on English datasets.

Additionally, while DBpedia serves as a vast and comprehensive KG, it has been noted for incompleteness, inaccuracies, and biases, due to its automatic extraction process from Wikipedia [35]. Given our reliance on this resource, such deficiencies may directly impact the entity linking process. Lastly, though our approach emphasises scalability, direct practical implementation into databases and generalisation to multidisciplinary publications remain untested.

Further research addressing these aspects, alongside ethical considerations, is warranted. These include transparency, fairness, and biases of the classification model, particularly concerning FoR due to the dataset's class imbalance, as well as mechanisms for accountability encompassing user feedback.

8 Conclusions

The research presented in this article addresses the challenge of accurately classifying publications into FoR using a novel dataset of 59,344 English publications from open-source repositories. Our approach employs a taxonomy of 123 FoR labels across four hierarchical levels, automatically linking them to DBpedia entities and utilising pre-trained KGEs.

[31] orcid.org

The SciNCL-based model, featuring a two-layered MLP, effectively combines textual embeddings of publications with entity and textual embeddings of classes. Our evaluation highlights the effectiveness of utilising textual representations of classes, irrespective of length. Challenges associated with KGEs, particularly the knowledge noise problem, are also emphasised.

Future work could explore different configurations of walking and sampling strategies in RDF2Vec, alternative embedding methods like TransE [38], and knowledge injection methods such as K-BERT to address knowledge noise. Additionally, avenues for further research include exploring publication features like authors and publishers by investigating new methods for their preprocessing, linking, and disambiguation.

Acknowledgments

This publication was supported by the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[32] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.). The NFDI is funded by the Federal Republic of Germany and its states. The paper received funding through the German Research Foundation (DFG) project NFDI4DS (no. 460234259).

We would like to thank our colleagues at DFKI Berlin, Malte Ostendorff and Ekaterina Borisova, for their assistance during this research. We also thank Jennifer D'Souza from the ORKG team for her assistance in the initial stages of the data construction process.

Statements and Declarations

- Funding: This research was funded by the German Research Foundation (DFG) project NFDI4DS (no. 460234259).
- Conflict of interest: The authors declare that they have no conflict of interest.
- Ethics approval and consent to participate: All data is collected from open-source resources. The study does not include human or animal participants.
- Consent for publication: Not applicable.
- Data availability: zenodo.org/records/10777735.
- Materials availability: Not applicable.
- Code availability: github.com/DFKI-NLP/nfdi4ds-forc, github.com/DFKI-NLP/for-classifier.
- Author contribution: Raia Abu Ahmad: conceptualisation, data construction, code development, original draft. Georg Rehm: conceptualisation, supervision, manuscript revision and editing.

[32] www.nfdi4datascience.de

References

1. **Abu-Ahmad, R., Borisova, E., Rehm, G. (2024).** FoRC@NSLP2024: Overview and insights from the field of research classification shared task. Natural Scientific Language Processing and Research Knowledge Graphs, Lecture Notes in Computer Science, Springer Cham, pp. 189–204. DOI: 10.1007/978-3-031-65794-8_12.

2. **Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z. (2007).** DBpedia: A nucleus for a web of open data. Springer Berlin Heidelberg, pp. 722–735. DOI: 10.1007/978-3-540-76298-0_52.

3. **Beltagy, I., Lo, K., Cohan, A. (2019).** SciBERT: A pretrained language model for scientific text. Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, pp. 3615–3620. DOI: 10.18653/v1/D19-1371.

4. **Bornmann, L., Haunschild, R., Mutz, R. (2021).** Growth rates of modern science: A latent piecewise growth curve approach to model publication numbers from established and new literature databases. Humanities and Social Sciences Communications, Vol. 8,

No. 1, pp. 1–15. DOI: 10.1057/s41599-021-00903-w.

5. **Cadeddu, A., Chessa, A., de-Leo, V., Fenu, G., Motta, E., Osborne, F., Recupero, D. R., Salatino, A., Secchi, L. (2023).** Enhancing scholarly understanding: A comparison of knowledge injection strategies in large language models. CEUR Deep Learning for Knowledge Graphs Workshop Proceedings, pp. 1–6.

6. **Chen, H., Ma, Q., Lin, Z., Yan, J. (2021).** Hierarchy-aware label semantics matching network for hierarchical text classification. Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Vol. 1, pp. 4370–4379. DOI: 10.18653/v1/2021.acl-long.337.

7. **Christensen, M. P., Lissandrini, M., Hose, K. (2022).** DBpedia RDF2Vec graph embeddings. DOI: 10.5281/zenodo.6384728.

8. **Cohan, A., Feldman, S., Beltagy, I., Downey, D., Weld, D. (2020).** SPECTER: Document-level representation learning using citation-informed transformers. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, pp. 2270–2282. DOI: 10.18653/v1/2020.acl-main.207.

9. **Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A. (2017).** Supervised learning of universal sentence representations from natural language inference data. Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, pp. 670–680. DOI: 10.18653/v1/D17-1070.

10. **Daradkeh, M., Abualigah, L., Atalla, S., Mansoor, W. (2022).** Scientometric analysis and classification of research using convolutional neural networks: A case study in data science and analytics. Electronics, Vol. 11, No. 13, pp. 2066. DOI: 10.3390/electronics11132066.

11. **Deng, Z., Peng, H., He, D., Li, J., Yu, P. (2021).** HTCInfoMax: A global model for hierarchical text classification via information maximization. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 3259–3265. DOI: 10.18653/v1/2021.naacl-main.260.

12. **Fortunato, S., Bergstrom, C. T., Börner, K., Evans, J. A., Helbing, D., Milojević, S., Petersen, A. M., Radicchi, F., Sinatra, R., Uzzi, B., Vespignani, A., Waltman, L., Wang, D., Barabási, A. L. (2018).** Science of science. Science, Vol. 359, No. 6379. DOI: 10.1126/science.aao0185.

13. **Hamel, R. E. (2007).** The dominance of English in the international scientific periodical literature and the future of language use in science. AILA Review, Vol. 20, pp. 53–71. DOI: 10.1075/aila.20.06ham.

14. **Hendricks, G., Tkaczyk, D., Lin, J., Feeney, P. (2020).** Crossref: The sustainable source of community-owned scholarly metadata. Quantitative Science Studies, Vol. 1, No. 1, pp. 414–427. DOI: 10.1162/qss_a_00022.

15. **Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A. C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A. (2021).** Knowledge graphs. ACM Computing Surveys, Vol. 54, No. 4, pp. 1–37. DOI: 10.1145/3447772.

16. **Hoppe, F., Dessì, D., Sack, H. (2021).** Deep learning meets knowledge graphs for scholarly data classification. Companion Proceedings of the Web Conference, pp. 417–421. DOI: 10.1145/3442442.3451361.

17. **Jaradeh, M. Y., Oelen, A., Farfar, K. E., Prinz, M., D'Souza, J., Kismihók, G., Stocker, M., Auer, S. (2019).** Open research knowledge graph: Next generation infrastructure for semantic

scholarly knowledge. Proceedings of the 10th International Conference on Knowledge Capture, pp. 243–246. DOI: 10.48550/ARXIV.1901.10816.

18. **Kandimalla, B., Rohatgi, S., Wu, J., Giles, C. L. (2021).** Large scale subject category classification of scholarly papers with deep attentive neural networks. Frontiers in Research Metrics and Analytics, Vol. 5, pp. 600382. DOI: 10.3389/frma.2020.600382.

19. **Kim, K., Sefid, A., Weinberg, B. A., Giles, C. L. (2018).** A web service for author name disambiguation in scholarly databases. Proceedings of the IEEE International Conference on Web Services, pp. 265–273. DOI: 10.1109/ICWS.2018.00041.

20. **Kinney, R., Anastasiades, C., Authur, R., Beltagy, I., Bragg, J., Buraczynski, A., Cachola, I., Candra, S., Chandrasekhar, Y., Cohan, A., Crawford, M., Downey, D., Dunkelberger, J., Etzioni, O., Evans, R., Feldman, S., Gorney, J., Graham, D., Hu, F., Huff, R., et al. (2023).** The semantic scholar open data platform. DOI: 10.48550/arXiv.2301.10140.

21. **Liang, J., Xiao, Y., Zhang, Y., Hwang, S. W., Wang, H. (2017).** Graph-based wrong IsA relation detection in a large-scale lexical taxonomy. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 31, No. 1.

22. **Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P. (2020).** K-BERT: Enabling language representation with knowledge graph. Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 3, pp. 2901–2908.

23. **Ostendorff, M., Bourgonje, P., Berger, M., Moreno-Schneider, J., Rehm, G., Gipp, B. (2019).** Enriching BERT with knowledge graph embeddings for document classification. Proceedings of the 15th Conference on Natural Language Processing, pp. 1–8. DOI: 10.48550/arXiv.1909.08402.

24. **Ostendorff, M., Rethmeier, N., Augenstein, I., Gipp, B., Rehm, G. (2022).** Neighborhood contrastive learning for scientific document representations with citation embeddings. Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 11670–11688. DOI: 10.18653/v1/2022.emnlp-main.802.

25. **Priem, J., Piwowar, H., Orr, R. (2022).** OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. DOI: 10.48550/ARXIV.2205.01833.

26. **Ristoski, P., Paulheim, H. (2016).** RDF2Vec: RDF graph embeddings for data mining. pp. 498–514. DOI: 10.1007/978-3-319-46523-4_30.

27. **Ristoski, P., Paulheim, H. (2016).** Semantic web in data mining and knowledge discovery: A comprehensive survey. Journal of Web Semantics, Vol. 36, pp. 1–22. DOI: 10.1016/j.websem.2016.01.001.

28. **Rivest, M., Vignola-Gagné, E., Archambault, E. (2021).** Article-level classification of scientific publications: A comparison of deep learning, direct citation and bibliographic coupling. PLOS ONE, Vol. 16, No. 5, pp. e0251493. DOI: 10.1371/journal.pone.0251493.

29. **Sakor, A., Mulang, I. O., Singh, K., Shekarpour, S., Vidal, M. E., Lehmann, J., Auer, S. (2019).** Old is gold: Linguistic driven approach for entity and relation linking of short text. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1, pp. 2336–2346.

30. **Salatino, A., Osborne, F., Motta, E. (2021).** CSO Classifier 3.0: A scalable unsupervised method for classifying documents in terms of research topics. International Journal on Digital Libraries, Vol. 23, No. 1, pp. 91–110. DOI: 10.1007/s00799-021-00305-y.

31. **Shan, Y., Bu, C., Liu, X., Ji, S., Li, L. (2018).** Confidence-aware negative sampling method for noisy knowledge graph embedding. IEEE

International Conference on Big Knowledge, pp. 33–40. DOI: 10.1109/ICBK.2018.00013.

32. **Shen, Z., Ma, H., Wang, K. (2018).** A web-scale system for scientific knowledge exploration. Proceedings of ACL, System Demonstrations, Association for Computational Linguistics, pp. 87–92. DOI: 10.18653/v1/P18-4015.

33. **Sood, S. K., Kumar, N., Saini, M. (2021).** Scientometric analysis of literature on distributed vehicular networks: Vosviewer visualization techniques. Artificial Intelligence Review, Vol. 54, No. 8, pp. 6309–6341. DOI: 10.1007/s10462-021-09980-4.

34. **Subramanian, S., King, D., Downey, D., Feldman, S. (2021).** S2AND: A benchmark and evaluation system for author name disambiguation. ACM/IEEE Joint Conference on Digital Libraries, IEEE, pp. 170–179. DOI: 10.1109/jcdl52503.2021.00029.

35. **Töpper, G., Knuth, M., Sack, H. (2012).** DBpedia ontology enrichment for inconsistency detection. Proceedings of the 8th International Conference on Semantic Systems, pp. 33–40. DOI: 10.1145/2362499.2362505.

36. **van-der-Maaten, L., Hinton, G. (2008).** Visualizing data using t-SNE. Journal of machine learning research, Vol. 9, No. 86, pp. 2579–2605.

37. **Wang, K., Shen, Z., Huang, C., Wu, C. H., Dong, Y., Kanakia, A. (2020).** Microsoft academic graph: When experts are not enough. Quantitative Science Studies, Vol. 1, No. 1, pp. 396–413. DOI: 10.1162/qss_a_00021.

38. **Wang, Q., Mao, Z., Wang, B., Guo, L. (2017).** Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering, Vol. 29, No. 12, pp. 2724–2743. DOI: 10.1109/TKDE.2017.2754499.

39. **Wang, Z., Wang, P., Huang, L., Sun, X., Wang, H. (2022).** Incorporating hierarchy into text encoder: A contrastive learning approach for hierarchical text classification. Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Vol. 1, pp. 7109–7119. DOI: 10.18653/v1/2022.acl-long.491.

40. **Zhang, Y., Shen, Z., Dong, Y., Wang, K., Han, J. (2021).** MATCH: Metadata-aware text classification in a large hierarchy. Proceedings of the Web Conference 2021, pp. 3246–3257. DOI: 10.1145/3442381.3449979.

41. **Zhao, Y., Feng, H., Gallinari, P. (2019).** Embedding learning with triple trustiness on noisy knowledge graph. Entropy, Vol. 21, No. 11, pp. 1083. DOI: 10.3390/e21111083.

Article received on 12/04/2024; accepted on 24/06/2024.
**Corresponding author is Raia Abu Ahmad.*