Chapter 16
Medical Knowledge Graphs in the Discovery of Future Research Collaborations

Nikolaos Giarelis, Nikos Kanakaris, and Nikos Karacapilidis

Abstract This chapter introduces a framework that is based on a novel graph-based text representation method and combines graph-based feature selection, text categorization and link prediction to advance the discovery of future research collaborations. Our approach integrates into a single knowledge graph both structured and unstructured textual data through a novel representation of multiple scientific documents. The Neo4j graph database is used for the representation of the proposed scientific knowledge graph. For the implementation of our approach, we use the Python programming language and the scikit-learn machine learning library. We assess our approach against classical link prediction algorithms using accuracy, recall and precision as our performance metrics. Our experiments achieve state-of-the-art accuracy in the task of predicting future research collaborations. The experimentations reported in this chapter use the COVID-19 Open Research Dataset.

Keywords Link prediction · Text categorization · Feature selection · Knowledge graphs · Natural language processing · Document representation

16.1 Introduction

In recent years, we have witnessed an increase in the adoption of graph-based approaches for predicting future research collaborations by utilizing tasks such as link prediction, feature selection and text categorization [1, 2]. In these approaches, graph-based text representations are being used as a means to select important features from

N. Giarelis · N. Kanakaris · N. Karacapilidis (✉)
Industrial Management and Information Systems Lab, MEAD, University of Patras, Rio, 26504 Patras, Greece
e-mail: karacap@upatras.gr

N. Giarelis
e-mail: giarelis@ceid.upatras.gr

N. Kanakaris
e-mail: nkanakaris@upnet.gr

C.-P. Lim et al. (eds.), *Handbook of Artificial Intelligence in Healthcare*,
Intelligent Systems Reference Library 212,
https://doi.org/10.1007/978-3-030-83620-7_16

all documents and build communities or clusters of similar documents, whereas a collaboration between two researchers is generally denoted by a scientific article written by them [3].

Graph-based approaches (particularly those concerning knowledge graphs) build on concepts and methods from graph theory (e.g. node centrality, link prediction and node similarity measures) to discover hidden knowledge from the structural characteristics of the corresponding research graph [4]. However, despite their broad adoption, existing graph-based approaches aiming to discover future research collaborations utilize only the structural characteristics of a research graph [5]. In cases where unstructured textual data is available (e.g. graph nodes that correspond to scientific articles), existing approaches are incapable of simultaneously exploiting both the structural and the textual information of the graph.

To remedy the above weakness, this chapter proposes the construction and utilization of a scientific knowledge graph where structured and unstructured data co-exist (e.g. document, author and word nodes). Building on our previous work, we represent the documents of a scientific graph as a *graph-of-docs* [6–8]. This enables us to exploit both the structural and textual characteristics of a research graph, and accordingly build a framework incorporating algorithms for tasks such as link prediction to discover future collaborations, text categorization to pair similar documents in communities studying a certain topic, and feature selection to identify the key features of the documents under consideration. The proposed approach uses the Neo4j graph database (https://neo4j.com) for the representation of the knowledge graph. For the implementation of our experiments, we use the Python programming language and the scikit-learn machine learning (ML) library (https://scikit-learn.org).

To evaluate the outcome of this chapter, we assess the proposed framework against different combinations of link prediction measures, which utilize only the structural information of a research graph. Our performance metrics include the accuracy, the precision, and the recall for each of the ML models considered. For our experiments, we use the COVID-19 Open Research Dataset (CORD-19). To examine whether our approach is affected by the size of the dataset (e.g. overfits or underfits), we extract and consider nine different well-balanced datasets. The experimental results demonstrate state-of-the-art accuracy in the link prediction problem. The remainder of the chapter is organized as follows: Sect. 16.2 introduces background issues and comments on related work; the proposed framework is thoroughly presented and evaluated in Sects. 16.3 and 16.4, respectively; finally, concluding remarks and future work directions are outlined in Sect. 16.5.

16.2 Background Issues

For the discovery of future research collaborations, the proposed approach exploits a combination of natural language processing (NLP), graph-based text representation, graph theory and knowledge graph techniques.

16.2.1 Graph Measures and Indices

Diverse graph measures and indices to capture knowledge related to the structural characteristics of a graph have been proposed in the literature [9]. Below, we mention a small subset of them, which is used in our approach. We define $|S|$ as the number of elements found in a set S.

The C*ommon Neighbors* measure, denoted by *CN (a, b)*, calculates the number of nodes that are common neighbors for a pair of nodes a and b [10]. It is defined as:

$$CN(a,\ b) = |\Gamma(a) \cap \Gamma(b)| \tag{16.1}$$

where $\Gamma(x)$ denotes the set of neighbors of a node x.

The *Total Neighbors* measure, denoted by *TN (a, b)*, takes into consideration all neighbors of a pair of nodes a and b (and not only the common ones as is the case in the previous measure). It is defined as:

$$TN(a,\ b) = |\Gamma(a) \cup \Gamma(b)| \tag{16.2}$$

The *Preferential Attachment* measure, denoted by *PA (a, b)*, calculates the product of the in-degree values of a pair of nodes a and b [11]. This measure assumes that two highly connected nodes are far more likely to be connected in the future, in contrast to two loosely connected ones. This measure is defined as:

$$PA(a,\ b) = |\Gamma(a)| * |\Gamma(b)| \tag{16.3}$$

The *Adamic Adar* measure, denoted by *AA (a, b)*, calculates the sum of the inverse logarithm of the degree of the set of neighbors shared by a pair of nodes a and b [12]. This measure assumes that nodes of a low degree are more likely to be influential in the future. It is defined as:

$$AA(a,\ b) = \sum_{c \in \Gamma(a) \cap \Gamma(b)} \left(\frac{1}{\log|\Gamma(c)|} \right) \tag{16.4}$$

Finally, the *Jaccard Coefficient* index, denoted by *J (a, b)*, resembles the CN measure mentioned above; however, it differs slightly in that, for a pair of nodes a and b, it considers the amount of the intersection of their neighbor nodes over the union of them [13]. It is defined as:

$$J(a,\ b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{|\Gamma(a) \cup \Gamma(b)|} \tag{16.5}$$

16.2.2 Graph-Based Text Representations

The *graph-of-words* textual representation [14] represents each document of a corpus as a single graph. In particular, each graph node corresponds to a unique word of a document and each edge denotes the co-occurrence between two words within a sliding window of text. Rousseau et al. [15] suggest that a window size of four seems to be the most appropriate value, in that it does not sacrifice either the performance or the accuracy of the ML models. Compared to the *bag-of-words* representation, it enables a more sophisticated feature engineering process due to the fact that it takes into consideration the co-occurrence between the terms. In any case, the limitations of the graph-of-words text representation are that: (i) it is unable to assess the importance of a word for a whole set of documents; (ii) it does not allow for representing multiple documents in a single graph, and (iii) it is not easily expandable to support more complicated data architectures.

16.2.3 Graph-Based Feature Selection

Various promising graph-based feature selection approaches have been already proposed in the literature. Rousseau et al. [15] proposed several combinations and arrangements of popular frequent subgraph mining techniques, such as *gSpan* [16], *Gaston* [17] and *gBoost* [18], to achieve unsupervised feature selection by utilizing the k-core subgraph. Especially, in order to get a performance boost, Rousseau and his colleagues build on the concept of a k-core subgraph to compress the most dense parts of the graph representation. Their experimental results indicate a significant increase in accuracy compared to common classification approaches. Henni et al. [19] applied centrality algorithms, such as *PageRank*, to calculate a centrality measure of each graph feature and accordingly select the most important ones. Fakhraei et al. [20] build on combinations of graph algorithms that belong in different classes, aiming to track strongly connected graph features. Such algorithms include the *Louvain* community detection algorithm and the *PageRank* centrality algorithm to discover influential nodes and other user defined graph measures.

Other approaches rely on recursively filtering out features in terms of reducing the existing feature space. For instance, one of them re-applies PageRank to find the most influential features in the feature space [21]. These approaches use graph-connected features to include contextual information, as modelled implicitly by a graph structure, using edges that describe connections among real data. They aim to reduce ambiguity in feature selection and improve accuracy in traditional ML methods.

16.2.4 Graph-Based Text Categorization

Many interesting approaches have been also proposed in the literature for the graph-based text categorization process. Depending on their underlying methods, these can be classified into two basic categories: (i) these that utilize *frequent subgraph mining* for feature extraction, and (ii) those that build on *graph kernels*. Well known frequent subgraph mining techniques were mentioned in the previous subsection. Rousseau et al. [15] propose several combinations of these methods, ranging from unsupervised feature mining using *gSpan* to unsupervised feature selection by utilizing the *k*-core subgraph.

Nikolentzos et al. [22] make a significant contribution to previous approaches, with their work on 'graph kernel'—based algorithms. A graph kernel is a measure that calculates the similarity between two graphs. For instance, a document similarity algorithm based on shortest path graph kernels has been proposed; common ML classifiers such as support vector machines (SVM) and k-nearest neighbors (*k*-NN) can use the results of this algorithm as a distance measure. Their experimental results indicate that classifiers that utilize graph kernel algorithms outperform several classical approaches. Siglidis et al. [23] collect several popular graph kernel libraries into a single unified framework, namely the *GraKeL Python library*, and provide a user-friendly API (similar to that of scikit-learn) that enables one to augment the library with new and custom graph kernels.

16.2.5 Graph-Based Link Prediction

As far as the discovery of future research collaborations using link prediction techniques is concerned, works that are closer to our approach are those of Liben-Nowell and Kleinberg [24–29]. Specifically, Liben-Nowell and Kleinberg [24] rely only on network topology aspects of a co-authors network, and the proximity of a pair of nodes to calculate the probability of future research collaborations between them. Sun et al. [27] propose the use of structural properties to predict future research collaborations in heterogeneous bibliographic networks, where multiple types of nodes (e.g. venues, topics, papers, authors) and edges (e.g. publish, mention, write, cite, contain) co-exist. They exploit the relationships between the papers to improve the accuracy of their link prediction algorithm.

Guns and Rousseau [25] recommend potential research collaborations using link prediction techniques and a random forest classifier. For each pair of nodes of a co-authorship network, they calculate a variety of topology-based measures such as Adamic Adar and Common Neighbors, and they combine them with location-based characteristics related to the authors. Hence, they propose future collaborations based on the location of the authors and their position on the co-authorship network. Huang et al. [26] construct a co-authorship network for the Computer Science field that represents research collaborations from 1980 to 2005. They rely on classical

statistical techniques and graph theory algorithms to describe the properties of the constructed co-authorship network. The dataset used contains 451,305 papers from 283,174 authors.

Yu et al. [28] utilize link prediction algorithms to discover future research collaborations in medical co-authorship networks. For a given author, they attempt to identify potential collaborators that complement her as far as her skillset is concerned. They calculate common topological and structural measures for each pair of author nodes, including Adamic Adar, Common Neighbors and Preferential Attachment. ML models are used for the identification of possible future collaborations.

Chuan et al. [29] propose a new content similarity algorithm for link prediction in co-authorship networks, namely *LDAcosin*. This algorithm initially performs topic modelling using the LDA model to produce a feature vector for each paper, and then calculates the similarity between authors by using cosine similarity between the produced vectors.

For a broader link prediction perspective, we refer to Fire et al. [30], Julian and Lu [31] and Panagopoulos et al. [32], these works describe approaches concerning the task of predicting possible relationship types between nodes (e.g. friendships in social networks).

16.3 The Proposed Framework

In this section, we propose a framework that builds on the concept of the *graph-of-docs* to support and eventually augment the quality of predicting future research collaborations.

16.3.1 Graph-Based Text Representation

As mentioned in the previous section, to remedy the shortcomings of the graph-of-words representation, Giarelis et al. [6–8] have proposed the *graph-of-docs* representation, which depicts and elaborates multiple textual documents as a single graph. This representation enables us to store different types of nodes and edges in a graph, ranging from node types such as `'document'` and `'word'` to edge types such as `'is_similar'`, `'connects'` and `'includes'`. In addition, it allows us to explore the significance of a term not just in terms of a single document but rather across many documents. Moreover, the proposed representation permits us to abstract each graph of words by using a document node. Finally, it supports relationship edges between documents, thus enabling the calculation of important metrics as far as the documents are concerned (e.g., spotting communities of similar documents, recognizing important document which are representative for the corpus, cluster documents that share the same topic in communities without any prior knowledge, etc.).

The graph-of-docs representation builds a directed dense graph which maintains all the connections between the documents and the words of a corpus. Each unique document node connects to all the unique word nodes that it includes, using the `'includes'` edge type; the `'connects'` edge types are applied to link two word nodes and designate their co-occurrence within a specific sliding text window. In the end, an `'is_similar'` edge type is used to connect a pair of document nodes and indicate their contextual similarity; this is done by utilizing the *Jaccard similarity index*, since it deals only with the percentage of common words, ignoring their document frequency.

The above transformation of a set of documents into a graph model enables the reduction of various NLP problems to well-studied graph problems, which can be tackled by employing techniques from graph theory [15]. These techniques investigate important graph properties, such as node centrality and frequent subgraphs, which are applied respectively to extract meaningful keywords and to discover similar documents.

In this chapter, we utilize the graph-of-docs model to represent the textual data of a knowledge graph. We argue that the accuracy of common NLP and text mining tasks can be improved by adopting the proposed graph-of-docs representation. The proposed representation: (i) enables the investigation of the importance of a term into a whole corpus of documents, and (ii) allows multiple node types to co-exist in the same graph, thus being easily expandable and adaptable to more complex data.

16.3.2 Graph-Based Feature Selection

The proposed graph-based feature selection process follows four steps. Firstly, a document similarity subgraph is created, based on the assumption that subgraphs of the entire graph-of-docs graph describing similar documents have common word nodes and similar structural characteristics. This enables us to calculate the similarity between two documents by utilizing classical similarity measures. The similarity subgraph consists of document nodes and edges of the `'is_similar'` type, which store the similarity score between two nodes.

Secondly, by exploiting the document similarity subgraph, we identify communities (groups) of contextually similar documents using the `'score'` property of the `'is_similar'` type edges as a distance value. This is made possible by the use of the Louvain community detection algorithm [33].

Thirdly, given the fact that documents belonging to the same community are contextually similar, we presume that it is also very likely that they share common terms. Aiming to retrieve the *top-N* most important terms for all documents belonging to the same community, our algorithm ranks them firstly by their document frequency and secondly by their PageRank score, both in descending order.

Finally, we perform feature selection for the whole document corpus by merging the top-N features of each community. This reduces the number of the candidate features, which results in accelerating the feature selection process, thus mitigating

the effects of the 'curse-of-dimensionality' phenomenon and enabling the training of more reliable ML models.

16.3.3 Graph-Based Text Categorization

Generally speaking, subgraphs extracted from similar documents share common word nodes as well as similar structural characteristics. This allows us to measure the similarity between two documents either by using classical data mining similarity measures, such as the *Jaccard* or *cosine* similarity, or by utilizing frequent subgraph mining techniques (see Sect. 16.2.3). In our current approach, we construct a similarity subgraph that contains document nodes and edges of type `'is_similar'`. It is evident that the creation of that subgraph is not practical in approaches that consider each document individually.

In the aforementioned subgraph, we group documents in contextually similar communities, by considering as a distance value the `'score'` property of the `'is_similar'` edge types. A plethora of community detection algorithms can be found in the literature, including Louvain [33], Label Propagation [34] and Weakly Connected Components (Monge and Elka [35], an in-depth review of these algorithms can be found in Fortunato [36] and Yang et al. [37]. Since each graph community contains contextually similar documents, there is an increased likelihood for each community to contain documents that belong to the same class, as identified by a text categorization task. Hence, we can easily deduce the document class either by utilizing the most frequent class in its community or by executing a nearest neighbors' algorithm (such as the k-nearest neighbors).

16.3.4 Graph-Based Link Prediction

The CORD-19 dataset used in our work consists of multiple textual documents (i.e. scientific papers) and a metadata file (in.csv format) that contains information about the papers themselves along with their authors and affiliations. The proposed ML pipeline for predicting future collaborations includes the following five steps (Fig. 16.1):

Data preprocessing. In this initial step, we preprocess the plain text of the abstract of each paper. We start by tokenizing the data in a list of terms. From this list, we first remove the English stop words; the remaining significant terms are then cleaned by unnecessary Unicode symbols, punctuation and leading whitespace.

The graph-of-docs representation. In this step, we use the list of significant terms obtained from the previous one. We start by creating each term in the graph database, without adding duplicate terms. Each term is connected to the next one in the list as long as it is part of the same sentence.

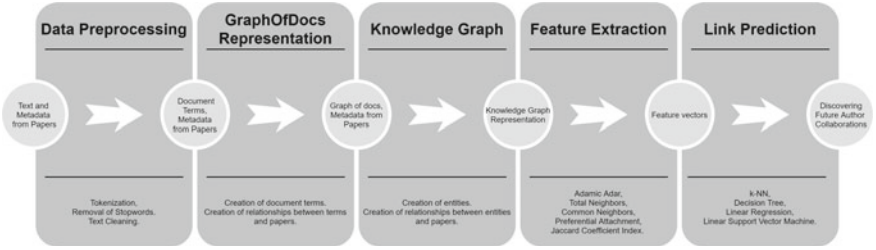

Fig. 16.1 The proposed ML pipeline for discovering future research collaborations

The sliding window size, by which we connect the terms, is in the range [2, 8]; however, as indicated by diverse experimental results in the literature, a window size of 4 seems ideal [15]. The connection between the terms is created in the database as an undirected edge connecting all terms in the specified window. This edge also contains a number, namely the *co-occurrence score*, which measures the number of co-appearances of a pair of terms in each iteration step of the text parsing process. Common edges between texts are aggregated in terms of their co-occurrence score. This implies that no duplicate edges are introduced in our graph, which reduces the memory footprint.

As long as the *graph-of-words* representation (for a single document) has been created, we then create a node in the database representing the paper itself, which is directly connected to all of its terms. This is crucial, since it allows us to compare papers given their common words.

Knowledge graph. In this step, we utilize the metadata of each paper. We start by creating nodes for all authors and their affiliations. Then, we link authors with their affiliations and the papers they have authored by using different types of edges. Moreover, to discover authors of similar papers who work on the same field but have not collaborated so far. This is accomplished by comparing the words of papers between a pair of authors of the papers under consideration. The aforementioned connection also allows us to generate a new knowledge subgraph, called the co-authorship graph, which connects all authors who have already collaborated in the authorship of papers, with an edge indicating the year of their first collaboration.

Feature extraction. Our goal in this step is to extract features for classification purposes. For each pair of authors, we apply various link prediction measures, which result in a numerical score (i.e. a positive number that indicates the likelihood of a future collaboration, or zero if there is no such likelihood). These measures exploit the structural characteristics of the co-authorship graph. The final list of features contains a pair of authors' ids and the value of each measure, which feed the final step of the pipeline.

Link prediction. In this step, we utilize the aforementioned features as input to a classification process. This process classifies each pair of authors by assigning a label `'1'` if the authors may work together in the future, or a label `'0'` in the opposite case. In other words, the link prediction problem is reduced to a binary classification problem, aiming to discover future research collaborations.

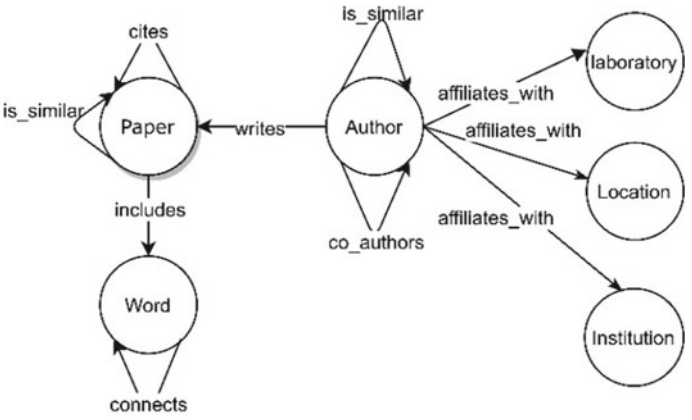

Fig. 16.2 The data schema of the scientific knowledge graph

Our knowledge graph allows diverse types of entities and relationships to co-exist in a the same graph data schema, including entity nodes with types such as `'Paper'`, `'Author'`, `'Laboratory'`, `'Location'`, `'Institution'` and `'Word'`, and relationship edges with types such as `'is_similar'`, `'cites'`, `'writes'`, `'includes'`, `'connects'`, `'co_authors'` and `'affiliates_with'` (see Fig. 16.2).

A `'Paper'` entity represents a scientific paper or document. An `'Author'` entity represents an author of a scientific paper or document. The `'Laboratory'` entity represents the laboratory of an author. The `'Location'` entity represents the location of a laboratory. The `'Institution'` entity represents the institution of an author. Each `'Word'` entity corresponds to a unique word of a scientific paper or document.

An `'includes'` relationship connects a `'Word'` with a `'Paper'` entity. It marks the presence of a specific word to a certain paper. A `'connects'` relationship is only applicable between two `'Word'` entities and denotes their co-occurrence within a predefined sliding window of text. The subgraph constructed by the `'Word'` and `'Paper'` entities, as well as the `'includes'`, `'connects'` and `'is_similar'` relationships, corresponds to the graph-of-docs representation of the textual data of the available papers (see Fig. 16.3).

An `'is_similar'` relationship links either a pair of `'Paper'` or `'Author'` nodes. In the former case, it denotes the graph similarity of the graph-of-words representation of each paper. In the latter, it denotes the graph similarity between the graph-of-docs representations associated to the two authors. The subgraph that consists of the `'Author'` entities and the `'is_similar'` relationships corresponds to the authors similarity subgraph.

A `'cites'` relationship links two `'Paper'` nodes. A `'writes'` relationship links an `'Author'` with a `'Paper'` entity. An `'affiliates_with'` relationship connects an `'Author'` entity with a `'Laboratory'`, `'Location'`

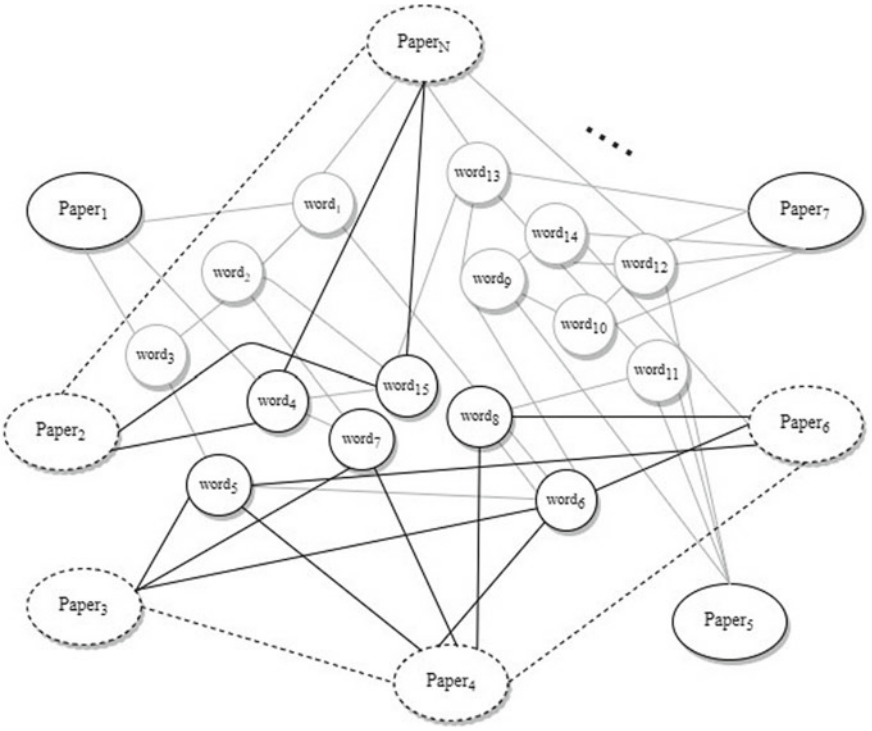

Fig. 16.3 Representing textual data of papers using the graph-of-docs model (relationships between papers are denoted with dotted lines). The graph-of-docs representation is associated to the `'Paper'` and `'Word'` entities, and the `'includes'`, `'connects'` and `'is_similar'` relationships of the scientific knowledge graph

or `'Institution'` entity. A `'co_authors'` relationship denotes a research collaboration between the connected `'Author'` entities. The subgraph constructed of the available `'Author'` entities and the `'co_authors'` relationships corresponds to the co-authors' subgraph.

The produced knowledge graph enables the utilization of well-studied graph algorithms, which in turn assists in gaining insights about various tasks, such as finding experts nearby based on the 'Location' entities, recommending similar research work, and discovering future research collaborations; this chapter focuses on the last of these tasks.

For the discovery of future research collaborations, we employ various link prediction and ML techniques. Particularly, we reduce the problem of predicting future research collaborations to the common binary classification problem. By using a binary classifier, we are able to predict the presence or the absence of a `'co_authors'` relationship between two `'Author'` entities, and thus build a

link prediction algorithm for the discovery of future research collaborations. Available binary classifiers include logistic regression, k-nearest neighbors, linear support vector machines, decision tree, and neural networks [38].

16.4 Experiments

For the implementation and evaluation of our approach, we used the Python programming language and the scikit-learn ML library (https://scikit-learn.org). The Neo4j graph database (https://neo4j.com) has been utilized for the representation of the graph-of-docs and the corresponding knowledge graph. The full code, datasets, and evaluation results of our experiments are freely available at https://github.com/imis-lab/book-chapter.

16.4.1 Cord-19

The COVID-19 Open Research Dataset (CORD-19) [39, 40] contains information about 63,000 research articles, related to COVID-19, SARS-CoV-2 and other similar coronaviruses. It is freely distributed from the Allen Institute for AI and Semantic Scholar (https://www.semanticscholar.org/cord19). The articles in CORD-19 have been collected from popular scientific repositories and publishing houses, including Elsevier, bioRxiv, medRxiv, World Health Organization (WHO) and PubMed Central (PMC). Each scientific article in CORD-19 has a list of specific attributes, namely 'citations', 'publish time', 'title', 'abstract' and 'authors', while the majority of the articles (51,000) also includes a 'full text' attribute. Undoubtfully, the CORD-19 dataset is a valuable source of knowledge as far as the COVID-19-related research is concerned; however, the fact that the majority of the data included is unstructured text renders a set of limitations in its processing. As advocated in the literature, the exploitation of a graph-based text representation in combination with a knowledge graph seems to be a promising step towards structuring this data [4, 5, 41]. For the construction of our scientific knowledge graph, we utilize the 'abstract', 'authors' and 'publish time' attributes of each scientific article. We do not exploit the 'full text' attribute due to hardware limitations, however, we assume that the abstract of a paper consists a representative piece of its full text.

16.4.2 Experimental Setup

Selection of measures and metrics. To construct the authors similarity subgraph and to populate the edges of the `'Author'.'is_similar'` type, we use the

Jaccard similarity index, since it deals only with the percentage of common set of words versus all words, ignoring their document frequency.

Construction of datasets for the link prediction problem. To test whether our approach performs well and does not overfit, regardless of the sample size of the dataset, we extract nine different datasets from the original one, corresponding to different volumes of papers (ranging from 1536 to 63,023). For the sample creation, we utilize (i) the authors similarity subgraph, and (ii) the co-authors subgraph (i.e. the subgraph generated from the `'co_authors'` edges; it is noted that edges also store the year of the first collaboration between authors, as a property). The features of a sample encapsulate either structural or textual characteristics of the whole knowledge graph (e.g. the similarity between the papers of two authors). Furthermore, each sample describes the relationship between two `'Author'` nodes of the knowledge graph. We consider the classical link prediction algorithms as the baseline methods to be compared against our approach, which they only utilize the structural characteristics of the graph.

The features of a sample are analytically described in Table 16.1. Each of the nine datasets consists of a different number of randomly chosen samples. All datasets are balanced, in that the number of positive and negative samples are equal (see Table 16.2). To examine whether the features taken into account each time affect the efficiency of the ML models, we execute a set of experiments with different combinations of selected features (see Table 16.3). Finally, it is noted that the samples for the training subset are selected from an earlier instance in time of the co-authors subgraph, which is created from `'co_authors'` edges first appeared within or before the year of 2013; respectively, the samples of the testing subset include

Table 16.1 A detailed explanation of the features of a sample. Each feature is associated to either a structural or a textual relationship between two given `'Author'` nodes

Feature	Description	Type
adamic_adar	The sum of the inverse logarithm of the degree of the set of common neighbor `'Author'` nodes shared by a pair of nodes	Structural
common_neighbors	The number of neighbor `'Author'` nodes that are common for a pair of `'Author'` nodes	Structural
preferential_attachment	The product of the in-degree values of a pair of `'Author'` nodes	Structural
total_neighbors	The total number of neighbor `'Author'` nodes of a pair of `'Author'` nodes	Structural
similarity	The textual similarity of the graph-of-docs graphs of two `'Author'` nodes. The Jaccard index is used to calculate the similarity	Textual
label	The existence or absence of a `'co_authors'` edge between two `'Author'` nodes. A positive label (1) denotes the existence, whereas the absence is denoted by a negative label (0)	Class

Table 16.2 Number of samples (|samples|) for each dataset (the number of positive and negative samples of the training and testing subsets are fully balanced); a positive sample denotes the existence of a `'co_authors'` edge between two `'Author'` nodes, while a negative sample denotes the absence of such an edge

	Training subset |samples|	Testing subset |samples|
Dataset 1	668	840
Dataset 2	858	1566
Dataset 3	1726	2636
Dataset 4	3346	7798
Dataset 5	5042	12,976
Dataset 6	5296	16,276
Dataset 7	6210	25,900
Dataset 8	8578	34,586
Dataset 9	13,034	49,236

Table 16.3 Combinations of features aiming to test how different set of features affect the performance of an ML model; "top n" indicates the top number of features, extracted from each community of similar documents

Combination name	Features included
Structural characteristics and authors similarity top 5 (STR-SIM_top5)	adamic_adar, common_neighbors, preferential_attachment, total_neighbors, similarity_top_5
Structural characteristics and authors similarity top 100 (STR-SIM_top100)	adamic_adar, common_neighbors, preferential_attachment, total_neighbors, similarity_top_100
Structural characteristics and authors similarity top 250 (STR-SIM_top250)	adamic_adar, common_neighbors, preferential_attachment, total_neighbors, similarity_top_250
Structural characteristics (STR-baseline)	adamic_adar, common_neighbors, preferential_attachment, total_neighbors

`'co_authors'` edges created after 2013. This separation in time ensures that we avoid any data leakage between the training and testing subsets [24].

16.4.3 Evaluation

To evaluate the effectiveness of our approach, we assess how the performance of various binary classifiers is affected by the similarity features. The list of the binary classifiers considered in this chapter includes: logistic regression (LR), *k*-nearest neighbors (50NN), linear support vector machines with a linear kernel function (LSVM), support vector machines with a RBF kernel function (SVM), decision tree (DT) and neural networks (NN). To normalize the features from our datasets, we employ the min–max normalization procedure. An extensive list of experiments using various classifiers along with different hyperparameter configurations can be found

Table 16.4 Hyperparameter configurations in Scikit-Learn for each of the utilized binary classifiers; further hyperparameter configurations are described in the Scikit-Learn documentation

Binary classifier	Hyperparameter configuration
LR	solver = 'lbfgs', multi_class = 'ovr'
50NN	k = 50, weights = 'uniform'
LSVM	kernel = 'linear'
SVM	kernel = 'rbf'
DT	max_depth = 5
NN	solver = 'adam', activation = 'relu', hidden_layers = 100 × 50

on the GitHub repository of this chapter (https://github.com/imis-lab/book-chapter). The hyperparameter configurations can be also found in Table 16.4. Our performance metrics include the *accuracy*, *precision* and *recall* of the binary classifiers.

The obtained results indicate that the inclusion of the similarity features (i) increase the average accuracy, precision and recall scores, and (ii) decrease the standard deviation of the aforementioned scores (Table 16.5). The decrement of the standard deviation in the accuracy score indicates that our approach is reliable regardless of the size of the given dataset. Furthermore, by comparing the average precision score to the average recall score, we conclude that our approach predicts most of the future collaborations correctly. The best average accuracy score is achieved by the LSVM classifier, using the STR-SIM_top100 and STR-SIM_top250 feature combinations.

As far as link prediction is concerned, our algorithm differs from existing ones in that it considers both the textual similarity between the abstracts of the papers for each pair of authors and the structural characteristics of the associated `'Author'` nodes, aiming to predict a future collaboration between them. The utilization of the textual information in combination with the structural information of a scientific knowledge graph results in better and more reliable ML models, which are less prone to overfitting. Contrary to existing algorithms for the discovery of future research collaborations, our approach exploits structural characteristics and does not ignore the importance of the information related to the unstructured text of papers written by authors. Finally, existing approaches that concentrate only on the exploitation of unstructured textual data rely heavily on NLP techniques and textual representations, which in turn necessitate the generation of sparse feature spaces; hence, in such approaches, the effects of the 'curse-of-dimensionality' phenomenon re-emerge.

16.5 Conclusions

This chapter considers the problem of discovering future research collaborations as a link prediction problem applied on scientific knowledge graphs. The proposed approach integrates into a single knowledge graph both structured and unstructured

Table 16.5 Mean (AVG), minimum (MIN), maximum (MAX) and standard deviation (SD) of accuracy, precision and recall metrics per text classifier for each combination of selected features on the nine different datasets.

Method		Accuracy				Precision				Recall			
		AVG	MIN	MAX	SD	AVG	MIN	MAX	SD	AVG	MIN	MAX	SD
LR	STR-SIM_top5	0.9550	0.9374	0.9646	0.0077	0.9563	0.9063	0.9798	0.0223	0.9545	0.9359	0.9757	0.0118
	STR-SIM_top100	**0.9555**	0.9381	0.9651	0.0076	**0.9573**	0.9074	0.9814	0.0224	0.9544	0.9358	0.9757	0.0119
	STR-SIM_top250	**0.9555**	0.9381	0.9651	0.0076	**0.9573**	0.9074	0.9814	0.0224	0.9543	0.9358	0.9757	0.0119
	STR-baseline	0.9549	0.9381	0.9646	0.0074	0.9551	0.9074	0.9809	0.0223	**0.9554**	0.9334	0.9757	0.0131
50NN	STR-SIM_top5	0.9587	0.9387	0.9754	0.0110	0.9343	0.8935	0.9673	0.0207	0.9874	0.9810	0.9962	0.0041
	STR-SIM_top100	**0.9591**	0.9361	0.9761	0.0113	**0.9347**	0.8894	0.9676	0.0211	0.9879	0.9833	0.9962	0.0037
	STR-SIM_top250	**0.9591**	0.9361	0.9762	0.0113	0.9346	0.8894	0.9676	0.0211	0.9880	0.9833	0.9962	0.0037
	STR-baseline	0.9555	0.9253	0.9753	0.0142	0.9284	0.8717	0.9653	0.0255	**0.9881**	0.9818	0.9974	0.0046
LSVM	STR-SIM_top5	0.9592	0.9323	0.9716	0.0117	0.9414	0.8842	0.9733	0.0262	**0.9804**	0.9698	0.9949	0.0087
	STR-SIM_top100	**0.9593**	0.9291	0.9718	0.0128	0.9418	0.8810	0.9735	0.0274	0.9802	0.9700	0.9929	0.0082
	STR-SIM_top250	**0.9593**	0.9291	0.9718	0.0128	**0.9419**	0.8810	0.9735	0.0274	0.9802	0.9700	0.9929	0.0082
	STR-baseline	0.9586	0.9310	0.9705	0.0120	0.9414	0.8840	0.9719	0.0261	0.9791	0.9691	0.9929	0.0082
SVM	STR-SIM_top5	0.9488	0.8723	0.9742	0.0291	0.9162	0.7984	0.9633	0.0461	**0.9915**	0.9860	0.9962	0.0029
	STR-SIM_top100	**0.9516**	0.8691	0.9748	0.0302	0.9210	0.7937	0.9639	0.0475	0.9915	0.9862	0.9974	0.0030
	STR-SIM_top250	**0.9516**	0.8691	0.9748	0.0302	**0.9211**	0.7937	0.9642	0.0475	0.9915	0.9862	0.9974	0.0030
	STR-baseline	0.9460	0.8263	0.9744	0.0434	0.9146	0.7431	0.9646	0.0631	0.9908	0.9850	0.9974	0.0033
DT	STR-SIM_top5	0.9190	0.8036	0.9770	0.0628	0.8748	0.7179	0.9654	0.0901	0.9938	0.9895	1.0000	0.0037
	STR-SIM_top100	0.9193	0.8036	0.9770	0.0624	0.8752	0.7179	0.9654	0.0896	0.9938	0.9896	1.0000	0.0038
	STR-SIM_top250	0.9193	0.8036	0.9770	0.0624	0.8753	0.7179	0.9654	0.0896	0.9938	0.9896	1.0000	0.0038

(continued)

Table 16.5 (continued)

Method		Accuracy				Precision				Recall			
		AVG	MIN	MAX	SD	AVG	MIN	MAX	SD	AVG	MIN	MAX	SD
	STR-baseline	**0.9219**	0.8262	0.9788	0.0574	**0.8775**	0.7420	0.9671	0.0841	**0.9941**	0.9894	1.0000	0.0036
NN	STR-SIM_top5	0.9290	0.8116	0.9796	0.0555	0.8885	0.7272	0.9681	0.0806	0.9930	0.9887	0.9974	0.0025
	STR-SIM_top100	**0.9314**	0.8167	0.9803	0.0558	**0.8924**	0.7326	0.9691	0.0813	**0.9931**	0.9895	0.9974	0.0023
	STR-SIM_top250	0.9313	0.8155	0.9803	0.0562	0.8923	0.7313	0.9691	0.0817	0.9931	0.9894	0.9974	0.0024
	STR-baseline	0.9278	0.8008	0.9816	0.0613	0.8881	0.7155	0.9751	0.0871	0.9931	0.9883	0.9987	0.0031

Bold font indicates the best method for each ML model as far as the mean and the standard deviation value of each individual metric are concerned

textual data using the graph-of-docs text representation. For the required experimentations, we generated nine different datasets using the CORD-19 dataset. For evaluation purposes, we assessed our approach against several link prediction settings, which use various combinations of a set of available features. The evaluation results demonstrate state-of-the-art average accuracy, precision and recall of the future collaborations prediction task. However, we expect that these results will be improved through the employment of *contextual similarity functions* that are based on graph kernels [22].

In any case, our approach has a performance issue, since the time required to build the scientific knowledge graph increases radically with the number of graph nodes. Aiming to address the above limitation, while also enhancing the performance and advancing the applicability of our approach, our future work directions include: (i) the utilization of in-memory graph databases in combination with Neo4j; (ii) the experimentation with word, node and graph embeddings [42–44],(iii) the integration of other scientific research graphs such as *OpenAIRE* [45] and *Microsoft Academic Graph* [46], and (iv) the integration and meaningful exploitation of our approach into collaborative research environments [47].

As far as the CORD19 dataset is concerned, it is worth noting here that it is increasingly explored nowadays in the investigation of various research topics. For instance, Colavizza et al. [48] attempt to produce a scientific overview of the dataset by employing various approaches such as a statistical analysis of the dataset's metadata, unsupervised key-phrase extraction, supervised citation clustering, and LDA topic modelling. Papadopoulos et al. [49] aim to visualize in a graph various triplet fact in the form of subject-predicate-object (i.e. a knowledge graph approach). They achieve their research goal by combing a set of pre-trained BERT models and keyword extraction tools. Guo et al. [50] aim to augment the task of semantic textual similarity (STS) by producing an ad-hoc dataset (namely, *CORD19STS*, which aims to alleviate the poor performance of generalized STS models by fine-tuning a BERT-like deep learning language model. Finally, Wang et al. [39, 40] introduce a weakly supervised Named Entity Recognition model by optimizing pre-trained *spaCy* models (https://spacy.io/), ranging from general English language to domain specific biology terms in English.

Acknowledgements The work presented in this chapter is supported by the OpenBio-C project (www.openbio.eu), which is co-financed by the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH—CREATE—INNOVATE (Project id: T1EDK- 05275).

References

1. D. Nathani, J. Chauhan, C. Sharma, M. Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)* (2019), pp. 4710–4723

2. S. Vahdati, G. Palma, R.J. Nath, C. Lange, S. Auer, M.E. Vidal, Unveiling scholarly communities over knowledge graphs, in *International Conference on Theory and Practice of Digital Libraries* (Springer, Cham, 2018), pp. 103–115
3. B. Ponomariov, C. Boardman, What is co-authorship? Scientometrics **109**(3), 1939–1963 (2016)
4. Q. Wang, Z. Mao, B. Wang, L. Guo, Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Know. Data Eng. **29**(12), 2724–2743 (2017)
5. N. Veira, B. Keng, K. Padmanabhan, A. Veneris, Unsupervised embedding enhancements of knowledge graphs using textual associations, in *Proceedings of the 28th International Joint Conference on Artificial Intelligence* (AAAI Press 2019), pp. 5218–5225
6. N. Giarelis, N. Kanakaris, N. Karacapilidis, An innovative graph-based approach to advance feature selection from multiple textual documents, in *IFIP International Conference on Artificial Intelligence Applications and Innovations* (Springer, Cham, 2020a), pp. 96–106
7. N. Giarelis, N. Kanakaris, N. Karacapilidis, On a novel representation of multiple textual documents in a single graph, in *Intelligent Decision Technologies 2020—Proceedings of the 12th KES International Conference on Intelligent Decision Technologies (KES-IDT-20)*, ed. by I. Czarnowski, R.J. Howlett, L.C. Jain Split (Croatia, Springer, 2020b)
8. N. Giarelis, N. Kanakaris, N. Karacapilidis, On the utilization of structural and textual information of a scientific knowledge graph to discover future research collaborations: a link prediction perspective, in *Proceedings of the 23rd International Conference on Discovery Science (DS 2020)*, ed. by A. Appice, G. Tsoumakas, Y. Manolopoulos and S. Matwin, vol. 12323 (Springer, Cham, LNAI, 2020c), pp. 437–450
9. Á. Vathy-Fogarassy, J. Abonyi, *Graph-Based Clustering and Data Visualization Algorithms* (Springer, London, 2013)
10. S. Li, J. Huang, Z. Zhang, J. Liu, T. Huang, H. Chen, Similarity-based future common neighbors model for link prediction in complex networks. Sci. Rep. **8**, 1–11 (2018)
11. R. Albert, A. Barabási, Statistical mechanics of complex networks. ArXiv, cond-mat/0106096 (2001)
12. L.A. Adamic, E. Adar, Friends and neighbors on the Web. Soc. Networks **25**, 211–230 (2003)
13. P. Jaccard, Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull. Soc. Vandoise Sci. Nat. **37**, 547–579 (1901)
14. F. Rousseau, M. Vazirgiannis, Graph-of-word and TW-IDF: new approach to ad hoc IR, in *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management* (ACM Press, 2013), pp. 59–68
15. F. Rousseau, E. Kiagias, M. Vazirgiannis, Text categorization as a graph classification problem, in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, vol. 1 (2015), pp. 1702–1712
16. X. Yan, J. Han, gspan: Graph-based substructure pattern mining, in *Proceedings of the IEEE International Conference on Data Mining* (IEEE Press, 2002), pp. 721–724
17. S. Nijssen, J.N. Kok, A quickstart in frequent structure mining can make a difference, in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (ACM Press 2004), pp. 647–652
18. H. Saigo, S. Nowozin, T. Kadowaki, T. Kudo, K. Tsuda, gBoost: a mathematical programming approach to graph classification and regression. Mach. Learn. **75**(1), 69–89 (2009)
19. K. Henni, N. Mezghani, C. Gouin-Vallerand, Unsupervised graph-based feature selection via subspace and PageRank centrality. Expert Syst. Appl. **114**, 46–53 (2018)
20. S. Fakhraei, J. Foulds, M. Shashanka, L. Getoor, Collective spammer detection in evolving multi-relational social networks, in *Proceedings of the 21 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 1769–1778
21. D. Ienco, R. Meo, M. Botta, Using page rank in feature selection, in *SEBD* (2008), pp. 93–100
22. G. Nikolentzos, G. Siglidis, M. Vazirgiannis, Graph Kernels: a survey. arXiv preprint arXiv: 1904.12218 (2019)

23. G. Siglidis, G. Nikolentzos, S. Limnios, C. Giatsidis, K. Skianis, M. Vazirgianis, Grakel: a graph kernel library in python. arXiv preprint arXiv:1806.02193 (2018)
24. D. Liben-Nowell, J.M. Kleinberg, The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. **58**, 1019–1031 (2007)
25. R. Guns, R. Rousseau, Recommending research collaborations using link prediction and random forest classifiers. Scientometrics **101**(2), 1461–1473 (2014)
26. J. Huang, Z. Zhuang, J. Li, C.L. Giles, Collaboration over time: characterizing and modeling network evolution, in *Proceedings of the 2008 International Conference on Web Search and Data Mining* (2008), pp. 107–116
27. Y. Sun, R. Barber, M. Gupta, C.C. Aggarwal, J. Han, Co-author relationship prediction in heterogeneous bibliographic networks, in *2011 International Conference on Advances in Social Networks Analysis and Mining* (IEEE, 2011), pp. 121–128
28. Q. Yu, C. Long, Y. Lv, H. Shao, P. He, Z. Duan, Predicting co-author relationship in medical co-authorship networks. PloS one **9**(7), e101214 (2014)
29. P.M. Chuan, M. Ali, T.D. Khang, N. Dey, Link prediction in co-authorship networks based on hybrid content similarity metric. Appl. Intell. **48**(8), 2470–2486 (2018)
30. M. Fire, L. Tenenboim-Chekina, O. Lesser, R. Puzis, L. Rokach, Y. Elovici, Link prediction in social networks using computationally efficient topological features, in *2011 IEEE Third Int'l Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third Int'l Conference on Social Computing* (2011), pp. 73–80
31. K. Julian, W. Lu, Application of machine learning to link prediction (2016)
32. G. Panagopoulos, G. Tsatsaronis, I. Varlamis, Detecting rising stars in dynamic collaborative networks. J. Infor. **11**, 198–222 (2017)
33. H. Lu, M. Halappanavar, A. Kalyanaraman, Parallel heuristics for scalable community detection. Parallel Comput. **47**, 19–37 (2015)
34. U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007)
35. A. Monge, C. Elkan, An efficient domain-independent algorithm for detecting approximately duplicate database records (1997)
36. S. Fortunato, Community detection in graphs. Phys. Rep. **486**(3–5), 75–174 (2010)
37. Z. Yang, R. Algesheimer, C.J. Tessone, A comparative analysis of community detection algorithms on artificial networks. Sci. Rep. **6**, 30750 (2016). https://doi.org/10.1038/srep30750
38. C.C. Aggarwal, *Machine Learning for Text*. Springer International Publishing (2018)
39. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Eide, P. Mooney, CORD-19: The Covid-19 Open Research Dataset. arXiv preprint arXiv:2004.10706 (2020)
40. X. Wang, X. Song, B. Li, Y. Guan, J. Han, Comprehensive Named Entity Recognition on CORD-19 with Distant or Weak Supervision. arXiv preprint arXiv:2003.12218 (2020)
41. Z. Wang, J. Li, Z. Liu, J. Tang, Text-enhanced representation learning for knowledge graph, in *Proceedings of International Joint Conference on Artificial Intelligent (IJCAI)* (2016), pp. 4–17
42. W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, in *Advances in Neural Information Processing Systems* (2017), pp. 1024–1034
43. T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in *Advances in Neural Information Processing Systems (NeurIPS)* (2013), pp. 3111–3119
44. G. Nikolentzos, P. Meladianos, M. Vazirgiannis, Matching node embeddings for graph similarity, in *Thirty-First AAAI Conference on Artificial Intelligence* (2017)
45. P. Manghi, C. Atzori, A. Bardi, J. Shirrwagen, H. Dimitropoulos, La Bruzzo, S.F. Summan, OpenAIRE Research Graph Dump (Version 1.0.0-beta) . Zenodo. (2019)
46. S. Arnab, S. Zhihong, H.M. Yang Song, B.H. Darrin Eide, W. Kuansan, An overview of microsoft academic service (MAS) and applications, in *Proceedings of the 24th International Conference on World Wide Web (WWW '15 Companion).* (ACM, New York, NY, USA, 2015), pp. 243–246

47. A. Kanterakis, G. Iatraki, K. Pityanou, L. Koumakis, N. Kanakaris, N. Karacapilidis, G. Potamias, Towards reproducible bioinformatics: The OpenBio-C Scientific Workflow Environment. in *Proceedings of the 19th IEEE International Conference on Bioinformatics and Bioengineering (BIBE)* (Athens, Greece, 2019), pp. 221–226
48. G. Colavizza, R. Costas, A. Traag, N. van Eck, T. van Leeuwen, L. Waltman, A Scientometric overview of CORD-19. bioRxiv preprint (2020)
49. D. Papadopoulos, N. Papadakis, A. Litke, A methodology for open information extraction and representation from large scientific corpora: the CORD-19 data exploration use case. Appl. Sci. **10**, 5630 (2020)
50. X. Guo, H. Mirzaalian, E. Sabir, A. Jaiswal, W. Abd-Almageed, CORD19STS: COVID-19 Semantic Textual Similarity Dataset. arXiv preprint arXiv:2007.02461 (2020)