Enhancing the Microsoft Academic Knowledge Graph via Author Name Disambiguation, Publication Classification, and Embeddings

Michael Färber [*], Lin Ao
Institute AIFB, Karlsruhe Institute of Technology (KIT), Germany
E-mail: michael.faerber@kit.edu

Abstract. Although several large knowledge graphs have been proposed in the scholarly field, such graphs are limited with respect to several data quality dimensions such as accuracy and coverage. In this article, we present methods for enhancing the Microsoft Academic Knowledge Graph (MAKG), a recently published large-scale knowledge graph containing metadata about scientific publications and associated authors, venues, and affiliations. Based on a qualitative analysis of the MAKG, we address three aspects. First, we adopt and evaluate unsupervised approaches for large-scale author name disambiguation. Second, we develop and evaluate methods for tagging publications by their discipline and by keywords, facilitating enhanced search and recommendation of publications and associated entities. Third, we compute and evaluate embeddings for all 254 million authors, 210 million papers, 49,000 journals, and 16,000 conference entities in the MAKG based on several state-of-the-art embedding techniques. Finally, we provide statistics for the updated MAKG. Our final MAKG is publicly available at https://makg.org and can be used for the search or recommendation of scholarly entities, as well as enhanced scientific impact quantification.

Keywords: Knowledge Graph, Scholarly Data, Publications, Digital Libraries

1. Introduction

In recent years, knowledge graphs have been proposed and made publicly available in the scholarly field, covering information about entities such as publications, authors, and venues. They can be used for a variety of use cases: (1) Using the semantics encoded in the knowledge graphs and RDF as a common data format, which allows an easy data integration from different data sources, scholarly knowledge graphs can be used for providing advanced search and recommender systems [1] in academia (e.g., recommending publications and citations [2, 3]). (2) The representation of knowledge as a graph and the interlinkage of entities of various entity types (e.g., publications, authors, institutions) allows us to propose novel ways to scientific impact quantification [4]. (3) If scholarly knowledge graphs model the key content of publications, such as data sets, methods, claims, and research contributions [5], they can be used as a reference point for scientific knowledge (e.g., claims) [6], similar to DBpedia and Wikidata in the case of cross-domain knowledge. In the light of the FAIR principles [7] and the overload of scientific information resulting from the increasing publishing rate in the various fields [8], one can envision that researchers' working styles will change considerably over the next few decades [9, 10] and that, in addition to PDF documents, scientific knowledge might be provided manually or semi-automatically via appropriate forms [5] or automatically based on information extraction on the publications' full-texts [4].

The Microsoft Academic Knowledge Graph (MAKG) [11], AMiner [12], OpenCitations [13], AceKG [14],

[*]Corresponding author. E-mail: michael.faerber@kit.edu.

and OpenAIRE [15] are examples of large domain-specific knowledge graphs with millions or sometimes billions of facts about publications and associated entities, such as authors, venues, and fields of study. In addition, scholarly knowledge graphs edited by the crowd [5] and providing scholarly key content [5, 16] have been proposed. Finally, freely available cross-domain knowledge graphs such as Wikidata[1] provide an increasing amount of information about the academic world, although not as systematic as the domain-specific offshoots.

The *Microsoft Academic Knowledge Graph (MAKG)* [11] was published in its first version in 2019 and is peculiar in the sense that (1) it is one of the largest freely available scholarly knowledge graphs (over 8 billion RDF triples as of 2019-09), (2) it is linked to other data sources in the Linked Open Data cloud, and (3) it provides metadata for entities that are – particularly in combination – often missing in other scholarly knowledge graphs (e.g., authors, institutions, journals, fields of study, in-text citations). As of June 2020, the MAKG contains metadata for more than 239 million publications from all scientific disciplines, as well as over 1.38 billion references between publications. As outlined in Sec. 2.2, since 2019, the MAKG has already been used in various scenarios, such as recommender systems [17], data analytics, bibliometrics and scientific impact quantification [4, 18–20], as well as knowledge graph query processing optimization [21].

Despite its data richness, the MAKG suffers from data quality issues arising primarily due to the application of automatic information extraction methods from the publications (see further analysis in Section 2). We highlight as major issues (1) the containment of author duplicates in the range of hundreds of thousands, (2) the inaccurate and limited tagging (i.e., assignment) of publications with keywords given by the fields of study [11], and (3) the lack of embeddings for the majority of MAKG entities, which hinders the development of machine learning approaches based on the MAKG.

In this article, we present methods for solving these issues and apply them to the MAKG, resulting in an enhanced MAKG.

First, we perform author name disambiguation on the MAKG's author set. To this end, we adopt an unsupervised approach to author name disambiguation that uses the rich publication representations in the MAKG and that scales for hundreds of millions of authors. We use ORCID iDs to evaluate our approach.

Second, we develop a method for tagging all publications with fields of study and with a newly generated set of keywords based on the publications' abstracts. While the existing field of study labels assigned to papers are often misleading (see [22] and Sec. 4) and, thus, often not beneficial for search and recommender systems, the enhanced field of study labels assigned to publications can be used, for instance, to search for and recommend publications, authors, and venues, as our evaluation results show.

Third, we create embeddings for all 239 million publications, 243 million authors, 49,000 journals, and 16,000 conference entities in the MAKG. We experimented with various state-of-the-art embedding approaches. Our evaluations show that the ComplEx embedding method [23] outperforms other embeddings in all metrics. To the best of our knowledge, RDF knowledge graph embeddings have not yet been computed for such a large (scholarly) knowledge graph.[2]

Finally, we provide statistics concerning the authors, papers, and fields of study in the newly created MAKG. For instance, we analyze the authors' citing behaviors, the number of authors per paper over the time, and the distribution of fields of study using the disambiguated author set and the new field of study assignments. We incorporate the results of all mentioned tasks into a final knowledge graph, which we provide online to the public at https://makg.org (formerly: http://ma-graph.org) and http://doi.org/10.5281/zenodo.4617285. Thanks to the entity embeddings, the new paper tags, and the disambiguated author set, the enhanced MAKG opens the door to improved scholarly search and recommender systems and advanced scientific impact quantification.

Overall, our contributions are as follows:

- We present and evaluate an approach for *large-scale author name disambiguation*, which can deal with the peculiarities of large knowledge graphs, such as heterogeneous entity types and 243 million author entries.
- We propose and evaluate transformer-based methods for *classifying publications* according to their fields of study based on the publications' abstracts.

[1] https://wikidata.org/

[2] For instance, RDF2Vec [24] was trained on 17 million Wikidata entities. Even DGL-KE [25], a recently published package optimized for training knowledge graph embeddings at a large scale, was evaluated on a benchmark with only 86 million entities.

- We apply state-of-the-art *entity embedding* approaches to provide entity embeddings for 243 million authors, 239 publications, 49,000 journals, and 16,000 conferences, and evaluate them.
- We provide a *statistical analysis* of the newly created MAKG.

Our implementation for enhancing scholarly knowledge graphs can be found online for further use.[3]

The remainder of this article is structured as follows. In Sec. 2, we describe the MAKG, along with typical application scenarios and its wide usage in the real world. We also outline the MAKG's limitations regarding its data quality, thereby providing our motivation for enhancing the MAKG. Subsequently, in Sec. 3, 4, and 5, we describe in detail our approaches to author name disambiguation, paper classification, and knowledge graph embedding computation. In Sec. 6, we describe the schema of the updated MAKG, information regarding the knowledge graph provisioning and statistical key figures of the enhanced MAKG. We provide a conclusion and give an outlook in Sec. 7.

2. Overview of the Microsoft Academic Knowledge Graph

2.1. Schema and Key Statistics

The MAKG is derived from the Microsoft Academic Graph (MAG)[4], a database consisting of tab-separated text files [26]. The MAKG uses the information provided by the MAG as its basis and enriches the content by modeling the data according to linked data principles to generate a Linked Open Data source (i.e., an RDF knowledge graph with resolvable URIs, a public SPARQL endpoint, and links to other data sources). As of June 2020, the MAKG contains metadata for more than 239 million publications (139 million publications with an abstract), 243 million authors, and more than 1.38 billion references between publications.

Table 1 shows the distribution of the entities among the most important entity types. Surprisingly, the MAKG contains more authors than publications. In addition, the number of affiliations (about 26,000) is relatively low, given that all research institutions in all fields should be represented.

Table 1
General statistics for MAG entities (as of 2020-06)

Key	Value
Papers	238,670,900
Papers with Link	224,325,750
Papers with Abstract	139,227,097
Authors	243,042,675
Affiliations	25,767
Journals	48,942
Conferences	4,468
Conference Instances	16,142
Fields of Study	740,460

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-
     syntax-ns#>
PREFIX magc: <https://makg.org/class/>
PREFIX magp: <https://makg.org/property/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX fabio: <http://purl.org/spar/fabio/>
PREFIX org: <http://www.w3.org/ns/org#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?affilName ?citCountAffil
WHERE {
?field rdf:type magc:FieldOfStudy .
?field foaf:name "Machine␣learning"^^xsd:string .
?paper fabio:hasDiscipline ?field .
?paper dcterms:creator ?author .
?author org:memberOf ?affiliation .
?affiliation foaf:name ?affilName .
?affiliation magp:citationCount ?citCountAffil . }
GROUP BY ?affilName ?citCountAffil
ORDER BY DESC(?citCountAffil)
LIMIT 100
```

List. 1: Querying the top 100 institutions in the area of machine learning according to their overall number of citations.

Compared to a previous analysis of the MAG in 2016 [27], the number of instances has increased for all entity types, except for the number of conference instances, which has dropped from 50,202 to 16,142. An obvious reason for this reduction is the data cleaning process. Although the number of journals, authors, and papers have doubled in size compared to the 2016 version [27], the number of conference series and fields of study have nearly quadrupled.

Figure 1 shows how many publications represented in the MAKG have been published per discipline (i.e., level-0 field of study). Medicine, materials science, and computer science occupy the top positions. This was not always the case. According to the analysis of the MAG in 2016 [27], physics, computer science,

[3] https://github.com/lin-ao/enhancing_the_makg

[4] https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/

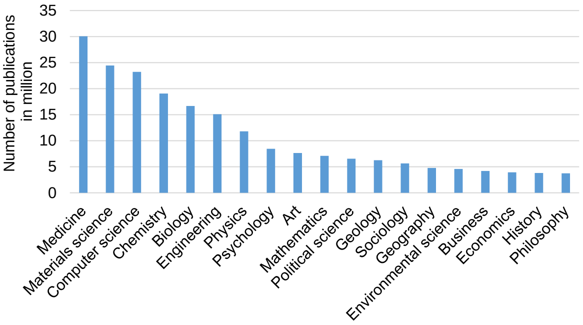

Figure 1. Number of publications per discipline.

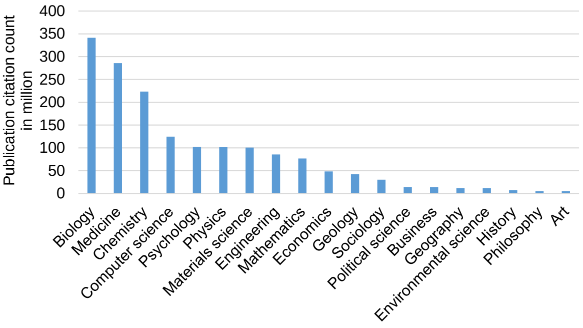

Figure 2. Paper citation count per discipline (i.e., level-0 field of study).

and engineering were the disciplines with the highest numbers of publications. We assume that additional and changing data sources of the MAG resulted in this change.

Figure 2 presents the overall number of publication citations per discipline. Although the descending order of the disciplines is similar to the descending order in terms of the publication count per discipline (see Figure 1), the disciplines biology, medicine, and chemistry have the highest publication citation count. The paper citation count per discipline is not provided by the 2016 MAG analysis [27].

Table 2 shows the frequency of instances per subclass of `mag:Paper`, generated by means of a SPARQL query using the MAKG SPARQL endpoint. List. 1 shows an example of how the MAKG can be queried using SPARQL.

2.2. *Current Usage and Application Scenarios*

The MAKG RDF dumps on Zenodo have been viewed more than 5,000 times and downloaded more than 41,000 times (as of 2021-03-04). As the RDF dumps were also available directly at https://makg.org/rdf-dumps/ (formerly: http://ma-graph.org/rdf-dumps/) until January 2021, the 21,725 visits (since 2019-04-04) to this web page are also relevant.

Table 2
Number of publications by document type

Document Type	**Number**
Journal	85,759,950
Patent	52,873,589
Conference	4,702,268
Book chapter	2,713,052
Book	2,143,939
No type given	90,478,102

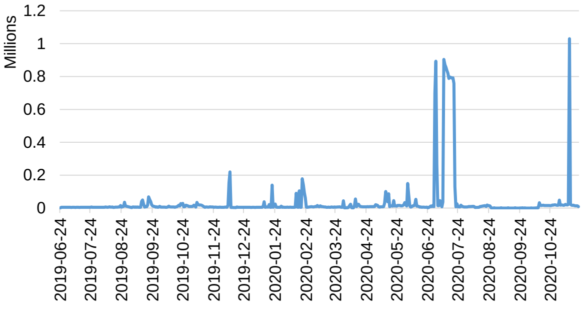

Figure 3. Number of queries.

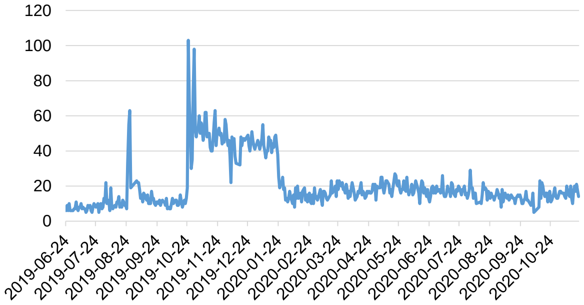

Figure 4. Number of unique users.

Figure 3, 4, and 5 were created based on the log files of the SPARQL endpoint. They show the number of SPARQL queries per day, the number of unique users per day, and which user agents were used to which extent. Overall, the following facts are observable:

- Except for in two months, the number of daily requests increased steadily.
- The number of unique user agents remained fairly constant, apart from a period between October 2019 and January 2020.
- The frequency of “more complex” queries (based on query length) is increasing.

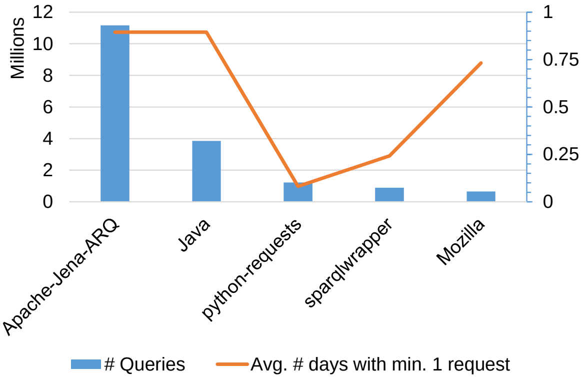

Figure 5. User agents.

Within only one year of its publication in November 2019, the MAKG has been used in diverse ways from various third-parties. Below we list some of them based on the citations of the MAKG publication [11].

Search and recommender systems and data analytics.

- The MAKG has been used for recommender systems, such as paper recommendation [17].
- Scholarly data is becoming increasingly important for businesses. Due to its large number of items (e.g., publications, researchers), the MAKG has been discussed as a data source in enterprises [28].
- The MAKG has been used by non-profit organizations for data analytics. For instance, Nesta uses the MAKG in its business intelligence tools.[5]
- As a unique data source for scholarly data, the MAKG has been used as one of several publicly available knowledge graphs to build a custom domain-specific knowledge graph that considers specific domains of interest [29].

Bibliometrics and scientific impact quantification.

- The Data Set Knowledge Graph [16] provides information about data sets as linked open data source and contains links to MAKG publications in which the data sets are mentioned. Utilizing the publications' metadata in the MAKG allows researchers to employ novel methods for scientific impact quantification (e.g., working on an "h-index" for data sets).
- SoftwareKG [18] is a knowledge graph that links about 50,000 scientific articles from the social sciences to the software mentioned in those articles. The knowledge graph also contains links to other knowledge graphs, such as the MAKG. In this way, the SoftwareKG provides the means to assess the current state of software usage.
- Publications modeled in the MAKG have been linked to the GitHub repositories containing the source code associated with the publications [19]. For instance, this facilitates the detection of trends on the implementation level and monitoring of how the FAIR principles are followed by which people (e.g., considering who provides the source code to the public in a reproducible way).
- According to Tzitzikas et al. [20], the scholarly data of the MAKG can be used to measure institutions' research output.
- In [4], an approach for extracting scientific methods and data sets used by the authors is presented. The extracted methods and data sets are linked to the publications in the MAKG enabling novel scientific impact quantification tasks (e.g., measuring how often which data sets and methods have been reused by researchers) and the recommendation of methods and data sets. Overall, linking the key content of scientific publications as modeled in knowledge graphs or integrating such information into the MAKG can be considered as a natural extension of the MAKG in the future.
- The MAKG has inspired other researchers to use it in the context of data-driven history of science,[6] i.e., for science of science [30].
- Daquino et al. [31] present the OpenCitations data model and evaluate the representation of citation data in several knowledge graphs, such as the MAKG.

Benchmarking.

- As very large RDF knowledge graph, the MAKG has served as a data set for evaluating novel approaches to streaming partitioning of RDF graphs [21].

2.3. *Current Limitations*

Based on the statistical analysis of the MAKG and the analysis of the usage scenarios of the MAKG so far, we identified the following shortcomings:

[5] See https://www.nesta.org.uk and https://github.com/michaelfaerber/MAG2RDF/issues/1.

[6] https://www.downes.ca/post/69870.

– Author name disambiguation is apparently one of the most pressing needs for enhancing the MAKG.
– The assigned fields of study associated with the papers in the MAKG are not accurate (e.g., architecture), and the field of study hierarchy is quite erroneous.
– The use cases of the MAKG show that the MAKG has not been used extensively for machine learning tasks. So far, only entity embeddings for the MAKG as of 2019 concerning the entity type *paper* are available, and these have not been evaluated. Thus, we perceive a need to provide state-of-the-art embeddings for the MAKG covering many instance types, such as papers, authors, journals, and conferences.

3. Author Name Disambiguation

3.1. Motivation

The MAKG is a highly comprehensive data set containing more than 243 million author entities alone. As is the case with any large database, duplicate entries cannot be easily avoided [32]. When adding a new publication to the database, the maintainers must determine whether the authors of the new paper already exist within the database or if a new author entity is to be created. This process is highly susceptible to errors, as certain names are common. Given a large enough sample size, it is not rare to find multiple people with identical surnames and given names. Thus, a plain string-matching algorithm is not sufficient for detecting duplicate authors. Table 3 showcases the ten most frequently occurring author names in the MAKG[7] to further emphasize the issue. All author names are of Asian origin. Whilst it is true that romanized Asian names are especially susceptible to cause duplicate entries within a database [33], the problem is not limited to any geographical or cultural origin and is, in fact, a common problem shared by Western names as well [34].

The goal of the author-name disambiguation task is to identify the maximum number of duplicate authors, while minimizing the number of "false positives;" that is, it aims to limit the number of authors classified as duplicates even though they are distinct persons in the real world.

Table 3
Most frequently occurring author names in the MAKG

Author Name	Frequency
Wang Wei	20,235
Zhang Wei	19,944
Li Li	19,049
Wang Jun	16,598
Li Jun	15,975
Li Wei	15,474
Wei Wang	14,020
Liu Wei	13,578
Zhang Jun	13,553
Wei Zhang	13,366

In Sec. 3.2, we dive into the existing literature concerning author name disambiguation and, more generally, entity resolution. In Sec. 3.3, we define our problem formally. In Sec. 3.4, we introduce our approach, and we present our evaluation in Sec. 3.5. Finally, we conclude with a discussion of our results and lessons learned in Sec. 3.6.

3.2. Related Work

Entity Resolution. Entity resolution is the task of identifying and removing duplicate entries in a data set that refer to the same real-world entity. This problem persists across many domains and, ironically, is itself affected by duplicate names: "object identification" in computer vision, "coreference resolution" in natural language processing, "database merging," "merge/purge processing," "deduplication," "data alignment" or "entity matching" in the database domain, and "entity resolution" in the machine learning domain [55]. The entities to be resolved are either part of the same data set or may reside in multiple data sources.

Newcombe et al. were the first ones to define the entity linking problem [56], which was later modeled mathematically by Fellegi and Sunter [57]. They derived a set of formulas to determine the probabilities of two entities being "matching" based on given preconditions (i.e., similarities between feature pairs). Later studies refer to the probabilistic formulas as equivalent to a naïve Bayes classifier [58, 59].

Generally speaking, there exists two approaches to dealing with entity resolution [60]. In statistics and machine learning, the task is formulated as a clas-

[7] We used the December 2019 version of the MAKG for this analysis.

Table 4

Approaches to author name disambiguation in the last 10 years (2011–2021)

Authors	**Year**	**Approach**	**Supervised**
[35]	2020	Graph-based combination of author similarity and topic graph	✗
[36]	2020	Adversarial representation learning	✓
[37]	2019	Matching e-mail address, self-citation and co-authorship with iterative clustering	✗
[38]	2019	Hierarchical clustering with edit distances	✗
[39]	2019	Graph-based approach	✗
[40]	2019	Deep neural network	✓
[41]	2019	Graph-based approach and clustering	✗
[38]	2019	Molecular cross clustering	✗
[42]	2018	Combination of single features	✓
[43]	2018	Rule-based clustering	✗
[34]	2017	Multi-level clustering	✗
[44]	2017	Hierarchical clustering with combination of similarity metrics	✗
[45]	2017	Neural network using embeddings	✓
[46]	2016	DBSCAN with random forest	✗
[47]	2016	Clustering based on co-authorship	✗
[48]	2016	Rule-based heuristic, linear regression, support vector machines and AdaBoost	✓
[49]	2015	Support vector machines	✓
[50]	2014	Deep neural network	✓
[51]	2014	Rule-based scoring	✗
[52]	2014	Pairwise comparison and clustering	✗
[53]	2013	Random forest, support vector machines and clustering	✓
[54]	2011	Single layer perceptron	✓

sification problem, in which all pairs of entries are compared to each other and classified as matching or non-matching by an existing classifier. In the database community, a rule-based approach is usually used to solve the task. Rule-based approaches can often be transformed into probabilistic classifiers, such as naïve Bayes, and require certain previous domain knowledge for its setup.

Author name disambiguation. Author name disambiguation is a subcategory of entity resolution and is performed on collections of authors. Table 4 provides an overview of papers specifically approaching the task of author name disambiguation in the scholarly field in the last decade.

Ferreira et al. surveyed existing methods for author name disambiguation [61]. They categorized existing methods by their types of approach, such as author grouping or author assignment methods, as well as their clustering features, such as citation information, web information, or implicit evidence.

Caron and van Eck applied a strict set of rules for scoring author similarities, such as 100 points for identical e-mail addresses [51]. Author pairs scoring above a certain threshold are classified as identical. Although the creation of such a rule set requires specific domain knowledge, the approach is still very simplistic in nature compared to other supervised learning approaches. In addition, it outperforms other clustering-based unsupervised approaches significantly [62]. For these reasons, we base our approach on the one presented in their paper.

3.3. Problem Formulation

Existing papers usually aim to introduce a new fundamental approach to author name disambiguation and do not focus mainly on the general applicability of their approaches. As a result, these approaches are often impractical when applied to a large data set. For example, some clustering-based approaches require the prior knowledge of the number of clusters [34] and other approaches require the pairwise comparison of all entities [49], whereas some require external information gathered through web queries [43], which cannot be feasibly done when dealing with millions of entries, as the inherent bottleneck of web requests greatly limits the speed of the overall processes. Therefore, instead of choosing a single approach, we aim to select

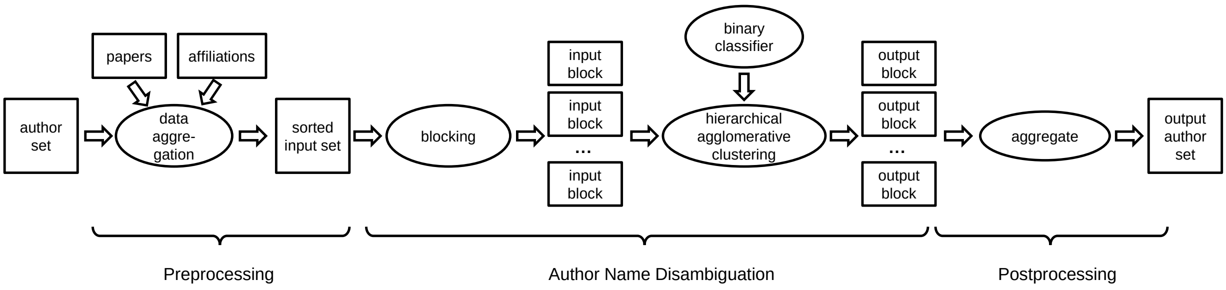

Figure 6. Author name disambiguation process.

features from different models and combine them to fit to our target data set containing millions of author names.

We favor the use of unsupervised learning for the reasons mentioned above: lack of training data, lack of need for maintaining and updating of training data, and generally more favorable time and space complexity. Thus, in our approach, we chose the *hierarchical agglomerative clustering algorithm* (HAC). We formulate the problem as follows:

Given a set of n authors $A = \{a_1, a_2, a_3, \ldots, a_n\}$ where a_i represents an individual entry in the data set. Furthermore, each individual author a_i consists of k features (i.e., $a_i = \{a_{i1}, a_{i2}, a_{i3}, \ldots, a_{ik}\}$). a_{ik} is the *k-th* feature of the *i-th* author. The goal of our approach is to eliminate duplicate entries in the data set that describe the same real-world entity, in this case the same person. To this end, we introduce a matching function f which determines whether two given input entities are "matching," i.e., describe the same real-world person, or "non-matching," i.e., describe two distinct people. Given an input of two authors a_i and a_j, the function returns the following:

$$f(a_i, a_j) = \begin{cases} 1 & \text{if } a_i \text{ and } a_j \text{ refer to the } \textit{same} \text{ real world entity, i.e., are ``matching''} \\ 0 & \text{if } a_i \text{ and } a_j \text{ refer to } \textit{different} \text{ real world entities, i.e., are ``non-matching''} \end{cases}$$

The goal of our entity resolution task is therefore to reduce the given set of authors A into a subset $\tilde{A}$ where $\forall a_i, a_j \in \tilde{A}, f\left(a_i, a_j\right) = 0$.

3.4. Approach

We follow established procedures from existing research for unsupervised author name disambiguation [51, 61] and utilize a two-part approach consisting of pairwise similarity measurement using author and paper metadata and clustering. Additionally, we use blocking (see Sec. 3.4.3) to reduce the complexity considerably. Figure 6 shows the entire system used for the author name disambiguation process. The system's steps are as follows:

1. **Preprocessing.** We preprocess the data by aggregating all relevant information (e.g., concerning authors, publications, and venues) into one single file for easier access. We then sort our data by author name for the final input.
2. **Disambiguation.** We apply blocking to significantly reduce the complexity of the task. We then use hierarchical agglomerative clustering with a rule-based binary classifier as our distance function to group authors into distinct disambiguated clusters.
3. **Postprocessing.** We aggregate the output clusters into our final disambiguated author set.

Below, the most important aspects of these steps are outlined in more detail.

3.4.1. Feature Selection

We use both author- and publication metadata for disambiguation. We choose the features based on their availability in the MAKG and on their previous use in similar works from Table 4. Overall, we use the following features:

– **Author name**: This is not used explicitly for disambiguation, but rather as a feature for blocking to reduce the complexity of the overall algorithm.
– **Affiliation**: This determines whether two authors share a common affiliation.
– **Co-authors**: This determines whether two authors share common co-authors.
– **Titles**: This calculates the most frequently used keywords in each author's published titles in order to determine common occurrences.

- **Years**: This compares the time frame in which authors published works.
- **Journals** and **conferences**: These compare the journals and conferences where each author published.
- **References**: This determines whether two authors share common referenced publications.

Although e-mail has proven to be a highly effective distinguishing feature for author name disambiguation [63][52][51], this information is not available to us directly and therefore omitted from our setup. Co-authorship, on the other hand, is one of the most important features for author name disambiguation [64]. Affiliation could be an important feature, though we could not rely solely on it as researchers often change their place of work. In addition, as the affiliation information is automatically extracted from the publications, it might be on varying levels (e.g., department vs. university) and written in different ways (e.g., full name vs. abbreviation). Journals and conferences could be effective features, as many researchers tend to publish in places familiar to them. For a similar reason, references can be an effective measure as well.

3.4.2. Binary Classifier

We adapt a rule-based binary classifier as seen in the work of Caron and van Eck [51]. We choose a simple rule-based classifier because of its simplicity, interpretability, and scalability. The unsupervised approach does not require any training data and is therefore well suited for our situation. Furthermore, it is easily adapted and fine-tuned to achieve the best performance based on our data set. Its lack of necessary training time, as well as fast run time, makes it ideal when working with large-scale data sets containing millions of authors.

The binary classifier uses as input two feature vectors representing two author entities. Given two authors a_i, a_j, each consisting of n features $a_i = \{a_{i1}, a_{i2}, a_{i3}, \ldots, a_{in}\}$, the similarity $sim\left(a_i, a_j\right)$ between these two authors is the sum of similarities between each of their respective features where sim_n is the similarity between the *n-th* feature of two authors.

$$sim\left(a_i, a_j\right) = \sum_{k=1}^{n} sim_k\left(a_{ik}, a_{jk}\right)$$

The classifier then compares the similarity $sim\left(a_i, a_j\right)$ with a predetermined threshold $\theta_{matching}$ in order to determine whether two authors are "matching" or "non-matching." Our classifier function takes the following shape:

$$f\left(a_i, a_j\right) = \begin{cases} 1, & \text{if } d\left(a_i, a_j\right) \geqslant \theta_{matching} \\ 0, & \text{if } d\left(a_i, a_j\right) < \theta_{matching} \end{cases}$$

For each feature, the similarity function consists of rule-based scoring. Below, we briefly describe how similarities between each individual feature are calculated.

1. For features with *one individual value*, as is the case with *affiliations* for the MAG because it does not record historical data, the classifier determines whether both entries match and assigns a fixed score $s_{affiliation}$.

$$sim_{affiliation}\left(a_i, a_j\right) = \begin{cases} s_{affiliation} & \text{if } a_{i,affiliation} \\ & \quad = a_{j,affiliation} \\ 0 & \text{else} \end{cases}$$

2. For other features consisting of *multiple values* such as *co-authors*, the classifier determines the intersection of both value sets. Here, we assign scores using a stepping function, i.e., fixed scores for an intersection of one, two, three, etc.
The following formula represents the similarity function for calculating similarities between two authors for the feature *co-authors*, though the same formula holds for features *journals*, *conferences*, *titles*, and *references* with their respective values.

$$sim_{co-authors}\left(a_i, a_j\right) = \begin{cases} s_{co-authors1} & \text{if } a_{i,co-authors} \\ & \quad \cap\, a_{j,co-authors} = 1 \\ s_{co-authors2} & \text{if } a_{i,co-authors} \\ & \quad \cap\, a_{j,co-authors} = 2 \\ s_{co-authors3} & \text{if } a_{i,co-authors} \\ & \quad \cap\, a_{j,co-authors} \geqslant 3 \\ 0 & \text{else} \end{cases}$$

Papers' titles are a special case for scoring, as they must be numericalized to allow a comparison. Ideally, we would use a form of word embeddings to measure the true semantic similarity between two titles, but, based on the results of preliminary experiments, we did not find it worth, as the added computation necessary would be significant and would most likely not translate directly into huge performance increases. We therefore adapt a plain surface form string comparison. Specifically, we extract the top ten most frequently used words from the tokenized and lemmatized titles of works published by an author and calculate their intersection with the set of another author.

3. A special case exists for the *references* feature. A bonus score $s_{self-reference}$ is applied to the case of self-referencing, that is if two compared authors directly reference each other in their respective works, as can be seen in the work of Caron and van Eck [51].

– For some features, such as *journals* and *conferences*, a large intersection between two authors may be uncommon. We only assign a non-zero value if both items share a common value.

$$sim_{journals}\left(a_i, a_j\right) = \begin{cases} s_{journals}, & \text{if } a_{i,journals} \\ & \quad \cap\, a_{j,journals} \geqslant 1 \\ 0, & \text{else} \end{cases}$$

5. Other features such as *publication year* also consist of multiple values, though we interpret them as extremes of a time span. Based on their feature values, we construct a time span for each author in which they were active and check for overlap in active years when comparing two authors (similar to [49]). Again, a fixed score is assigned based on the binary decision. For example, if author A published papers in 2002, 2005, and 2009, we extrapolate the active research period for author A as 2002–2009. If another author B was active during the same time period or within 10 years of both ends of the time span, i.e., 1992–2019, we assign a score s_{years} as the output. We expect most author comparisons to share an overlap in research time span and thus receive a score of greater than zero. Therefore, this feature is more aimed at "punishing" obvious non-matches. The scoring function takes the following shape:

$$sim_{years}\left(a_i, a_j\right) = \begin{cases} s_{years} & \text{if } a_i \text{ and } a_j \text{ were active} \\ & \text{within 10 years of one} \\ & \text{another} \\ 0 & \text{else} \end{cases}$$

3.4.3. Blocking

Due to the high complexity of traditional clustering algorithms (e.g., $O\left(n^2\right)$), there is a need to implement a blocking mechanism to improve scalability of the algorithm to accommodate large amounts of input data. We implement sorted neighborhood [65] as a blocking mechanism. We sort authors based on their names as provided to us by the MAKG and measure the similarity using the Jaro-Winkler distance [66], as Winkler [67] provides good performances for name-matching tasks on top of being a fast heuristic [68].

The Jaro-Winkler similarity returns values between 0 and 1, where a greater value signifies a closer match. We choose 0.95 as the threshold $\theta_{blocking}$, based on performance on our evaluation data set, and we choose 0.1 as the standard value for the scaling factor p. Similar names will be formed into blocks where we perform pair-wise comparison and cluster authors which were classified as similar by our binary classifier.

3.4.4. Clustering

The final step of our author name disambiguation approach consists of clustering the authors. To this end, we choose the traditional hierarchical agglomerative clustering approach. We generate all possible pairs between authors for each block and apply our binary classifier to distinguish matching and non-matching entities. We then aggregate the resulting disambiguated blocks and receive the final collection of unique authors as output.

3.5. Evaluation

3.5.1. Evaluation Data

The MAKG contains bibliographical data on scientific publications, researchers, organizations, and their relationships. We use the version published in December 2019 for evaluation, though our final published results were performed on an updated version (with only minor changes) from June 2020 consisiting of 243,042,675 authors.

3.5.2. Evaluation Setup

For the evaluation, we use the ORCID iD, a persistent digital identifier for researchers, as a ground truth, following [69]. ORCID iDs have been established as a common way to identify researchers. Although the ORCID iD is still in the process of being adopted, it is already widely used. More than 7,000 journals already collect ORCID iDs from authors.[8] Our ORCID evaluation set consists of 69,742 author entities.

Although using ORCID as a ground truth, we are aware that this data set may be characterized by imbalanced metadata. First of all, ORCID became widely adopted only a few years ago. Thus, primarily author

[8] https://info.orcid.org/requiring-orcid-in-publications/

Table 5
Hyperparameter values for high precision setup

Hyperparameter	**Value**
$s_{affiliation}$	1
$s_{co-authors1}$	3
$s_{co-authors2}$	5
$s_{co-authors3}$	8
$s_{titles1}$	3
$s_{titles2}$	5
$s_{titles3}$	8
$s_{journals}$	3
$s_{conferences}$	3
s_{years}	3
$s_{references1}$	2
$s_{references2}$	3
$s_{references3}$	5
$s_{self-references}$	8
$\theta_{matching}$	10
$\theta_{blocking}$	0.95
p	0.1

Table 6
Diffusion matrix of high precision setup

	Positive Label	**Negative Label**	**Total**
Positive Classification	37	2	39
Negative Classification	12	1424	1436
Total	49	1426	1475

Table 7
Average disambiguation score per feature for high precision setup (TP=True Positive; TN=True Negative; False Positive; False Negative)

	TP	**TN**	**FP**	**FN**
score_affiliation	0.0	0.004	0.0	0.083
score_coauthors	0.0	0.0	0.0	0.0
score_titles	0.162	0.0	0.0	0.25
score_years	3.0	2.89	3.0	3.0
score_journals	3.0	0.034	3.0	1.75
score_conferences	3.0	2.823	3.0	3.0
score_self_reference	0.0	0.0	0.0	0.0
score_references	2.027	0.023	2.0	0.167

names from publications published in recent years are considered in our evaluation. Furthermore, we can assume that ORCID is more likely to be used by active researchers with a comparatively higher number of publications and that the more publications' metadata we have available for one author, the higher the probability is for a correct author name disambiguation.

We set the parameters as given in Table 5. We refer to these as the *high precision configuration*. These values were chosen based on choices in other similar approaches [51] and adjusted through experimentations with our evaluation data as well as analysis of the relevancy of each individual feature (see Sec. 3.5.3).

We rely on the traditional metrics of *precision*, *recall* and *accuracy* for our evaluation.

3.5.3. Evaluation Results

Due to blocking, the total number of pairwise comparisons was reduced from 2,431,938,411 to 1,475. Out of them, 49 pairs were *positive* according to our ORCID labels; i.e., they refer to the same real-world person; the other 1,426 were *negative*. Full classification results can be found in Table 6. We have a heavily imbalanced evaluation set, with a majority of pairings being negative. Nevertheless, we were able to correctly classify the majority of negative labels (1,424 out of 1,426). The great number of false negative classifications is immediately noticeable. This is due to the selection of features or lack of distinguishing features overall to classify certain difficult pairings. We have therefore chosen to opt for a high percentage of false negatives to minimize the amount of false positive classifications, as those are tremendously more damaging to an author disambiguation result.

Table 7 showcases the average scores for each feature separated into each possible category of outcome. For example, the average score for the feature *titles* from all comparisons falling under the true positive class was 0.162, and the average score for the feature *years* for comparisons from the true negative class was 2.899. Based on these results, *journals* and *references* play a significant role in identifying duplicate author entities within the MAKG; that is, they contribute high scores for true positives and true negatives. Every single author pair from the true positive classification cluster shared a common journal value, whereas almost none from the true negative class did. Similar observations can be made for the feature *references* as well.

Our current setup results in a precision of **0.949**, recall of **0.755** and an accuracy of **0.991**.

By varying the scores assigned by each feature level distance function, we can affect the focus of the entire system from achieving a high level of precision to a high level of recall.

Table 8

Updated disambiguation scores for high recall setup

	High Precision	High Recall
$s_{affiliation}$	**1**	**5**
$s_{co-authors1}$	3	3
$s_{co-authors2}$	5	5
$s_{co-authors3}$	8	8
$s_{titles1}$	3	3
$s_{titles2}$	5	5
$s_{titles3}$	8	8
s_{years}	3	3
$s_{journals}$	**3**	**4**
$s_{conferences}$	**3**	**4**
$s_{self-references}$	8	8
$s_{references1}$	2	2
$s_{references2}$	3	3
$s_{references3}$	5	5

To improve our relatively poor recall value, we have experimented with different setups for distance scores. At high performance levels, a trade-off persists between precision and recall. By applying changes to score assignment as seen in Table 8, we arrive at the following results in Table 9.

We were able to increase the recall from 0.755 to **0.918**. At the same time, our precision plummeted from the original 0.949 to **0.776**. As a result, the accuracy stayed at a similar level of **0.988**. The exact diffusion matrix can be found in Table 9. With our new setup, we were able to identify the majority of all duplicates (45 out of 49), though at the cost of a significant increase in the number of false positives (from 2 to 13). By further analyzing the exact reasoning behind each type of classification through analysis of individual feature scores in Table 10, we found that the true positive and false positive classifications from the same feature similarities, therefore creating a theoretical upper limit to the performance of our specific approach and data set. We hypothesize that additional external data may be necessary to exceed this upper limit of performance.

We must consider the heavily imbalanced nature of our classification labels when evaluating the results in order to avoid falling into the trap of the "high accuracy paradox." That is the resulting high accuracy score of a model on highly imbalanced data sets, where negative labels significantly outnumber positive labels. The model's favorable ability to predict the true negatives outweigh its shortcomings for identifying the few positive labels.

Table 9

Diffusion matrix for high recall setup

	Positive Label	Negative Label	Total
Positive Classification	45	13	58
Negative Classification	4	1413	1417
Total	49	1426	1475

Table 10

Average disambiguation score per feature for the high recall setup (TP=True Positive; TN=True Negative; FP = False Positive; FN = False Negative). As we consider the scores for disambiguation and not the confusion matrix for the classification, values can be greater than one

	TP	TN	FP	FN
score_affiliation	0.111	0.004	1.538	0.0
score_coauthors	0.0	0.0	0.0	0.0
score_titles	0.133	0.0	0.0	0.75
score_years	3.0	2.89	3.0	3.0
score_journals	3.911	0.023	3.077	0.0
score_conferences	4.0	3.762	4.0	4.0
score_self_reference	0.0	0.0	0.0	0.0
score_references	1.667	0.023	0.308	0.5

Ultimately, we decided to use the high-precision setup to create the final knowledge graph, as precision is a much more meaningful metric for author name disambiguation as opposed to recall. It is often preferable to avoid removing non-duplicate entities rather than identifying all duplicates at the cost of false positives.

We also analyzed the average feature density per author in the MAKG and the ORCID evaluation data set to gain deeper insight into the validity of our results. Feature density here refers to the average number of data entries within an individual feature, such as the number of papers for the feature "published papers." The results can be found in Table 11.

As we can observe, there is a variation in "feature richness" between the evaluation set and the overall data set. However, for the most important features used for disambiguation—namely *journals*, *conferences* and *references*—the difference is not as pronounced. Therefore, we can assume that the disambiguation results will not be strongly affected by this variation.

Performing our author name disambiguation approach to the whole MAKG containing *243,042,675 authors* (MAKG version from June 2020) resulted in a reduced set of *151,355,324 authors*. This is a reduc-

Table 11
Comparison between the overall MAG data set and the evaluation set

	MAG	Evaluation
AuthorID	1.0	1.0
Rank	1.0	1.0
NormalizedName	1.0	1.0
DisplayName	1.003	1.0
LastKnownAffiliationID	**0.172**	**0.530**
PaperCount	1.0	1.0
CitationCount	1.0	1.0
CreateDate	1.0	1.0
PaperID	**2.612**	**1.196**
DOI	1.240	1.0
Coauthors	**11.187**	**4.992**
Titles	**2.620**	**1.198**
Year	1.528	1.107
Journal	0.698	0.819
Conference	0.041	0.025
References	20.530	26.590
ORCID	0.0003	1.0

tion by 37.7% and shows that applying author name disambiguation is highly beneficial.

Importantly, we introduced a maximum block size of 500 in our final approach. Without it, the number of authors grouped into the same block would theoretically be unlimited. The introduction of a limit to block size further improves performance significantly, reducing the run-time from over a week down to about 48 hours.[9] We have therefore opted to keep the limit, as the tradeoff in performance decrease is manageable and as we aimed to provide an approach for real application rather than a proof of concept. However, the limit can be easily removed or adjusted.

3.6. Discussion

Due to the high number of authors with identical names within the MAG, our blocking algorithm sometimes still generates large blocks with more than 20,000 authors. The number of pairwise classifications necessary equates to the number of combinations, namely $\binom{n}{2}$, leading to high computational complexity for larger block sizes. One way of dealing with this issue would be to manually limit the maximum number of entities within one block, as we have done. Doing so will split potential duplicate entities into distinct blocks, meaning they will never be subject to comparison by the binary classifier, although the entire process may be sped up significantly depending on the exact size limit selected. To highlight the challenge, Table 12 showcases the author names with the largest block sizes created by our blocking algorithm, i.e., author names generating the most complexity. The difference in total comparisons for the name block of "Wang Wei" would be 204,717,495 comparisons[10] with no block size limit, compared to 5,017,495 comparisons[11] for a block limit of 500 authors. We have found the difference in performance to be negligible compared to the total amount of duplicate authors found, as it differs by less than 2 million authors compared to the almost 100 million duplicate authors found.

Table 12
Largest author name blocks during disambiguation

Author Name	Block size
Wang Wei	20,235
Zhang Wei	19,944
Li Li	19,049
Wang Jun	16,598
Li Jun	15,975
Li Wei	15,474
Wei Wang	14,020
Liu Wei	13,580
Zhang Jun	13,553
Wei Zhang	13,366

Our approach can be further optimized through hand-crafted rules for dealing with certain author names. Names of certain origins such as Chinese or Korean names possess certain nuances. While the alphabetized Romanized forms of two Chinese names may be similar or identical, the original language text often shows a distinct difference. Furthermore, understanding the composition of surnames and given names in this case may also help further reduce the complexity. As an example, the names "Zhang Lei" and "Zhang Wei" only differ by one single character in their Romanized forms and would be classified as potential duplicates or typos due to their similarity, even though

[9] Using an Intel Xeon E5-2660 v4 processor and 128 GB of RAM

[10] Total comparisons for 20,235 authors with no block size limit: $\binom{20,235}{2} = 204,717,495$

[11] Total comparisons for 20,235 authors with a block size limit of 500: $40 * \binom{500}{2} + \binom{235}{2} = 5,017,495$

Table 13
Overview of MAG field of study hierarchy

Level	# of fields of study
0	19
1	292
2	138,192
3	208,368
4	135,913
5	167,676

for native Chinese speakers, such names signify two distinctly separate names, especially when written in the original Chinese character form. Chinese research publications are rising in number in the past years [8]. Given their susceptibility to creating duplicate entries as well as their significant presence in the MAKG already, future researchers might be well suited to isolate this problem as a focal point.

Additionally, there is the possibility to apply multiple classifiers and combine their results in a hybrid approach. If we were able to generate training data of sufficient volume and quality, we would be able to apply certain supervised learning approaches such as neural networks or support vector machines using our generate feature vectors as input.

4. Field of Study Classification

4.1. Motivation

Publications modeled in the MAKG are assigned to specific fields of study. Additionally, the fields of study are organized in a hierarchy. In the MAKG as of June 2020, 709,940 fields of study are organized in a multilevel hierarchical system (see Table 13). Both the field of study paper assignments and the field of study hierarchy in the MAKG originate from the MAG data provided by Microsoft Research. The entire classification scheme is highly comprehensive and covers a huge variety of research areas, but the labeling of papers contains many shortcomings. Thus, the second task in this article for improving the MAKG is the revision of field of study assignment of individual papers.

Many of the higher-level fields of study in the hierarchical system are highly specific, and therefore lead to many misclassifications purely based on certain matching keywords in the paper's textual information. For instance, papers on the topic of machine learning architecture are sometimes classified as "Architecture." Since the MAG does not contain any full texts of papers, but is limited to the titles and abstracts only, we do not believe that the information provided in the MAG is comprehensive enough for effective classification on such a sophisticated level.

On top of that, an organized structure is highly rigid and difficult to change. When introducing a previously unincorporated field of study, we have to not only modify the entire classification scheme, but ideally also relabel all papers in case some fall under the new label.

We believe the underlying problem to be the complexity of the entire classification scheme. We aim to create a simpler structure that is extendable. Our idea is not aimed at replacing the existing structure and field of study labels, but rather enhancing and extending the current system. Instead of limiting each paper to being part of a comprehensive structured system, we (1) merely assign a single field of study label at the top level (also called "discipline" in the following, level 0 in the MAKG), such as computer science, physics, or mathematics. We then (2) assign to each publication a list of keywords (i.e., tags), which are used to describe the publication in further detail. Our system is therefore essentially descriptive in nature rather than restrictive.

Compared to the classification scheme of the original MAKG and the MAG so far, our proposed system is more fluid and extendable since its labels or tags are not constrained to a rigid hierarchy. New concepts are freely introduced without affecting existing labels.

Our idea therefore is to classify papers on a basic level, then extract keywords in the form of tags for each paper. These can be used to describe the content of a specific work, while leaving the structuring of concepts to domain experts in each field. We classify papers into their respective fields of study using a transformer-based classifier and generate tags for papers using keyword extraction from the publications' abstracts.

In Sec. 4.2, we introduce related work concerning text classification and tagging. We describe our approach in Sec. 4.3 and our evaluation of the field of study classification in Sec. 4.4. In Sec. 4.4.5, we describe our approach for extracting paper keywords for publication tagging. Finally, we conclude with Sec. 4.5.

4.2. Related Work

Text classification. The tagging of papers based on their abstracts can be regarded as a text classification task. Text classification aims to categorize given texts into distinct subgroups according to predefined characteristics. As with any classification task, text classification can be separated into binary, multi-label and multi-class classification.

Kowsari et al. [70] provide a recent survey of text classification approaches. Traditional approaches include techniques such as the Rocchio algorithm [71], boosting [72] and bagging [73], and logistic regression [74], as well as naïve Bayes. Clustering-based approaches include k-nearest neighbor and support vector machines [75]. More recent approaches mostly utilize deep learning. Recurrent neural networks [76] and long short-term memory networks (LSTMs) [77] had been the predominant approaches for representing language and solving language-related tasks until the rise of transformer-based models.

Transformer-based models can be generally separated into autoregressive and autoencoding models. Autoregressive models such as Transformer-XL [78] learn representations for individual word tokens sequentially, whereas autoencoding models such as BERT [79] are able to learn representations in parallel using the entirety of the document, even words found after the word token. Newer autoregressive models such as XLNet [80] combine features from both categories and are able to achieve state-of-the-art performance. Additionally, other variants of the BERT model exist, such as ALBERT [81] and RoBERTa [82]. Furthermore, specialized BERT variants have been created. One such variant is SciBERT [83], which specializes in academic texts.

Tagging. Tagging—based on extracting the tags from a text—can be considered synonymous with keyword extraction. To extract keywords from publications' full texts, several approaches and challenges have been proposed [84–86], exploiting publications' structures, such as citation networks [87]. In our scenario, we use publications' abstracts, as the full texts are not available in the MAKG. Furthermore, we focus on keyphrase extraction methods requiring no additional background information and not designed for specific tasks, such as text summarization.

TextRank [88] is a graph-based ranking model for text processing. It performs well for tasks such as keyword extraction as it does not rely on local context to determine the importance of a word, but rather uses the entire context through a graph. For every input text, the algorithm splits the input into fundamental units (words or phrases depending on the task) and structures them into a graph. Afterward, an algorithm similar to PageRank determines the relevance of each word or phrase in order to extract the most important ones.

Another popular algorithm for keyword extraction is RAKE, which stands for rapid automatic keyword extraction [89]. In RAKE, the text is split by a previously defined list of keywords. Thus, a less comprehensive list would lead to longer phrases. In contrast, TextRank splits the text into individual words first and combines words which benefit from each other's context at a later stage in the algorithm. Overall, RAKE is more suitable for text summarization tasks due to its longer extracted key phrases, whereas TextRank is suitable for extracting shorter keywords used for tagging, in line with our task. In their original publication, the authors of TextRank applied their algorithm for keyword extraction from publications' abstracts. Due to all these reasons, we use TextRank for publication tagging.

4.3. Approach

Our approach is to fine-tune a state-of-the-art transformer model for the task of text classification. We use the given publications' abstracts as input in order to classify each paper into one of 19 top-level field of study labels (i.e., level 0) predefined by the MAG (see Table 11). After that, we apply TextRank to extract keyphrases and assign them to papers.

4.4. Evaluation

4.4.1. Evaluation Data

For the evaluation, we produce three labeled data sets in an automatic fashion. Two of the data sets are used to evaluate the *current* field of study labels in the MAKG (and MAG) and the given *MAKG field of study hierarchy*, while the last data set acts as our source for training and evaluating our approach for the field of study classification.

In the following, we describe our approaches for generating our three data sets.

1. For our first data set, we select field of study labels directly from the MAKG. As mentioned previously, the MAKG's fields of study are provided in a hierarchical structure, i.e., fields of study (e.g., research topics) can have several fields of study below them. We filter the field of study labels associated with papers for level-0 labels only, that is we consider only the 19 top-level labels and their assignments to papers. Table 14 lists all 19 level-0 fields of study in the MAG; these, as-

Table 14
List of level-0 fields of study from the MAG

MAG ID	Field of Study
41008148	Computer Science
86803240	Biology
17744445	Political Science
192562407	Materials Science
205649164	Geography
185592680	Chemistry
162324750	Economics
33923547	Mathematics
127313418	Geology
127413603	Engineering
121332964	Physics
144024400	Sociology
144133560	Business
71924100	Medicine
15744967	Psychology
142362112	Art
95457728	History
138885662	Philosophy
39432304	Environmental Science

sociated with the MAG papers, are also our 19 target labels for our classifier.

This data set will be representative of the field of study assignment quality of the MAKG overall as we compare its field of study labels with our ground truth (see Sec. 4.4.4).

2. For our second data set, we extrapolate field of study labels from the MAKG/MAG using the field of study hierarchy—that is, we relabel the papers using their associated top-level fields of study on level 0. For example, if a paper is currently labeled as "neural network," we identify its associated level-0 field of study (the top-level field of study in the MAKG). In this case, the paper would be assigned the field of study of "computer science."

   We prepare our data set by first replacing all field of study labels using their respective top-level fields of study. Each field of study assignment in the MAKG has a corresponding confidence score. We thus sort all labels by their corresponding level-0 fields of study and calculate the final field of study of a given paper by summarizing their individual scores. For example, consider a paper that originally has the field of study labels "neural network" with a confidence score of 0.6, "convolutional neural network" with a confidence score of 0.5 and "graph theory" with a confidence score of 0.8. The labels "neural network" and "convolutional neural network" are mapped back to the top-level field of study of "computer science", whereas "graph theory" is mapped back to "mathematics." In order to calculate the final score for each discipline, we totaled the weights of every occurrence of a given label. In our example, "computer science" would have a score of 0.5 + 0.6 = 1.1, and "mathematics" a score of 0.8, resulting in the paper being labeled as "computer science."

   This approach can be interpreted as an addition of weights on the direct labels we generated for our previous approach. By analyzing the differences in results from these two data sets, we aim to gather some insights into the validity of the hierarchical structure of the fields of study found in the MAG.

3. Our third data set is created by utilizing the papers' journal information. We first select a specific set of journals from the MAKG for which the journal papers' fields of study can easily be identified. This is achieved through simple string matching between the names of top-level fields of study and the names of journals. For instance, if the phrase "computer science" occurs in the name of a journal, we assume it publishes papers in the field of computer science.

   We expect the data generated by this approach to be highly accurate as the journal is an identifying factor of the field of study. We cannot rely on this approach to match all papers from the MAKG as a majority of papers were published in journals whose main disciplines could not be discerned directly from their names. Also, there exists a portion of papers that do not have any associated journal entries in the MAKG.

   We are able to label 2,553 journals in this fashion. We then label all 2,863,258 papers from these given journals using their journal-level field of study labels. We use the resulting data set to evaluate the fields of study in the MAKG as well as to generate training data for the classifier.

   In the latter case, we randomly selected 20,000 abstracts per field of study label, resulting in a total of 333,455 training samples (i.e., paper-field-of-study assignment pairs). The mismatch compared to the theoretical training data size of 380,000 comes from the fact that some labels had fewer than 20,000 papers available to select from.

Our data for evaluating the classifier comes from our third approach, namely the field of study assignment based on journal names. We randomly drew 2,000 samples for each label from the labeled set to form our test data set. Note that the test set does not overlap in any way with the training data set generated through the same approach, as both consist of distinctly separate samples (covering all scientific disciplines). In total, the evaluation set consists of 38,000 samples spread over the 19 disciplines.

4.4.2. *Evaluation Setup*

All our implementations use the Python module *Simple Transformers*[12] (based on *Transformers*[13]), which provides a ready-made implementation of transformer-based models for the task of multi-class classification. We set the number of output classes to 19, corresponding to the number of top-level fields of study we are trying to label. As mentioned in Section 4.4.1, we prepare our evaluation data set based on labels generated via journal names. We also prepare out training set from the same data set.

We choose the following model variants for each architecture:

1. *bert-large-uncased* for BERT,
2. *scibert_scivocab_uncased* for SciBERT,
3. *albert-base-v2* for ALBERT,
4. *roberta-large* for RoBERTa, and
5. *xlnet-large-cased* for XLNet.

All transformer models were trained on the bwUnicluster using GPU nodes containing 4 Nvidia Tesla V100 GPUs and an Intel Xeon Gold 6230 processor.

4.4.3. *Evaluation Metrics*

We evaluate our model performances using two specific metrics, the micro-F_1 score and Mathews correlation coefficient.

Table 15
Evaluation results of existing field of study labels

Label	**# labels**	**# matching**	**% matching**
Computer Science	21,157	15,056	71.163
Biology	212,356	132,203	62.255
Political Science	12,043	4,083	33.904
Materials Science	23,561	18,475	78.413
Geography	4,286	575	13.416
Chemistry	339,501	285,569	84.114
Economics	91,411	62,482	68.353
Mathematics	109,797	92,519	84.264
Geology	22600	18,377	81.314
Engineering	731,505	187,807	25.674
Physics	694,631	500,723	72.085
Sociology	10,725	9,245	86.200
Business	141,498	33,641	23.775
Medicine	311,197	186,194	59.832
Psychology	36,080	31,834	88.232
Art	23,728	4,336	18.274
History	39,938	5,161	12.923
Philosophy	19,517	6,363	32.602
Environm. Science	17,727	936	5.280
Total	2,863,258	1,595,579	55.726

The Matthews correlation coefficient (MCC), also known as the phi coefficient, is another standard metric used for multi-class classifications. It is often preferred for binary classification or multi-class classification with unevenly distributed class sizes. The MCC only achieves high values if all four classes of the diffusion matrix are classified accurately, and is therefore preferred for evaluating unbalanced data sets [90]. Even though our evaluation set is balanced, we nevertheless provide MCC as an alternative metric. The MCC is calculated as follows:

$$MCC = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}$$

with TP = true positives, FP = false positives, TN = true negatives, and FN = false negatives.

4.4.4. *Evaluation Results*

Evaluation of Existing Field of Study Labels

In the following, we outline our evaluation concerning the validity of the existing MAG field of study labels. We take our two labeled sets generated by our direct labeling (1st data set; 2,863,258 papers) as well as labeling through journal names (3rd data set) and compare the associated labels on level 0.

As we can see from the results in Table 15, the quality of top-level labels in the MAG can be improved. Out of the 2,863,258 papers, 1,595,579 matching labels were found, corresponding to a 55.73% match, meaning 55.73% of fields of study were labeled correctly according to our ground truth. Table 15 also showcases an in-depth view of the quality of labels for each discipline. We show the total number of papers for each field of study and the number of papers which are correctly classified according to our ground truth, followed by the percentage.

[12] https://github.com/ThilinaRajapakse/simpletransformers

[13] https://github.com/huggingface/transformers

Table 16
Evaluation results of the field of study hierarchy

Label	# labels	# matching	% matching
Computer Science	21,157	13,055	61.705
Biology	212,356	145,671	68.598
Political Science	12,043	8,035	66.719
Materials Science	23,561	13,618	57.799
Geography	4,286	285	6.650
Chemistry	339,501	239,576	70.567
Economics	91,411	62,025	67.853
Mathematics	109,797	79,959	72.824
Geology	22600	15,777	69.810
Engineering	731,505	207,063	28.306
Physics	694,631	464,083	66.810
Sociology	10,725	4,418	41.193
Business	141,498	26,095	18.442
Medicine	311,197	192,397	61.825
Psychology	36,080	25,548	70.809
Art	23,728	4,901	20.655
History	39,938	3,391	8.491
Philosophy	19,517	8,641	44.274
Environm. Science	17,727	302	1.704
Total	2,863,258	1,514,840	52.906

Evaluation of MAKG Field of Study Hierarchy

To determine the validity of the existing field of study hierarchy, we compare the indirectly labeled data set (2nd data set) with our ground truth based on journal names (3rd data set). The indirectly labeled data set is labeled using inferred information based on the overall MAKG field of study hierarchy (see Sec. 4.4.1). Here, we want to examine the effect the hierarchical structure would have on the truthfulness of field of study labels. The results can be found in Table 16.

Our result based on this approach is very similar to the previous evaluation. Out of the 2,863,258 papers, we found 1,514,840 labels matching those based on journal names, resulting in a 52.91% match (compared to 55.73% in the previous evaluation). Including the MAKG field of study hierarchy did not improve the quality of labels. For many disciplines, the number of mislabelings increased significantly, further devaluing the quality of existing MAG labels.

Evaluation of Classification

In the following, we evaluate the newly created field of study labels for papers determined by our transformer-based classifiers.

We first analyze the effect of training size on the overall results. Although we observe a steady increase in performance with each increase in size of our training set, the marginal increment deteriorates after a certain value. Therefore, with training time in mind, we decided to limit the training input size to 20,000 samples per label, leading a theoretical training data size of 390,000 samples. The number is slightly smaller in reality, however, due to certain labels having fewer than 20,000 training samples in total.

Table 17
Result comparison of various transformer-based classifiers

Model	MCC	F_1-Score
$BERT_{base}$	0.7452	0.7584
$BERT_{large}$	0.6853	0.7014
SciBERT	**0.7552**	**0.7678**
Albert	0.7037	0.7188
RoBERTa	0.7170	0.7316
XLNet	0.6755	0.6920

We then compared the performances of various transformer-based models for our task. Table 17 shows performances of our models trained on the same training set after one epoch. As we can see, SciBERT and $BERT_{base}$ outperform other models significantly, with SciBERT slightly edging out in comparison. Surprisingly, the larger BERT variant performs significantly worse than its smaller counterpart.

We then compare the effect of training epochs on performance. We limit our comparison to the SciBERT model in this case. We choose SciBERT as it achieves the best performance after one epoch of training. We fine-tune the same SciBERT model using an identical training set (20,000 samples per label) as well as the same evaluation set. We observe a peak in performance after two epochs (see Table 18). Although performance for certain individual labels keep improving steadily afterward, the overall performance starts to deteriorate. Therefore, training was stopped after two epochs for our final classifier. Note that we have perform similar analysis with some other models in a limited fashion as well. The best performance was generally achieved after two or three epochs depending on the model.

Table 19 showcases the performance per label for our SciBERT model after two training epochs on the evaluation set. On average, the classifier achieves an macro average F1 score of 0.78. In the detailed results for each label, we highlighted labels that achieved scores one standard deviation above and below the average.

Table 18
Comparison between various number of training epochs

# of epoch	MCC	F_1-Score
1	0.7552	0.7678
2	**0.7708**	**0.7826**
3	0.7665	0.7787
4	0.7615	0.7739
5	0.7558	0.7685

Table 19
Detailed evaluation results per label.

Label	Prec.	Recall	F_1	# samples
Computer Science	0.77	0.83	0.80	2,000
Biology	0.83	0.84	0.84	2,000
Political Science	0.83	0.81	0.82	2,000
Materials Science	0.78	0.83	0.80	2,000
Geography	**0.96**	**0.67**	0.79	2,000
Chemistry	0.79	0.80	0.80	2,000
Economics	**0.66**	**0.68**	**0.67**	2,000
Mathematics	0.79	0.81	0.80	2,000
Geology	**0.90**	**0.94**	**0.92**	2,000
Engineering	**0.58**	**0.49**	**0.53**	2,000
Physics	0.84	0.81	0.83	2,000
Sociology	0.81	0.70	0.75	2,000
Business	0.65	0.69	0.67	2,000
Medicine	0.84	0.84	0.84	2,000
Psychology	0.85	**0.89**	**0.87**	2,000
Art	**0.68**	0.76	0.72	2,000
History	0.70	0.75	0.72	2,000
Philosophy	0.81	0.81	0.81	2,000
Environm. Science	0.79	**0.86**	0.82	2,000
macro average	0.78	0.78	0.78	38,000

Classification performances for the majority of labels are similar to the overall average, though some outliers can be found.

Overall, the setup is especially adept at classifying papers from the fields of geology (0.94), psychology (0.87), medicine (0.84), and biology (0.84); whereas it performs the worst for engineering (0.53), economics (0.67), and business (0.67). The values in parentheses are the respective F_1-scores achieved during classification.

We suspect the performance differences to be a result of the breadth of vocabularies used in each discipline. Disciplines for which the classifier performs well usually use highly specific and technical vocabularies. Engineering especially follows this assumption, as engineering is an agglomeration of a multitude of disciplines, such as physics, chemistry, biology, and would encompass their respective vocabularies as well.

4.4.5. Keyword Extraction

As outlined in Sec. 4.3, we apply TextRank to extract keywords from text and assign them to publications. We use "pytextrank,"[14] a Python implementation of the TextRank algorithm as our keyword extractor. Due to the generally smaller text size of an abstract, we limit the number of keywords/key phrases to five. A greater number of keywords would inevitably introduce additional "filler phrases," which are not conducive for representing the content of a given abstract. Further statistics about the keywords are given in Sec. 6.

4.5. Discussion

In the following, we discuss certain challenges faced, lessons learned and future outlooks.

Our classification approach relied on the existing top-level fields of study (level 0) found in the MAKG. Instead, we could have established an entirely new selection of disciplines as our label set. It is also possible to adapt an establish classification scheme such as the ACM Computing Classification System[15] or the Computer Science Ontology[16] [91]. However, to the best of our knowledge, there is not an equivalent classification scheme covering the entirety of research topics found in the MAKG, which was a major factor leading us to adapt the field of study system.

On the side of keyword extraction, grouping of extracted keywords and key phrases and building a taxonomy or ontology are natural continuations of the work. We suggest categories to be constructed on an individual discipline level, rather than having a fixed category scheme for all possible fields of study. For instance, within the discipline of computer science, we could try to categorize tasks, data sets, approaches and so forth from the list of extracted keywords. Brack et al. [92] and Färber et. al. [4] recently published such an entity recognition approach. Both have also adapted the SciBERT architecture to extract scientific concepts from paper abstracts.

Future researchers can expand our extracted tags by enriching them with additional relationships to recre-

[14] https://github.com/DerwenAI/pytextrank/
[15] https://dl.acm.org/ccs
[16] https://cso.kmi.open.ac.uk/home

ate a similar structure to the current MAKG field of study hierarchy. Approaches such as the Scientific Information Extractor [93] could be applied to categorize or to establish relationships between keywords, building an ontology or rich knowledge graph.

5. Knowledge Graph Embeddings

5.1. Motivation

Embeddings provide an implicit knowledge representation for otherwise symbolic information. They are often used to represent concepts in a fixed low dimensional space. Traditionally, embeddings are used in the field of natural language processing to represent vocabularies, allowing computer models to capture the context of words and, thus, the contextual meaning.

Knowledge graph embeddings follow a similar principle, in which the vocabulary consists of entities and relation types. The final embedding encompasses the relationships between specific entities but also generalizes relations for entities of similar types. The embeddings retain the structure and relationships of information from the original knowledge graph and facilitate a series of tasks, such as knowledge graph completion, relation extraction, entity classification, question answering and entity resolution [94].

Färber [11] published pretrained embeddings for MAKG publications using RDF2Vec [95] as an "add-on" to the MAKG. Here, we provide an updated version of embeddings for a newer version of the MAG data set and for a variety of entity types instead of papers alone. We experiment with various types of embeddings and provide evaluation results for each approach. Finally, we provide embeddings for millions of papers and thousands of journals and conferences, as well as millions of disambiguated authors.

In the following, we introduce related work in Sec. 5.2. Sec. 5.3 describes our approach to knowledge graph embedding computation, followed by our evaluation in Sec. 5.4. We conclude in Sec. 5.6.

5.2. Related Work

Generally, knowledge graphs are described using triplets in the form of (h, r, t), referring to the head entity $h \in \mathbb{E}$, the relationship between both entities $r \in \mathbb{R}$, and the tail entity $t \in \mathbb{E}$. Nguyen [96] and Wang et al. [94] provide overviews of existing approaches for creating knowledge graph embeddings, as well as differences in complexity and performance.

Within existing literature, there have been numerous approaches to train embeddings for knowledge graphs. Generally speaking, the main difference between the approaches lies in the scoring function used to calculate the similarity or distance between two triplets. Overall, two major families of algorithms exist, ones using translational distance models and ones using semantic matching models.

Translational distance models use distance function scores to determine the plausibility of specific sets of triplets existing within a given knowledge graph context [94]. More specifically, the head entity of a triplet is projected as a point in a fixed dimensional space; the relationship entity is herein, for example, a directional vector originating from the head entity. The distance between the end point of the relationship entity and the tail entity in this given fixed dimensional space describes the accuracy or quality of the embeddings. One such example is the TransE [97] algorithm. The standard TransE model does not perform well on knowledge graphs with one-to-many, many-to-one, or many-to-many relationships [98] because the tail entities' embeddings are heavily influenced by the relations. Two tail entities that share the same head entity as well as relation are therefore similar in the embedding space created by TransE, even if they may be different concepts entirely in the real world. As an effort to overcome the deficits of TransE, TransH [98] was introduced to distinguish the subtleties of tail entities sharing a common head entity as well as relation. Later on, TransR was introduced to further model relations as separate vectors rather than hyperplanes, as is the case with TransH. The efficiency was later improved with the TransD [99] model.

Semantic matching models compare similarity scores in order to determine the plausibility of a given triplet. Here, relations are not modeled as vectors similar to entities, but rather as matrices describing interactions between entities. Such approaches include RESCAL [100], DistMult [101], HolE [102], ComplEx [23] and others.

More recent approaches use neural network architectures to represent relation embeddings. ConvE, for instance, represents head entity and relations as input and tail entity as output of a convolutional neural network [103]. ParamE extends the approach by representing relations as parameters of a neural network used to "translate" the input of head entity into the corresponding output of tail entity [104].

Table 20
Hyperparameters for training embeddings

Hyperparameter	Value
Embedding size	100
Max training step	1,000,000
Batch size	1,000
Negative sampling size	1,000

In addition, there are newer variations of knowledge graph embeddings, for example using textual information [105] and literals [106, 107]. Overall, we decided to use established methods to generate our embeddings for stability in results, performance during training and compatibility with file formats and graph structure.

5.3. Approach

We experiment with various embedding types and compare their performances on our data set. We include both translational distance models and semantic matching models of the following types: TransE [97], TransR [108], DistMult [101], ComplEx [23], and RESCAL [102] (see Sec. 5.2 for an overview how these approaches differ from each other). The reasoning behind the choices is as follows: the embedding types need to be state-of-the-art and widespread, therein acting as the basis of comparison. In addition, there needs to be an efficient implementation to train each embedding type, as runtime is a limiting factor. For example, the paper embeddings by Färber [11] were trained using RDF2Vec [95] and took two weeks to complete. RDF2Vec did not scale well enough using all authors and other entities in the MAKG.[17]

5.4. Evaluation

5.4.1. Evaluation Data

Our aim is to generate knowledge graph embeddings for the entities of type papers, journals, conferences, and authors to solve machine learning-based tasks, such as search and recommendation tasks. The RDF representations can be downloaded from the MAKG website.[18]

We first select the required data files containing the entities of our chosen entity types and combine them into a single input. Ideally, we would train paper and author embeddings simultaneously, such that they benefit from each other's context. However, the required memory space proved to be a limiting factor given the more than 200 million authors and more than 200 million papers. Ultimately, we train embeddings for papers, journals, and conferences together;we train the embeddings for authors separately.

Due to the large number of input entities within the knowledge graph, we try to minimize the overall input size and thereby the memory requirement for training. We first filter out the relationships we aim to model. To further reduce memory consumption, we "abbreviate" relations by removing their prefixes.

Furthermore, we use a mapping for entities and relations to further reduce memory consumption. All entities and relations are mapped to a specific index in the form of an integer. In this way, all statements within the knowledge graph are reduced to a triple of integers and used as input for training together with the mapping files.

5.4.2. Evaluation Setup

We use the Python package DGL-KE [25] for our implementation of knowledge graph embedding algorithms. DGL-KE is a recently published package optimized for training knowledge graph embeddings at a large scale. It outperforms other state-of-the-art packages while achieving linear scaling with machine resources as well as high model accuracies. We set the dimension size of our output embeddings to 100. We set the limit due to greater memory constraints for training higher dimensional embeddings. We experiment with a dimension size of 150 and did not observe any improvements to our metrics. Embedding sizes any higher will result in out of memory errors on our setup. The exact choices of hyperparameters are in Table 20. We perform evaluation through randomly masking entities and statements and trying to re-predict the correct missing part.

We perform training on the bwUnicluster using GPU nodes with 8 Nvidia Tesla V100 GPUs and 752GB of RAM.

We use standard ranking metrics Hit@k, mean rank (MR), and mean reciprocal rank (MRR).

[17] Also current implementations of RDF2Vec, such as pyRDF2Vec, are not designed for such a large scale: "Loading large RDF files into memory will cause memory issues as the code is not optimized for larger files."https://github.com/IBCNServices/pyRDF2Vec. This turned out to be true when running RDF2Vec on the MAKG. For the difference between RDF2Vec and other algorithms, such as TransE, we can refer to [109].

[18] https://makg.org/

Table 21
Evaluation results of various embedding types

	TransR*	**TransE**	**RESCAL**	**ComplEx**	**DistMult**
average MR	*105.598*	15.224	4.912	**1.301**	2.094
average MRR	*0.388*	0.640	0.803	**0.958**	0.923
average HITS@1	*0.338*	0.578	0.734	**0.937**	0.893
average HITS@3	*0.403*	0.659	0.851	**0.975**	0.945
average HITS@10	*0.474*	0.769	0.920	**0.992**	0.977
training time	*10 hours*	8 hours	18 hours	8 hours	8 hours

Table 22
Evaluation of final embeddings

	Author	**Paper/Journal/Conf.**
average MR	2.644	1.301
average MRR	0.896	0.958
average HITS@1	0.862	0.937
average HITS@3	0.918	0.975
average HITS@10	0.960	0.992

5.5. Evaluation Result

Evaluation results can be found in Table 21. Results are based on embeddings trained on paper, journal, and conference entities. We observed an average mean rank of 1.301 and a mean reciprocal rank of 0.958 for the best performing embedding type.

Interestingly, TransE and TransR greatly outperform other algorithms during fewer training steps (1,000). For higher training steps, the more modern models, such as ComplEx and DistMult, achieve state-of-the-art performance. Across all metrics, ComplEx, which is based on complex embeddings instead of real-valued embeddings, achieves the best results (e.g., MRR of 0.958 and HITS@1 of 0.937) while having competitive training times to other methods. Due to our focus on the MAKG and not on machine learning and limited resources, we were unable to perform a full-scale analysis of the effect of hyperparameters on the embedding quality. It is important to note that we train the TransR embedding type on 250,000 max training steps compared to 1,000,000 for all others embedding types. This is due to the extremely long training time for this specific embedding; we were unable to finish training in 48 hours, and, therefore, had to adjust the training steps manually. The effect can be seen in its performance; though for lower training steps, TransR performed similarly to TransE.

Table 22 showcases the quality of our final embeddings, which we published on our website.[19]

5.6. Discussion

The main challenge of the task lies in the hardware requirement for training embeddings on such a large scale. For publications, for instance, even after the approaches we have carried out for reducing memory consumption, it still required a significant amount of memory. For example, we were not able to train publications and author embeddings simultaneously given 750 GB of memory space. Given additional resources, future researchers could increase the dimensionality of embeddings, which might increase performance.

Other embedding approaches may be suitable for our case as well, though the limiting factor here is the large file size of the input graph. Any approach needs to be scalable and perform efficiently on such large data sets. One of the limiting factors for choosing embedding types (e.g., TransE) is the availability of an efficient implementation. The DGL-KE provides such implementations, but only for a select number of embedding types. In the future, as other implementations become publicly available, further evaluations may be performed. Alternatively, custom implementations can also be developed, though such tasks are not subject to our paper.

Future researchers might further experiment with various combinations of hyperparameters. We have noticed a great effect of training steps on embedding qualities of various models. Other effects might be learnable with additional experimentations.

[19] https://makg.org/

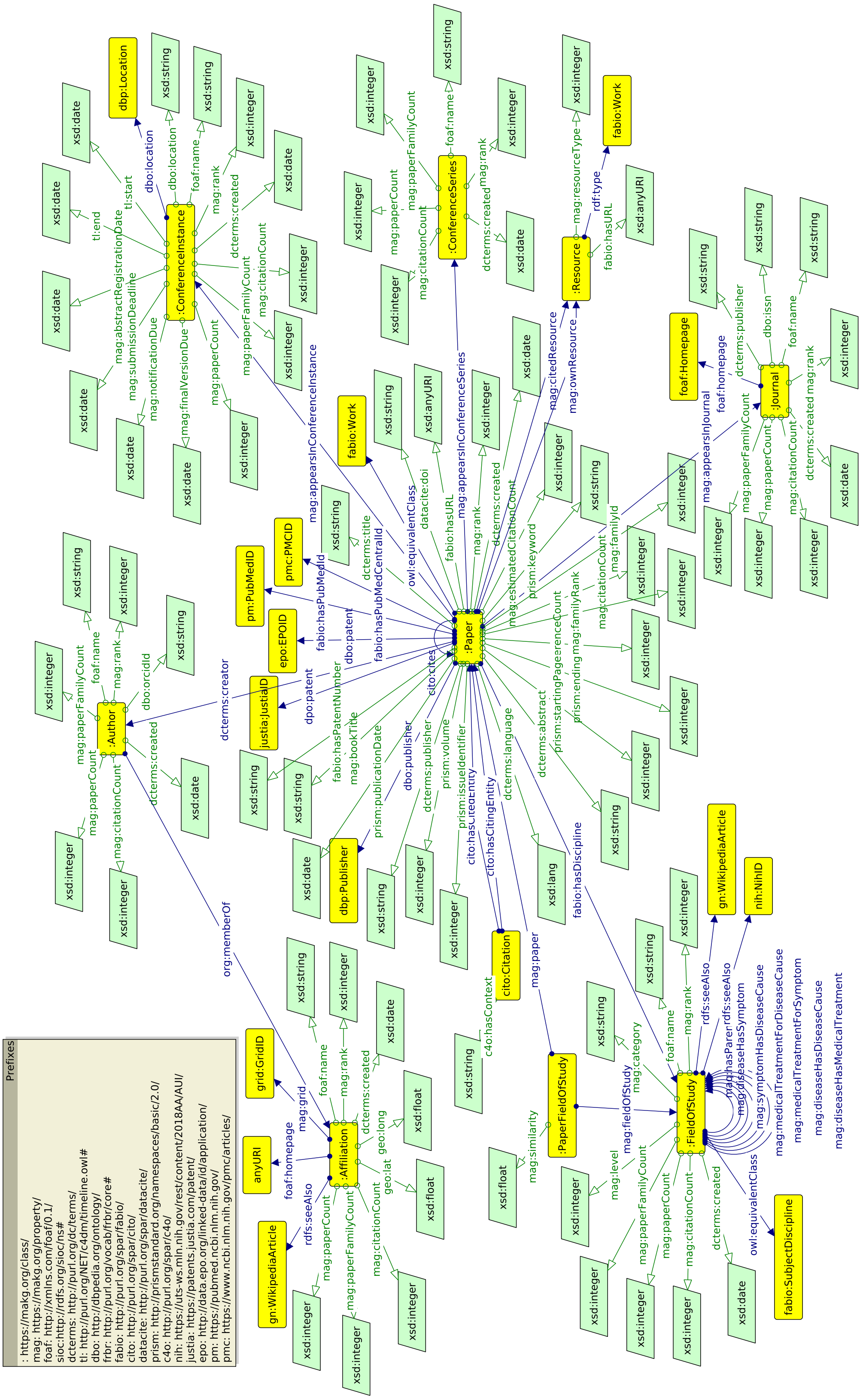

Figure 7.: Updated MAKG schema.

Table 23
Properties added to the MAKG using the prefixes shown in Figure 7

Property	Domain	Range
`https://makg.org/property/paperFamilyCount`	`:Author`	`xsd:integer`
	`:Affiliation`	`xsd:integer`
	`:Journal`	`xsd:integer`
	`:ConferenceSeries`	`xsd:integer`
	`:ConferenceInstance`	`xsd:integer`
	`:FieldOfStudy`	`xsd:integer`
`https://makg.org/property/ownResource`	`:Paper`	`:Resource`
`https://makg.org/property/citedResource`	`:Paper`	`:Resource`
`https://makg.org/property/resourceType`	`:Resource`	`xsd:integer`
`http://www.w3.org/1999/02/22-rdf-syntax-ns#type`	`:Resource`	`fabio:Work`
`http://purl.org/spar/fabio/hasURL`	`:Resource`	`xsd:anyURI`
`https://makg.org/property/familyId`	`:Paper`	`xsd:integer`
`https://makg.org/property/isRelatedTo`	`:Affiliation`	`:Affiliation`
	`:Journal`	`:Journal`
	`:ConferenceSeries`	`:ConferenceSeries`
	`:FieldOfStudy`	`:FieldOfStudy`
`https://makg.org/property/recommends`	`:Paper`	`:Paper`
`http://prismstandard.org/namespaces/basic/2.0/keyword`	`:Paper`	`xsd:string`
`http://www.w3.org/2003/01/geo/wgs84_pos#lat`	`:Affiliation`	`xsd:float`
`http://www.w3.org/2003/01/geo/wgs84_pos#long`	`:Affiliation`	`xsd:float`
`http://dbpedia.org/ontology/location`	`:ConferenceInstance`	`dbp:location`
`http://dbpedia.org/ontology/publisher`	`:Paper`	`dbp:Publisher`
`http://dbpedia.org/ontology/patent`	`:Paper`	`epo:EPOID` `justia:JustiaID`
`http://purl.org/spar/fabio/hasPatentNumber`	`:Paper`	`xsd:string`
`http://purl.org/spar/fabio/hasPubMedId`	`:Paper`	`pm:PubMedID`
`http://purl.org/spar/fabio/hasPubMedCentrialId`	`:Paper`	`pmc:PMCID`
`http://www.w3.org/2000/01/rdf-schema#seeAlso`	`:FieldOfStudy`	`gn:WikipediaArticle` `nih:NihID`

6. Knowledge Graph Provisioning and Statistical Analysis

In this section, we outline how we provide the enhanced MAKG. Furthermore, we show the results of a statistical analysis on various aspects of the MAKG.

6.1. Knowledge Graph Provisioning

For creating the enhanced MAKG, we followed the initial schema and data model of Färber [11]. However, we introduced new properties to model novel relationships and data attributes. A list of all new properties to the MAKG ontology can be found in Table 23. An updated schema for the MAKG is in Figure 7 and on the MAKG homepage,[20] together with the updated ontology.

Besides the MAKG, Wikidata models millions of scientific publications. Thus, similar to the initial MAKG [11], we created mappings between the MAKG and Wikidata in the form of `owl:sameAs` statements. Using the DOI as unique identifier for publications, we were able to create 20,872,925 links between the MAKG and Wikidata.

The MAKG RDF files—containing 8.7 billion RDF triples as the core part—are available at http://doi.org/10.5281/zenodo.4617285. The updated SPARQL endpoint is available at https://makg.org/sparql.

[20] https://makg.org/

Table 24
General author and paper statistics

Metric	**Value**
Average Author per Paper	2.6994
Maximum Author per Paper	7,545
Average Paper per Author	2.6504
Maximum Paper per Author	8,551
Average Coauthors per Author	10.6882
Maximum Coauthors per Author	65,793

6.2. General Statistics

Similar to analyses performed by Herrmannova & Knoth [110] and Färber [11], we aim to provide some general data set statistics regarding the content of the MAKG. Since the last publication, the MAG has received many updates in the form of additional data entries, as well as some small to moderate data schema changes. Therefore, we aim to provide some up-to-date statistics of the MAKG and further detailed analyses of other areas.

We carried out all analysis using the MAKG based on the MAG data as of June 2020 and our modified variants (i.e., custom fields of study and enhanced author set).

6.2.1. Authors

The original MAKG encompasses 243,042,675 authors, of which 43,514,250 had an affiliation given in the MAG. Our disambiguation approach reduced this set to 151,355,324 authors.

Table 24 showcases certain author statistics with respect to publication and cooperation. The average paper in the MAG has 2.7 authors with the most having 7,545 authors. On average, an author published 2.65 papers according to the MAKG. The author with the highest number of papers published 8,551 papers. The average author cooperated with 10.69 other authors in their combined work, with the most "connected" author having 65,793 coauthors overall, which might be plausible, but is likely misleading due to unclean data to some extent.

6.2.2. Papers

We first analyze the composition of paper entities by their associated type (see Table 2 on page 4). The most frequently found document type is journal articles, followed by patents. A huge proportion of paper entities in the MAKG do not have a document type.

Table 25
General reference and citation statistics

Key statistics	Value
Average references	6.8511
At least one reference	78,684,683
Average references (filtered)	20.7813
Median references (filtered)	12
Most references	26,690
Average citations	6.8511
At least one citation	90,887,343
Average citations (filtered)	17.9912
Median citations (filtered)	4
Most citations	252,077

In the following, we analyze the number of citations and references for papers within the MAKG. The results can be found in Table 25.

The average paper in the MAKG references 6.85 papers and received 6.85 citations. The exact match in numbers here seems too unlikely to be coincidental. Therefore, we suspect these numbers to be a result of a closed referencing system of the original MAG, meaning references for a paper are only counted if they reference another paper within the MAG; and citations are only counted if a paper is cited by another paper found in the MAKG. When we remove papers with zero references, we are left with a set of 78,684,683 papers. The average references per paper from the filtered paper set is now 20.78. In the MAKG, 90,887,343 papers are cited at least once, with the average among this new set being 17.99. As averages are highly susceptible to outliers, which were frequent in our data set due to unclean data and due to the power law distribution of scientific output, we also calculated the median of references and citations. These values should give us a more representative picture of reality. The paper with the most references from the MAG has 26,690 references, whereas the paper with the most citations received 252,077 citations as of June 2020.

Table 26 showcases detailed reference and citation statistics for each document type found in our (enhanced) MAKG. Unsurprisingly, books have the most amount of references on average due to their significant lengths, followed by journal papers (and book sections). However, the median value for books is less than for journals, likely due to outliers. Citation wise, books and journal papers again are the most cited document types on average. Again, journal papers have fewer citations on average but a higher median value.

Table 26

Detailed reference and citation statistics

	Journal	**Conference**	**Patent**	**Book**	**BookSection**	**Repository**	**Data Set**	***No Data***
Average references	13.089	10.309	3.470	2.460	3.286	11.649	0.063	2.782
At least one reference	42,660,071	3,913,744	19,023,288	93,644	339,439	1,305,000	130	11,349,367
Average references (filtered)	26.313	12.400	9.643	56.315	26.268	14.988	18.969	21.758
Median references (filtered)	20	10	5	15	6	7	7	10
Most references	13,220	4,156	19,352	5,296	7,747	2,092 196	26,690	
Average citations	14.729	9.024	3.225	29.206	0.813	2.251	0.188	1.019
At least one citation	50,599,935	3,063,123	22,591,991	1,299,728	351,448	549,526	1,187	12,430,405
Average citations (filtered)	24.963	13.869	7.547	48.177	6.277	6.878	6.240	7.274
Median citations (filtered)	8	4	3	7	2	2	1	2
Most citations	252,077	34,134	32,096	137,596	4,119	20,503	633	103,540

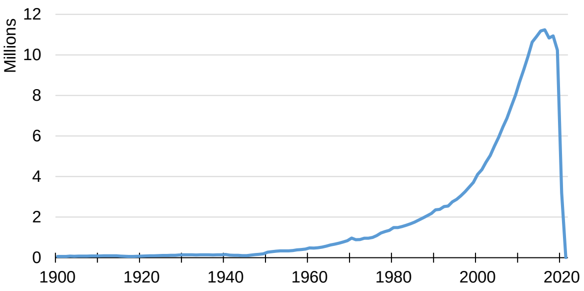

Figure 8. Number of papers published per year (starting with 1900).

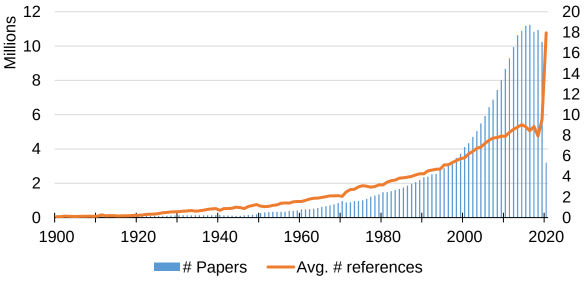

Figure 9. Average number of references of a paper per year

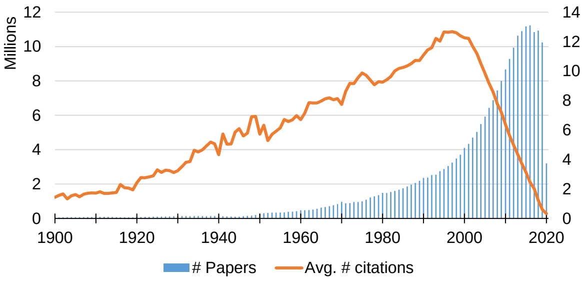

Figure 10. Average number of citations of a paper per year

Figure 8 shows the number of papers published each year in the time span recorded by the MAKG (1800 – present). The number of publications has been on a steady exponential trajectory. This is, of course, partly due to advancements in the digitalization of libraries and journals, as well as the increasing ease of accessing new research papers. However, we can certainly attribute a large part of the growth to the increasing number of publications every year [8].

Interestingly, the average number of references per paper has been on a steady increase (see Figure 9 and [8]). This could be due to a couple of reasons. First, as scientific fields develop and grow, novel work becomes increasingly rare. Rather, researchers publish work built on top of previous research ("on the shoulders of giants"), leading to a growing number of references for new publications. Furthermore, the increasing number of research papers further contribute to more works being considered for referencing. Second, development in technology, such as digital libraries, enable the spread of research and ease the sharing of ideas and communication between researchers (see, e.g., the open access efforts [111]). Therefore, a researcher from the modern age has a huge advantage in accessing other papers and publications. The ease of access could contribute to more works being referenced in this way. Third, as the MAKG is (most likely) a closed reference system, meaning papers referenced are only included if they are part of the MAKG, and as modern publications are more likely to be included in the MAKG, newer papers will automatically have a higher number of recorded references in the MAKG. Although this is a possibility, we do not suspect it to be the main reason behind the rising number of references. Most likely, the cause is a combination of several factors.

Surprisingly, the average number of citations a paper receives has increased as shown in Figure 10. Intuitively, one would assume older papers to receive more

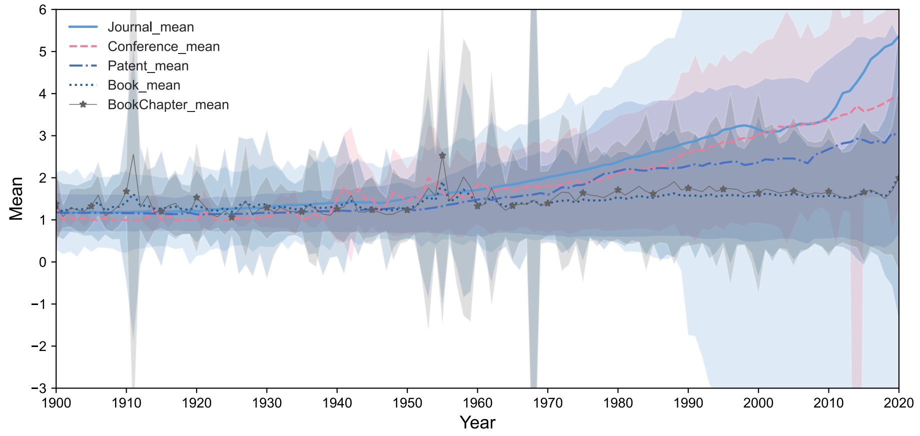

Figure 11. Average number of authors per paper and paper type over the years including standard deviation.

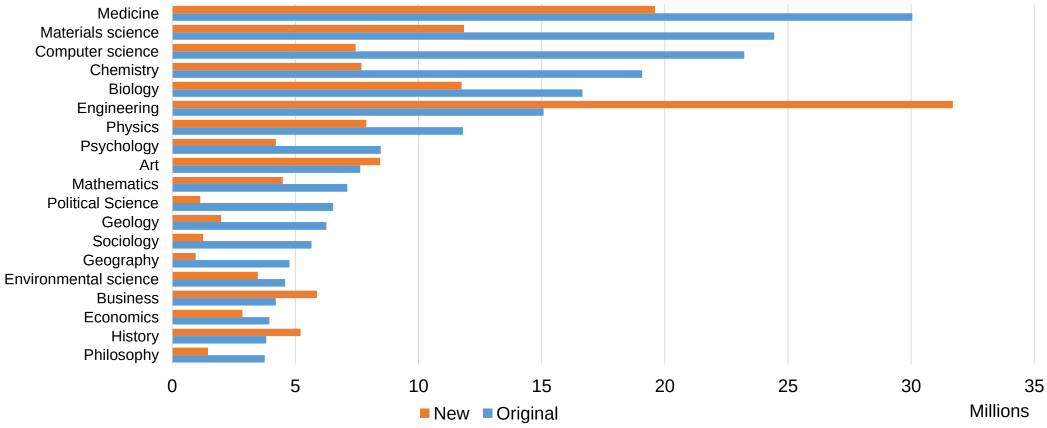

Figure 12. Number of papers per field of study.

citations on average purely due to longevity. However, as our graph shows, the number of citations an average paper receives has increased since the turn of the last century. We observe a peak of growth around 1996, which might be where the age of a paper exhibits its effect. Coupled with the exponential growth of publications, the average citations per paper plummets.

Figure 11 shows the average number of authors per paper per year and publication type, using the MAKG paper's publication year. As we can observe, there has been a clear upward trend for the average number of authors per paper specifically concerning journal articles, conference papers, and patents since the 1970s. The level of cooperation within the scientific community has grown, partly led by the technological developments that enable researchers to easily connect and cooperate. This finding reconfirms the results from the STM report 2018 [8].

6.2.3. *Fields of Study*

In the following, we analyze the development of fields of study over time. First, Figure 12 showcases the current number of publications per top-level field of study within the MAKG. Each field of study here has two distinct values. The blue bars represent the field of study as labeled by the MAKG, whereas the red bars are labels as generated by our custom classifier. Importantly, there is a discrepancy between the total number of paper labels between the original MAKG field of study labels and our custom labels. The original MAG hierarchy includes labels for 199,846,956 papers. Our custom labels are created through classification of paper abstracts and are therefore limited by the number of abstracts available in the data set; thus, we only generated labels for 139,227,097 papers. Rather surprisingly, the disciplines of medicine and materials science are the most common fields of study within the MAG, according to the original MAG field

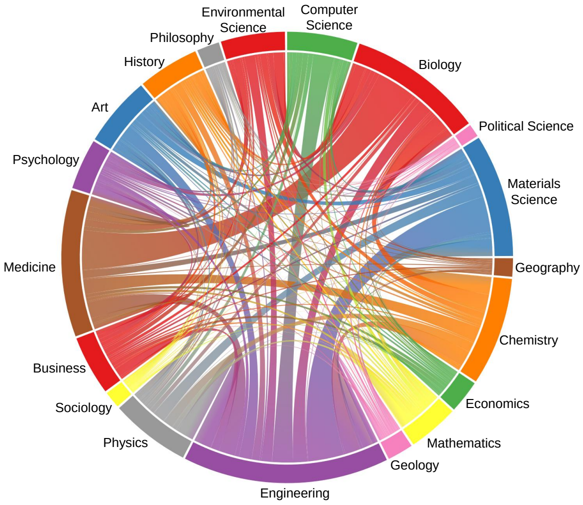

Figure 13. Interdisciplinary researchers in form of authors who publish in multiple fields of study.

of study labels. According to our classification, engineering and medicine are the most represented disciplines.

Evaluating the cumulative number of papers associated to the different fields of study over the years, we can confirm the exponential growth of scientific output shown by Larsen and von Ins [112]. In many areas, our data shows greater rates of growth than previously anticipated.

Figure 13 showcases the interdisciplinary works of authors. Here, we modeled the relationships between fields of study in a chord graph. Each chord between two fields of study represents authors who have published papers in both disciplines. The thickness of each chord is representative of the number of authors who have done so. We observe strong relationships between the disciplines of biology and medicine, materials science and engineering, and computer science and engineering. Furthermore, there is a moderately strong relationship between the disciplines of chemistry and medicine, biology and engineering, and chemistry and biology. The multitude of links between engineering and other disciplines could be due to mislabeling of engineering papers, as our classifier is not adept at classifying papers from engineering in comparison to other fields of study, as shown in Table 19.

7. Conclusion and Outlook

In this paper, we developed and applied several methods for enhancing the Microsoft Academic Knowledge Graph (MAKG), a large-scale scholarly knowledge graph. First, we performed author name disambiguation on the set of 243 million authors using background information, such as the metadata of 239 million publications. Our classifier achieved a precision of 0.949, a recall of 0.755, and an accuracy of 0.991. We managed to reduce the number of total author entities from 243 million to 151 million.

Second, we reclassified existing papers from the MAKG into a distinct set of 19 disciplines (i.e., level-0 fields of study). We performed an evaluation of existing labels and determined 55% of the existing labels to be accurate, whereas our newly generated labels achieved an accuracy of approximately 78%. We then assigned tags to papers based on the papers' abstracts to create a more suitable description of paper content in comparison to the preexisting rigid field of study hierarchy in the MAKG.

Third, we generated entity embeddings for all paper, journal, conference, and author entities. Our evaluation showed that ComplEx was the best performing large-scale entity embedding method that we could apply to the MAKG.

Finally, we performed a statistical analysis on key features of the enhanced MAKG. We updated the MAKG based on our results and provided all data sets, as well as the updated MAKG, online at https://makg.org and http://doi.org/10.5281/zenodo.4617285.

Future researchers could further improve upon our results. For author name disambiguation, we believe the results could be further improved by incorporating additional author information from other sources. For field of study classification, future approaches could develop ways to organize our generated paper tags into a more hierarchical system. For the trained entity embeddings, future research could generate embeddings at a higher dimensionality. This was not possible because of the lack of existing efficient scalable implementations of most algorithms. Beyond these enhancements, the MAKG should be enriched with the key content of scientific publications, such as research data sets [16], scientific methods [4], and research contributions [5].

References

[1] T.D. Noia, R. Mirizzi, V.C. Ostuni, D. Romito and M. Zanker, Linked Open Data to support Content-based Recommender Systems, in: *Proceedings of the 8th International Conference on Semantic Systems*, I-SEMANTICS '12, 2012, pp. 1–8.

[2] J. Beel, S. Langer, M. Genzmehr, B. Gipp, C. Breitinger and A. Nürnberger, Research paper recommender system evaluation: a quantitative literature survey, in: *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, RepSys'13, 2013, pp. 15–22.

[3] M. Färber and A. Jatowt, Citation recommendation: approaches and datasets, *Int. J. Digit. Libr.* **21**(4) (2020), 375–405.

[4] M. Färber, A. Albers and F. Schüber, Identifying Used Methods and Datasets in Scientific Publications, in: *Proceedings of the AAAI-21 Workshop on Scientific Document Understanding (SDU'21)@AAAI'21*, 2021.

[5] M.Y. Jaradeh, A. Oelen, K.E. Farfar, M. Prinz, J. D'Souza, G. Kismihók, M. Stocker and S. Auer, Open Research Knowledge Graph: Next Generation Infrastructure for Semantic Scholarly Knowledge, in: *Proceedings of the 10th International Conference on Knowledge Capture*, K-CAP'19, 2019, pp. 243–246.

[6] S. Fathalla, S. Vahdati, S. Auer and C. Lange, Towards a Knowledge Graph Representing Research Findings by Semantifying Survey Articles, in: *Proceedings of the 21st International Conference on Theory and Practice of Digital Libraries*, TPDL'17, 2017, pp. 315–327.

[7] M.D. Wilkinson, M. Dumontier, I.J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L.B. da Silva Santos, P.E. Bourne et al., The FAIR Guiding Principles for scientific data management and stewardship, *Scientific data* **3**(1) (2016), 1–9.

[8] R. Johnson, A. Watkinson and M. Mabe, The STM Report: An Overview of Scientific and Scholarly Publishing, Technical Report, 2018.

[9] M.R. Hoffman, L.D. Ibáñez, H. Fryer and E. Simperl, Smart Papers: Dynamic Publications on the Blockchain, in: *Proceedings of the 15th Extended Semantic Web Conference*, ESWC'18, 2018, pp. 304–318.

[10] M.Y. Jaradeh, S. Auer, M. Prinz, V. Kovtun, G. Kismihók and M. Stocker, Open Research Knowledge Graph: Towards Machine Actionability in Scholarly Communication, *CoRR* **abs/1901.10816** (2019).

[11] M. Färber, The Microsoft Academic Knowledge Graph: A Linked Data Source with 8 Billion Triples of Scholarly Data, in: *Proceedings of the 18th International Semantic Web Conference*, ISWC'19, Springer, 2019, pp. 113–129.

[12] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang and Z. Su, ArnetMiner: Extraction and Mining of Academic Social Networks, in: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'08, 2008, pp. 990–998.

[13] S. Peroni, A. Dutton, T. Gray and D.M. Shotton, Setting our bibliographic references free: towards open citation data, *Journal of Documentation* **71**(2) (2015), 253–277.

[14] R. Wang, Y. Yan, J. Wang, Y. Jia, Y. Zhang, W. Zhang and X. Wang, AceKG: A Large-scale Knowledge Graph for Academic Data Mining, in: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM'18, 2018, pp. 1487–1490.

[15] OpenAIRE, OpenAIRE Research Graph, https://graph.openaire.eu/, Accessed: 2021-02-11.

[16] Michael Färber and David Lamprecht, Data Set Knowledge Graph, https://dskg.org/, Accessed: 2021-02-11.

[17] A. Kanakia, Z. Shen, D. Eide and K. Wang, A Scalable Hybrid Research Paper Recommender System for Microsoft Academic, in: *Proceedings of the 28th World Wide Web Conference*, WWW'19, 2019, pp. 2893–2899.

[18] D. Schindler, B. Zapilko and F. Krüger, Investigating Software Usage in the Social Sciences: A Knowledge Graph Approach, in: *Proceedings of the 17th Extended Semantic Web Conference*, ESWC'20, 2020, pp. 271–286.

[19] M. Färber, Analyzing the GitHub Repositories of Research Papers, in: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, JCDL'20, 2020, pp. 491–492.

[20] Yannis Tzitzikas, Marios Pitikakis, Giorgos Giakoumis, Kalliopi Varouha and Eleni Karkanaki, How Can a University Take its First Steps in Open Data?, in: *Proceedings of the 14th Metadata and Semantics Research Conference*, MTSR'20, 2020.

[21] Temitope Ajileye, Boris Motik, Ian Horrocks, Streaming Partitioning of RDF Graphs for Datalog Reasoning, in: *Proceedings of the 18th Extended Semantic Web Conference*, ESWC'21, 2021.

[22] K. Wang, Z. Shen, C. Huang, C. Wu, D. Eide, Y. Dong, J. Qian, A. Kanakia, A. Chen and R. Rogahn, A Review of Microsoft Academic Services for Science of Science Studies, *Frontiers Big Data* **2** (2019), 45.

[23] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex Embeddings for Simple Link Prediction, in: *Proceedings of the 33rd International Conference on Machine Learning*, ICML'16, 2016, pp. 2071–2080.

[24] P. Ristoski, J. Rosati, T.D. Noia, R.D. Leone and H. Paulheim, RDF2Vec: RDF graph embeddings and their applications, *Semantic Web* **10**(4) (2019), 721–752.

[25] D. Zheng, X. Song, C. Ma, Z. Tan, Z. Ye, J. Dong, H. Xiong, Z. Zhang and G. Karypis, DGL-KE: Training Knowledge Graph Embeddings at Scale, in: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'20, 2020, pp. 739–748.

[26] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.P. Hsu and K. Wang, An Overview of Microsoft Academic Service (MAS) and Applications, in: *Proceedings of the 24th International Conference on World Wide Web Companion*, WWW'15, 2015, pp. 243–246.

[27] D. Herrmannova and P. Knoth, An Analysis of the Microsoft Academic Graph, *D-Lib Magazine* **22**(9/10) (2016).

[28] T. Schubert, A. Jäger, S. Türkeli and F. Visentin, Addressing the Productivity Paradox with Big Data. A literature review and adaptation of the CDM econometric model, Technical Report, 2019.

[29] Y. Qiu, Data Wrangling: Using publicly available Knowledge Graphs (KGs) to construct a domain-specific KG, 2020.

[30] S. Fortunato, C.T. Bergstrom, K. Börner, J.A. Evans, D. Helbing, S. Milojević, A.M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi, A. Vespignani, L. Waltman, D. Wang and A.-L. Barabási, Science of science, *Science* **359**(6379) (2018).

[31] M. Daquino, S. Peroni, D.M. Shotton, G. Colavizza, B. Ghavimi, A. Lauscher, P. Mayr, M. Romanello and P. Zumstein, The OpenCitations Data Model, in: *Proceedings of the 19th International Semantic Web Conference*, ISWC'20, 2020, pp. 447–463.

[32] K. Wang, Z. Shen, C. Huang, C.-H. Wu, Y. Dong and A. Kanakia, Microsoft Academic Graph: When experts are not enough, *Quantitative Science Studies* **1**(1) (2020), 396–413.

[33] B. Roark, L. Wolf-Sonkin, C. Kirov, S.J. Mielke, C. Johny, I. Demirsahin and K.B. Hall, Processing South Asian Languages Written in the Latin Script: the Dakshina Dataset, in: *Proceedings of The 12th Language Resources and Evaluation Conference*, LREC'20, 2020, pp. 2413–2423.

[34] S. Sun, H. Zhang, N. Li and Y. Chen, Name Disambiguation for Chinese Scientific Authors with Multi-Level Clustering, in: *Proceedings of the 2017 IEEE International Conference on Computational Science and Engineering and IEEE International Conference on Embedded and Ubiquitous Computing*, CSE'17/EUC'17, IEEE Computer Society, 2017, pp. 176–182.

[35] K.M. Pooja, S. Mondal and J. Chandra, A Graph Combination With Edge Pruning-Based Approach for Author Name Disambiguation, *J. Assoc. Inf. Sci. Technol.* **71**(1) (2020), 69–83.

[36] H. Wang, R. Wang, C. Wen, S. Li, Y. Jia, W. Zhang and X. Wang, Author Name Disambiguation on Heterogeneous Information Network with Adversarial Representation Learning, in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, AAAI'20, 2020, pp. 238–245.

[37] J. Kim, J. Kim and J. Owen-Smith, Generating automatically labeled data for author name disambiguation: an iterative clustering method, *Scientometrics* **118**(1) (2019), 253–280.

[38] S. Zhang, X. E and T. Pan, A Multi-Level Author Name Disambiguation Algorithm, *IEEE Access* **7** (2019), 104250–104257.

[39] X. Ma, R. Wang and Y. Zhang, Author Name Disambiguation in Heterogeneous Academic Networks, in: *Proceedings of the 16th International Conference on Web Information Systems and Applications*, WISA'19, 2019, pp. 126–137.

[40] K. Kim, S. Rohatgi and C.L. Giles, Hybrid Deep Pairwise Classification for Author Name Disambiguation, in: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM'19, 2019, pp. 2369–2372.

[41] W. Zhang, Z. Yan and Y. Zheng, Author Name Disambiguation Using Graph Node Embedding Method, in: *Proceedings of the 23rd IEEE International Conference on Computer Supported Cooperative Work in Design*, CSCWD'19, 2019, pp. 410–415.

[42] X. Xu, Y. Li, M. Liptrott and N. Bessis, NDFMF: An Author Name Disambiguation Algorithm Based on the Fusion of Multiple Features, in: *Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference*, COMPSAC'18, 2018, pp. 187–190.

[43] K.M. Pooja, S. Mondal and J. Chandra, An Unsupervised Heuristic Based Approach for Author Name Disambiguation, in: *Proceedings of the 10th International Conference on Communication Systems & Networks*, COMSNETS'18, 2018, pp. 540–542.

[44] X. Lin, J. Zhu, Y. Tang, F. Yang, B. Peng and W. Li, A Novel Approach for Author Name Disambiguation Using Ranking Confidence, in: *Proceedings of the 2017 International Workshops on Database Systems for Advanced Applications*, DASFAA'17, 2017, pp. 169–182.

[45] M. Müller, Semantic Author Name Disambiguation with Word Embeddings, in: *Proceedings of the 21st International Conference on Theory and Practice of Digital Libraries*, TPDL'17, 2017, pp. 300–311.

[46] K. Kim, M. Khabsa and C.L. Giles, Random Forest DBSCAN for USPTO Inventor Name Disambiguation, *CoRR* **abs/1602.01792** (2016).

[47] F. Momeni and P. Mayr, Using Co-authorship Networks for Author Name Disambiguation, in: *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*, JCDL'16, 2016, pp. 261–262.

[48] J. Protasiewicz and S. Dadas, A hybrid knowledge-based framework for author name disambiguation, in: *Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics*, SMC'16, 2016, pp. 594–600.

[49] Y. Qian, Q. Zheng, T. Sakai, J. Ye and J. Liu, Dynamic author name disambiguation for growing digital libraries, *Inf. Retr. J.* **18**(5) (2015), 379–412.

[50] H.N. Tran, T. Huynh and T. Do, Author Name Disambiguation by Using Deep Neural Network, in: *Proceedings of the 6th Asian Conference on Intelligent Information and Database Systems*, ACIIDS'14, 2014, pp. 123–132.

[51] E. Caron and N.J. van Eck, Large scale author name disambiguation using rule-based scoring and clustering, in: *Proceedings of the 19th International Conference on Science and Technology Indicators*, STI'14, 2014, pp. 79–86.

[52] C. Schulz, A. Mazloumian, A.M. Petersen, O. Penner and D. Helbing, Exploiting citation networks for large-scale author name disambiguation, *EPJ Data Sci.* **3**(1) (2014), 11.

[53] S. Kastner, S. Choi and H. Jung, Author Name Disambiguation in Technology Trend Analysis Using SVM and Random Forests and Novel Topic Based Features, in: *Proceedings of the 2013 IEEE International Conference on Green Computing and Communications (GreenCom) and IEEE Internet of Things (iThings) and IEEE Cyber, Physical and Social Computing (CPSCom)*, 2013, pp. 2141–2144.

[54] D.R. Wilson, Beyond probabilistic record linkage: Using neural networks and complex features to improve genealogical record linkage, in: *Proceedings of the 2011 International Joint Conference on Neural Networks*, IJCNN'11, 2011, pp. 9–14.

[55] H. Maidasani, G. Namata, B. Huang and L. Getoor, Entity resolution evaluation measure, Technical Report, 2012. https://web.archive.org/web/20180414024919/http://honors.cs.umd.edu/reports/hitesh.pdf.

[56] H.B. Newcombe, J.M. Kennedy, S. Axford and A.P. James, Automatic linkage of vital records, *Science* **130**(3381) (1959), 954–959.

[57] I.P. Fellegi and A.B. Sunter, A theory for record linkage, *Journal of the American Statistical Association* **64**(328) (1969), 1183–1210.

[58] D. Quass and P. Starkey, Record Linkage for Genealogical Databases, in: *Proceedings of the ACM SIGKDD 2003 Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 2003, pp. 40–42.

[59] P. Singla and P.M. Domingos, Entity Resolution with Markov Logic, in: *Proceedings of the 6th IEEE International Conference on Data Mining*, ICDM'06, IEEE Computer Society, 2006, pp. 572–582.

[60] J. Wang, G. Li, J.X. Yu and J. Feng, Entity Matching: How Similar Is Similar, *Proc. VLDB Endow.* **4**(10) (2011), 622–633.

[61] A.A. Ferreira, M.A. Gonçalves and A.H.F. Laender, A brief survey of automatic methods for author name disambiguation, *SIGMOD Rec.* **41**(2) (2012), 15–26.

[62] A. Tekles and L. Bornmann, Author name disambiguation of bibliometric data: A comparison of several unsupervised approaches, in: *Proceedings of the 17th International Conference on Scientometrics and Informetrics*, ISSI'19, 2019, pp. 1548–1559.

[63] J. Kim, Evaluating author name disambiguation for digital libraries: a case of DBLP, *Scientometrics* **116**(3) (2018), 1867–1886.

[64] H. Han, C.L. Giles, H. Zha, C. Li and K. Tsioutsiouliklis, Two supervised learning approaches for name disambiguation in author citations, in: *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, JCDL'04, 2004, pp. 296–305.

[65] M.A. Hernández and S.J. Stolfo, The merge/purge problem for large databases, *ACM Sigmod Record* **24**(2) (1995), 127–138.

[66] M.A. Jaro, Advances in record-linkage methodology as applied to matching the 1985 census of Tampa, Florida, *Journal of the American Statistical Association* **84**(406) (1989), 414–420.

[67] W.E. Winkler, The state of record linkage and current research problems, in: *Statistical Research Division, US Census Bureau*, 1999.

[68] W.W. Cohen, P. Ravikumar and S.E. Fienberg, A Comparison of String Distance Metrics for Name-Matching Tasks, in: *Proceedings of IJCAI-03 Workshop on Information Integration on the Web*, IIWeb'03, 2003, pp. 73–78.

[69] J. Kim, Scale-free collaboration networks: An author name disambiguation perspective, *J. Assoc. Inf. Sci. Technol.* **70**(7) (2019), 685–700.

[70] K. Kowsari, K.J. Meimandi, M. Heidarysafa, S. Mendu, L.E. Barnes and D.E. Brown, Text Classification Algorithms: A Survey, *Inf.* **10**(4) (2019), 150.

[71] J.J. Rocchio, *Relevance Feedback in Information Retrieval*, in: *The SMART Retrieval System - Experiments in Automatic Document Processing*, G. Salton, ed., Prentice Hall, Englewood, Cliffs, New Jersey, 1971.

[72] R.E. Schapire, The Strength of Weak Learnability, *Mach. Learn.* **5** (1990), 197–227.

[73] L. Breiman, Bagging Predictors, *Mach. Learn.* **24**(2) (1996), 123–140. doi:10.1007/BF00058655.

[74] D.R. Cox and E.J. Snell, *Analysis of binary data*, Vol. 32, CRC press, 1989.

[75] V. Vapnik and A.Y. Chervonenkis, A class of algorithms for pattern recognition learning, *Avtomat. i Telemekh* **25**(6) (1964), 937–945.

[76] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning representations by back-propagating errors, *nature* **323**(6088) (1986), 533–536.

[77] S. Hochreiter and J. Schmidhuber, Long Short-Term Memory, *Neural Comput.* **9**(8) (1997), 1735–1780.

[78] Z. Dai, Z. Yang, Y. Yang, J.G. Carbonell, Q.V. Le and R. Salakhutdinov, Transformer-XL: Attentive Language Models beyond a Fixed-Length Context, in: *Proceedings of the 57th Conference of the Association for Computational Linguistics*, ACL'19, 2019, pp. 2978–2988.

[79] J. Devlin, M. Chang, K. Lee and K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'19, 2019, pp. 4171–4186.

[80] Z. Yang, Z. Dai, Y. Yang, J.G. Carbonell, R. Salakhutdinov and Q.V. Le, XLNet: Generalized Autoregressive Pretraining for Language Understanding, in: *Proceedings of the Annual Conference on Neural Information Processing Systems*, NeurIPS'19, 2019, pp. 5754–5764.

[81] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma and R. Soricut, ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, in: *Proceedings of the 8th International Conference on Learning Representations*, ICLR'20, 2020, pp. 1–17.

[82] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer and V. Stoyanov, RoBERTa: A Robustly Optimized BERT Pretraining Approach, *CoRR* **abs/1907.11692** (2019). http://arxiv.org/abs/1907.11692.

[83] I. Beltagy, K. Lo and A. Cohan, SciBERT: A Pretrained Language Model for Scientific Text, in: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, EMNLP-IJCNLP'19, 2019, pp. 3613–3618.

[84] R. Alzaidy, C. Caragea and C.L. Giles, Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents, in: *Proceedings of the 28th World Wide Web Conference*, WWW'19, 2019, pp. 2551–2557.

[85] C. Florescu and C. Caragea, PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents, in: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, ACL'17, 2017, pp. 1105–1115.

[86] S.N. Kim, O. Medelyan, M. Kan and T. Baldwin, Automatic keyphrase extraction from scientific articles, *Lang. Resour. Evaluation* **47**(3) (2013), 723–742.

[87] C. Caragea, F.A. Bulgarov, A. Godea and S.D. Gollapalli, Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach, in: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, EMNLP'14, 2014, pp. 1435–1446.

[88] R. Mihalcea and P. Tarau, TextRank: Bringing Order into Text, in: *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, EMNLP'04, 2004, pp. 404–411.

[89] S. Rose, D. Engel, N. Cramer and W. Cowley, *Automatic Keyword Extraction from Individual Documents*, John Wiley & Sons, 2010, pp. 1–20.

[90] D. Chicco and G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, *BMC genomics* **21**(1) (2020), 6.

[91] A.A. Salatino, T. Thanapalasingam, A. Mannocci, F. Osborne and E. Motta, The Computer Science Ontology: A Large-Scale Taxonomy of Research Areas, in: *Proceedings of the 17th International Semantic Web Conference*, ISWC'18, 2018, pp. 187–205.

[92] A. Brack, J. D'Souza, A. Hoppe, S. Auer and R. Ewerth, Domain-Independent Extraction of Scientific Concepts from Research Articles, in: *Proceedings of the 42nd European Conference on IR*, ECIR'20, 2020, pp. 251–266.

[93] Y. Luan, L. He, M. Ostendorf and H. Hajishirzi, Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction, in: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3219–3232.

[94] Q. Wang, Z. Mao, B. Wang and L. Guo, Knowledge Graph Embedding: A Survey of Approaches and Applications, *IEEE Trans. Knowl. Data Eng.* **29**(12) (2017), 2724–2743.

[95] P. Ristoski, Exploiting Semantic Web Knowledge Graphs in Data Mining, PhD thesis, 2017.

[96] D.Q. Nguyen, An overview of embedding models of entities and relationships for knowledge base completion, *CoRR* **abs/1703.08098** (2017). http://arxiv.org/abs/1703.08098.

[97] A. Bordes, N. Usunier, A. García-Durán, J. Weston and O. Yakhnenko, Translating Embeddings for Modeling Multi-relational Data, in: *Proceedings of the 27th Annual Conference on Neural Information Processing Systems*, NIPS'13, 2013, pp. 2787–2795.

[98] Z. Wang, J. Zhang, J. Feng and Z. Chen, Knowledge Graph Embedding by Translating on Hyperplanes, in: *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI'14, 2014, pp. 1112–1119.

[99] G. Ji, S. He, L. Xu, K. Liu and J. Zhao, Knowledge Graph Embedding via Dynamic Mapping Matrix, in: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, ACL'15, 2015, pp. 687–696.

[100] M. Nickel, V. Tresp and H. Kriegel, A Three-Way Model for Collective Learning on Multi-Relational Data, in: *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, 2011, pp. 809–816.

[101] B. Yang, W. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: *Proceedings of the 3rd International Conference on Learning Representations*, ICLR'15, 2015.

[102] M. Nickel, L. Rosasco and T.A. Poggio, Holographic Embeddings of Knowledge Graphs, in: *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, AAAI'16, 2016, pp. 1955–1961.

[103] T. Dettmers, P. Minervini, P. Stenetorp and S. Riedel, Convolutional 2D Knowledge Graph Embeddings, in: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, AAAI'18, 2018, pp. 1811–1818.

[104] F. Che, D. Zhang, J. Tao, M. Niu and B. Zhao, ParamE: Regarding Neural Network Parameters as Relation Embeddings for Knowledge Graph Completion, in: *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, AAAI'20, 2020, pp. 2774–2781.

[105] F. Lu, P. Cong and X. Huang, Utilizing Textual Information in Knowledge Graph Embedding: A Survey of Methods and Applications, *IEEE Access* **8** (2020), 92072–92088.

[106] G.A. Gesese, R. Biswas, M. Alam and H. Sack, A Survey on Knowledge Graph Embeddings with Literals: Which model links better Literal-ly?, *CoRR* **abs/1910.12507** (2019).

[107] A. Kristiadi, M.A. Khan, D. Lukovnikov, J. Lehmann and A. Fischer, Incorporating Literals into Knowledge Graph Embeddings, in: *Proceedings of the 18th International Semantic Web Conference*, ISWC'2019, 2019, pp. 347–363.

[108] Y. Lin, Z. Liu, M. Sun, Y. Liu and X. Zhu, Learning Entity and Relation Embeddings for Knowledge Graph Completion, in: *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI'15, 2015, pp. 2181–2187.

[109] J. Portisch, N. Heist and H. Paulheim, Knowledge Graph Embedding for Data Mining vs. Knowledge Graph Embedding for Link Prediction–Two Sides of the Same Coin? (2021).

[110] D. Herrmannova and P. Knoth, An Analysis of the Microsoft Academic Graph, *D Lib Mag.* **22**(9/10) (2016).

[111] H. Piwowar, J. Priem, V. Larivière, J.P. Alperin, L. Matthias, B. Norlander, A. Farley, J. West and S. Haustein, The state of OA: a large-scale analysis of the prevalence and impact of Open Access articles, *PeerJ* **6** (2018), e4375.

[112] P.O. Larsen and M. von Ins, The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index, *Scientometrics* **84**(3) (2010), 575–603.

[113] W. Ammar, D. Groeneveld, C. Bhagavatula, I. Beltagy, M. Crawford, D. Downey, J. Dunkelberger, A. Elgohary, S. Feldman, V. Ha, R. Kinney, S. Kohlmeier, K. Lo, T. Murray, H. Ooi, M.E. Peters, J. Power, S. Skjonsberg, L.L. Wang, C. Wilhelm, Z. Yuan, M. van Zuylen and O. Etzioni, Construction of the Literature Graph in Semantic Scholar, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL-HLT'18, 2018, pp. 84–91.

[114] T. Backes, The Impact of Name-Matching and Blocking on Author Disambiguation, in: *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM'18, 2018, pp. 803–812.

[115] I. Bhattacharya and L. Getoor, A Latent Dirichlet Model for Unsupervised Entity Resolution, in: *Proceedings of the Sixth SIAM International Conference on Data Mining*, SIAM'06, 2006, pp. 47–58.

[116] S. Chaudhuri, V. Ganti and R. Kaushik, A Primitive Operator for Similarity Joins in Data Cleaning, in: *Proceedings of the 22nd International Conference on Data Engineering*, ICDE'06, IEEE Computer Society, 2006, p. 5.

[117] P. Christen, *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*, Data-Centric Systems and Applications, Springer, 2012.

[118] C.A. D'Angelo, C. Giuffrida and G. Abramo, A heuristic approach to author name disambiguation in bibliometrics databases for large-scale research assessments, *J. Assoc. Inf. Sci. Technol.* **62**(2) (2011), 257–269.

[119] M. Färber, F. Bartscherer, C. Menne and A. Rettinger, Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO, *Semantic Web* **9**(1) (2018), 77–129.

[120] L. Getoor and A. Machanavajjhala, Entity Resolution: Theory, Practice & Open Challenges, *Proc. VLDB Endow.* **5**(12) (2012), 2018–2019.

[121] R.D. Gottapu, C. Dagli and B. Ali, Entity resolution using convolutional neural network, *Procedia Computer Science* **95** (2016), 153–158.

[122] J. Huang, S. Ertekin and C.L. Giles, Efficient Name Disambiguation for Large-Scale Databases, in: *Proceedings of the 10th European Conference on Principles and Practice*

of Knowledge Discovery in Databases, PKDD'06, 2006, pp. 536–544.

[123] G. Li and H. Wang, Improved Automatic Keyword Extraction Based on TextRank Using Domain Knowledge, in: *Proceedings of the 3rd Conference on Natural Language Processing and Chinese Computing*, NLPCC'14, Vol. 496, 2014, pp. 403–413.

[124] W. Li and J. Zhao, TextRank Algorithm by Exploiting Wikipedia for Short Text Keywords Extraction, in: *Proceedings of the 3rd International Conference on Information Science and Control Engineering*, ICISCE'16, 2016, pp. 683–686.

[125] B. Marr, How much data do we create every day? The mind-blowing stats everyone should read, 2018, https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/, Accessed: 2021-03-03.

[126] D.G. Mestre, C.E.S. Pires, D.C. Nascimento, A.R.M. de Queiroz, V.B. Santos and T.B. Araújo, An efficient spark-based adaptive windowing for entity matching, *J. Syst. Softw.* **128** (2017), 1–10.

[127] A.N. Ngomo and S. Auer, LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data, in: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, IJCAI'11, 2011, pp. 2312–2317.

[128] T. Pay, S. Lucci and J.L. Cox, An Ensemble of Automatic Keyword Extractors: TextRank, RAKE and TAKE, *Computación y Sistemas* **23**(3) (2019).

[129] R. Singh, V.V. Meduri, A.K. Elmagarmid, S. Madden, P. Papotti, J. Quiané-Ruiz, A. Solar-Lezama and N. Tang, Synthesizing Entity Matching Rules by Examples, *Proc. VLDB Endow.* **11**(2) (2017), 189–202.

[130] A. Tang, C. Wu, J. Liu, W. Wang, X. Yang and Y. Xing, Parallel Computing for Large-Scale Author Name Disambiguation in MEDLINE, in: *Proceedings of the 21st IEEE International Conference on High Performance Computing and Communications; 17th IEEE International Conference on Smart City; 5th IEEE International Conference on Data Science and Systems*, HPCC/SmartCity/DSS'19, 2019, pp. 1580–1586.

[131] I. Turc, M. Chang, K. Lee and K. Toutanova, Well-Read Students Learn Better: The Impact of Student Initialization on Knowledge Distillation, *CoRR* **abs/1908.08962** (2019).

[132] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser and I. Polosukhin, Attention is All you Need, in: *Proceedings of the Annual Conference on Neural Information Processing Systems*, NIPS'17, 2017, pp. 5998–6008.

[133] Y. Wen, H. Yuan and P. Zhang, Research on Keyword Extraction Based on word2vec Weighted TextRank, in: *Proceedings of the 2nd IEEE International Conference on Computer and Communications*, ICCC'16, 2016, pp. 2109–2113.

[134] X. Tian, Study on Keyword Extraction Using Word Position Weighted TextRank, *New Technology of Library and Information Service* **29**(9) (2013), 30–34.

[135] S. Zhang, E. Xinhua, T. Huang and F. Yang, ANDMC: An Algorithm for Author Name Disambiguation Based on Molecular Cross Clustering, in: *Proceedings of the International Conference on Database Systems for Advanced Applications*, DASFAA'19, 2019, pp. 173–185.

[136] L. Zhu, M. Ghasemi-Gol, P.A. Szekely, A. Galstyan and C.A. Knoblock, Unsupervised Entity Resolution on Multi-type Graphs, in: *Proceedings of the 15th International Semantic Web Conference*, ISWC'16, 2016, pp. 649–667.

[137] P.J. Dendek, L. Bolikowski and M. Lukasik, Evaluation of Features for Author Name Disambiguation Using Linear Support Vector Machines, in: *Proceedings of the 10th IAPR International Workshop on Document Analysis Systems*, DAS'12, 2012, pp. 440–444.

[138] D. Fensel, U. Simsek, K. Angele, E. Huaman, E. Kärle, O. Panasiuk, I. Toma, J. Umbrich and A. Wahler, *Knowledge Graphs - Methodology, Tools and Selected Use Cases*, Springer, 2020.

[139] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, J.E.L. Gayo, S. Kirrane, S. Neumaier, A. Polleres, R. Navigli, A.N. Ngomo, S.M. Rashid, A. Rula, L. Schmelzeisen, J.F. Sequeda, S. Staab and A. Zimmermann, Knowledge Graphs, *CoRR* **abs/2003.02320** (2020).

[140] M. Färber, A. Sampath and A. Jatowt, PaperHunter: A System for Exploring Papers and Citation Contexts, in: *Proceedings of the 41th European Conference on Information Retrieval*, ECIR'19, 2019, pp. 246–250.