Meta2KG: An Embeddings-based Approach for Transforming Metadata to Knowledge Graphs

Nora Abdelmageed[1,2,*], Birgitta König-Ries[1,2]

[1]Heinz Nixdorf Chair for Distributed Information Systems, Friedrich Schiller University Jena
[2]Michael Stifel Center Jena, Friedrich Schiller University Jena

Abstract
Metadata is used to describe data. It includes information about the who, when, where, how, and why of data collection. Ideally, it should be in a machine-understandable format like RDF. This enables data queries using structured query languages like SPARQL and empowers further data usage. In this paper, we investigate metadata as a source for generating Knowledge Graphs (KGs). We introduce a semi-automated approach that transforms raw metadata files into a KG. We develop the Biodiversity Metadata Ontology (BMO) as an underlying schema for our technique. We auto-populate the constructed ontology with instances from several metadata files as a unified KG. Finally, we discuss the common obstacles that face such a transformation procedure. Our results show that metadata files are a promising source for KG construction. In addition, our resources and code are publicly available[1].

Keywords
Metadata Analysis, RDF, Ontology Matching, Ontology Population, Knowledge Graphs, Embeddings

1. Introduction

A Knowledge Graph (KG) is a graph-based model built to accumulate and convey knowledge of the real world; it contains a set of nodes and edges representing entities of interest and their relations [1, 2]. Auer et al. [3] propose them to bring scholarly communication to the 21st century. While publications are one way to encode knowledge around an area of research, observational and experimental data are another. For example, a large amount of heterogeneous data is collected and generated in biodiversity research. Integrating this data remains a significant challenge [4]. Metadata are often associated with biodiversity datasets, describing them in various ways, for example, who, when, where, how, and why the data is collected. A metadata file contains essential information for various applications, like dataset search and Question Answering (QA) [5]. One way to exploit this untapped wealth is by transforming this raw metadata into KGs. Page [6] demonstrates a biodiversity-specific KG example, with this, we can increase the FAIRness [7] of the data by enhancing its re-usability. For example, we enable data querying using a structured query language like SPARQL.

Embeddings are a well-established technique that captures the semantics of a given word or sentence. Previous works have shown their significant impact on many Natural Language

[1]https://github.com/fusion-jena/Meta2KG

KGCW'23: 4th International Workshop on Knowledge Graph Construction, May 28, 2023, Crete, GRE

*Corresponding author.
nora.abdelmageed@uni-jena.de (N. Abdelmageed); birgitta.koenig-ries@uni-jena.de (B. König-Ries)
0000-0002-1405-6860 (N. Abdelmageed); 0000-0002-2382-9722 (B. König-Ries)

Processing (NLP) applications [8, 9, 10]. In this work, we transform raw metadata files into a KG using an embedding-based matching technique. We demonstrate the effectiveness of the method and discuss common challenges in the automated transformation process. We tested our technique on a biodiversity use case; however, we expect our method to be domain-independent since we do not rely on any domain-specific mapping rules.

The proposed approach yields several research questions that we try to answer for the first time in this paper:

- ***RQ1****: Is it possible to construct a KG using metadata as the only data source?*
- ***RQ2****: How can we automate the transformation of metadata to KG?*
- ***RQ3****: What are the challenges facing such automated transformation?*

We distinguish the contributions of this paper compared to the previously published poster paper [11] as follows:

- Biodiversity Metadata Ontology (BMO), our data model, is a hand-crafted schema for biodiversity metadata.
- Embedding-based approach that maps from metadata to BMO.
- Auto population technique of BMO with triple validation.
- Evaluation of the matching technique and discussion on the faced challenges.
- Biodiversity Metadata Knowledge Graph (BMKG), the resulting knowledge graph that is automatically generated for the biodiversity metadata.

The rest of this paper is organized as follows: We give an overview of related work in Section 2. In Section 3, we demonstrate our approach. We discuss the results and answer our research questions in Section 4 and Section 5. Finally, we conclude in Section 6.

2. Related Work

In this section, we give an overview of the related work.

SCM-KG [12] integrates scholarly communication metadata into a KG from two different sources DBLP[1] and Microsoft Academic Graph (MAG)[2]. Their motivation is the disambiguation of personal entities that represent authors. Such entities included a list of publication IDs as a disambiguation property. The authors claimed the completeness of SCM-KG since each data source covers different aspects. For example, DBLP has a complete listing of authors and publications. However, MAG has more keywords and abstracts. The authors introduced a pipeline that consists of 1) two manual steps concerning data acquisition and pre-processing. 2) three automatic steps, including ontology matching using rule-based techniques, similarity measurement, and instance linking. The authors deal with various data sources like CSVs, PDFs, and structured databases. Such heterogenous input may use different schemas. e.g., DBLP and MAG model the same concepts (e.g., affiliation) differently. Thus, the authors involved a mapping step in creating their target unified graph through an ontology engineering phase. They used subsets from Dublin Core and FOAF ontology, and they created missing vocabulary

[1] https://dblp.org/

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

themselves. They provided an entity linking step to their pipeline for ontology matching via a Jaccard similarity. They used the common title and the publication year, if provided, to match the instances to the ontology.

ENVENTS [13] introduced a dataset for top-tier conferences in the computer science field, e.g., ISWC, ESWC, CVPR. It encapsulates scientific events in terms of historical data about the publications, submissions, start date, end date, location, and homepage for 25 top-prestigious event series (718 editions in total) in five computer science communities. The authors manually collected and analyzed the metadata (raw data) since 1990 of these conferences from different sources like DBLP, and ACM Digital Libraries[3]. Then, they applied a pre-processing data phase where they aimed to fill in the missing data, identify and correct incorrect data, and remove irrelevant information. Thus, four tasks are involved in this phase: data integration, data cleansing, data transformation, and event name unification. Then, the authors analyzed the collected metadata of the events in terms of, e.g., the h5 index, average acceptance rate, and the number of editions of each event. The primary use case of such work is Question Answering (QA). The dataset is publicly available online in three formats (CSV, XML, and RDF).

ENVENTSKG [14, 15] is the successor of the previous work. The authors released their dataset as a unified KG instead of individual RDF dumps, including data for more computer science communities. I.e., EVENTSKG is a KG that contains metadata of top-40 prestigious events series. Like EVENTS, the main goal of EVENTSKG is to facilitate the analysis of events metadata by enabling them to be queried using semantic query languages like SPARQL. This work relies on the Scientific Event Ontology (SEO) [16] as a data model. Two steps are included to enhance their previous pipeline. On the one hand, for the linked data generation, where the authors developed an RDFer, a Java tool to convert input data from CSV to linked data (RDF/XML syntax). On the other hand, the linked data enrichment (LDE) is included to infer the interlinking relationships between RDF triples using inference engines, i.e., reasoners.

Schröder et al. [17] managed to create a Personal Knowledge Graph (PKG) from file names as the only data source used in a semi-automatic approach. File names are considered metadata for files that have minimal context. Despite the unusual source to create a KG, a user that is defined as a knowledge engineer is responsible for creating the RDF triples. However, an active learning technique aids the knowledge engineer by suggesting entity types. The authors used rule-based techniques to extract terminologies of interest. They followed several steps to unify the extracted entities and populate the ontology. Then, they conducted taxonomic and non-taxonomic relations using language resources. The authors used Jaccard and Embedding-based similarities, for instance and type matching, respectively.

3. Approach

Figure 1 shows the seven phases of our semi-automated pipeline that we detail in the following sections. It consists of 1) A description of the data sources we used to develop the data model and evaluate our matching technique (Data Acquisition). 2) The preprocessing that we applied on the collected metadata files to facilitate its interpretation (Preprocessing). 3) The process of our data modeling (Ontology Development). 4) The embedding sources we used to generate the

[3] https://dl.acm.org/

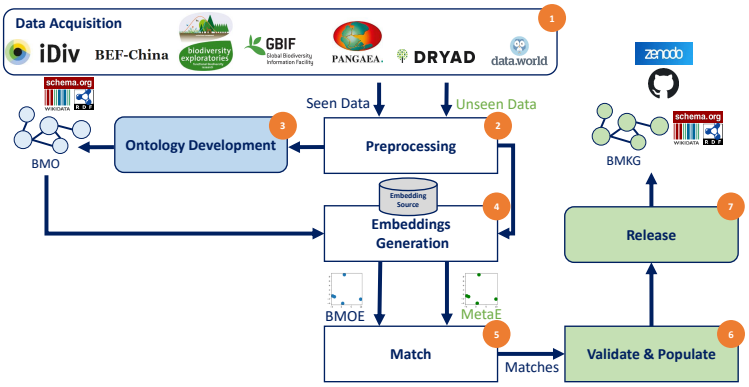


Figure 1: Overview of our raw metadata to KG transformation workflow. Rectangle shapes are automated modules. Oval shapes are manual steps.

word embeddings in addition to vectors construction methods (Embeddings Generation). 5) Our similarity measurement and ontology matching techniques (Match). 6) Our auto-population technique with supported datatype validations (Validate & Populate), and finally, 7) how we published and indexed our contributions, including the Biodiversity Metadata Knowledge Graph (BMKG) (Release). We used the first fold of the collected metadata, "Seen Data" to develop the underlying ontology Biodiversity Metadata Ontology (BMO) and to generate the ontological embeddings BMOE. We used the second fold of the collected metadata, "Unseen Data" for ontology matching and auto-population. Both "Data Acquisition", "Ontology Development", and "Release" stages involve manual labor. The rest of the modules are fully automated.

3.1. Data Acquisition

The first step in this work is to decide the sources of metadata files. We decided to collect them from seven biodiversity data portals that have various characteristics. These portals are German Centre for Integrative Biodiversity Research (iDiv)[4], BEF-China[5], Biodiversity Exploratories (BExIS)[6], Global Biodiversity Information Facility (GBIF)[7] and data.world[8]. In addition, we included biodiversity-related metadata from PANGAEA[9] and, Dryad[10], both are well-established data publishers for ecological data. We queried these portals using 20 keywords identified as typical for the biodiversity domain [18] including, e.g., "abundance", "benthic", "biomass", "carbon", "climate change", "decomposition", "earthworms", "ecosystem". We picked the first 50 datasets from each repository from the search results and selected the complete ones, those that have mostly completed their metadata fields. This manual inspection ensures domain specificity as well. Figure 2 shows the overall distribution of the selected metadata files over the repositories. We divided the collected data into **Seen** and **Unseen** data. For the

[4]https://data.botanik.uni-halle.de/bef-china/
[5]https://bef-china.com/
[6]https://www.biodiversity-exploratories.de/en/
[7]https://www.gbif.org/
[8]https://data.world/
[9]https://www.pangaea.de/
[10]https://datadryad.org/stash

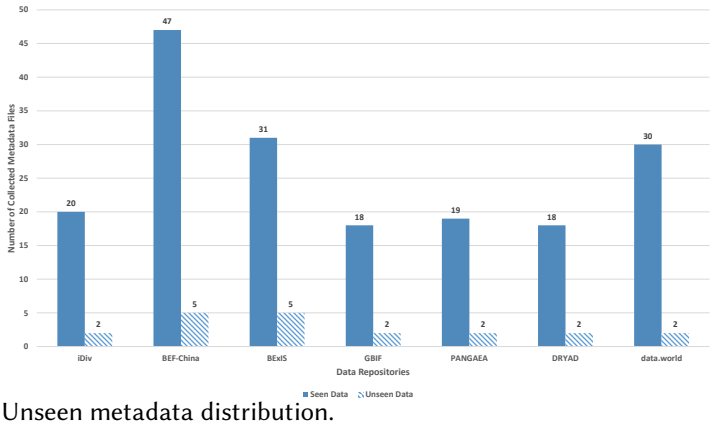


Figure 2: Seen & Unseen metadata distribution.

Unseen data, we picked five files from each repository with the most samples: BEF-China and BExIS, and we selected two files from each of the rest. We considered the remaining metadata samples as Seen data. We use Seen data for modeling the underlying ontology and creating its embeddings. The Unseen data is used to create the ground truth by manually annotating its fields to the BMO; thus, we validate the matching technique. In addition, it is also used to auto-populate the final resultant KG.

3.2. Preprocessing

We applied a preprocessing step to the "Seen data". It included the conversion of the XML files into a key-value data structure. That way, a key encodes the entire hierarchy of a metadata field. For example, the key `dataset.temporalCoverage...calendarDate` corresponds to the XML in Listing 1. Moreover, we cleaned the keys from generic words, e.g., `dataset`, `calandarDate`, `id`, `#text`. We decided on these generic terms by manual analysis of the entire repositories. So, a clean key for this example is `temporalCoverage.beginDate`. We keep the key-value structure, "flat dictionary", in a separate file, and we use it to pre-train word embeddings and triple validation later in this work.

Listing 1: Metadata field XML snippet

```
<dataset id="171">
        <temporalCoverage>
                <rangeOfDates>
                        <beginDate>
                                <calendarDate>
                                        2009/07/31
                                </calendarDate>
                        </beginDate>
                </rangeOfDates>
        </temporalCoverage>
</dataset>
```

3.3. Ontology Development

The target of this phase is to find a common vocabulary for the seven data repositories we decided to work with. After the preprocessing step, we calculated the frequency of each key in the Seen data to analyze the used keys for each data repository and get insights on the most common keys in the biodiversity metadata in general. Table 1 shows a sample of the auto-generated keys after we apply the cleaning steps and their frequencies. The last column depicts our chosen key that would appear in the ontology. The selected format would be the shared vocabulary among all data portals. The selected repositories use various syntactic representations for the same semantic meaning. For example, "abstract" conveys the information from both fields: "Short_Abstract", and "Abstract.Abstract". We manually analyzed the resultant cleaned and grouped keys to develop a shared schema that aligns our data repositories. We kept the "Selected" key with all its synonyms. Such selected keys represent our schema. We use its synonyms to generate the embedding of the key later in this work. We held several meetings with a biodiversity expert to validate and review such schema. During those meetings, we integrated the biodiversity expert's opinion, e.g., we included other vocabularies for one data repository, i.e., BExIS. Thus, in this phase, ontology development is an iterative process where we integrate the feedback from the domain expert.

We used the Python module, rdflib[11] to create the RDF file for the schema, the Biodiversity Metadata Ontology (BMO). We reused existing vocabulary from schema.org. In addition, we defined a new concept under BMO namespace if it did not exist. For example, we reused "Organization", "Person", and "Address" from schema.org. However, we defined both "Taxonomic Coverage" and "Geographic Coverage" using BMO namespace. In addition, we used datatype properties from Wikidata [19] and Dublin Core[12].

Figure 3 depicts the concepts and relations of the Biodiversity Metadata Ontology (BMO). The dashed lines represent the *subClassOf* relation where the dashed node notes the parent class. Other nodes and lines represent concrete classes and relations, respectively. We demonstrate the properties of our main concept `Dataset` in Table 2. The "Match" column denotes the `skos:exactMatch` from the corresponding source except for both *license* and *accessRights*, they represent `skos:closeMatch` due to a range mismatch between our properties and those defined in Dublin Core.

3.4. Embeddings Generation

In this section, we explain embeddings sources and methods we developed to transform the keywords into embedding space.

Embedding Sources We supported two variations of embeddings. On the one hand, for domain-specific embeddings, we trained a fasttext [10] model on the Seen data by converting the key-value pairs, "flat dictionary" (see Section 3.2), into synthetic sentences. Iteratively, we used both the key and its value in such a dictionary to create the corresponding sentence. On the

[11] https://rdflib.readthedocs.io/en/stable/
[12] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

Table 1
Auto-generated keys examples, frequencies, and the selected key name.

Auto-generated	Frequency	Selected
versionID	19	version
version	49	version
Short_Abstract	2	abstract
Abstract.Abstract	18	abstract
abstract	362	abstract
DOI	15	doi
doi	2	doi
contact.phone	8	contactPerson_phone
contacts.contactPerson.phone	12	contactPerson_phone
coverage .geographicCoverage .boundingCoordinates .eastBoundingCoordinate	57	geographicCoverage _boundingCoordinates _eastBoundingCoordinate
SpaceBoundingBoxes .BoundingBox .eastBoundingCoordinate	250	geographicCoverage _boundingCoordinates _eastBoundingCoordinate

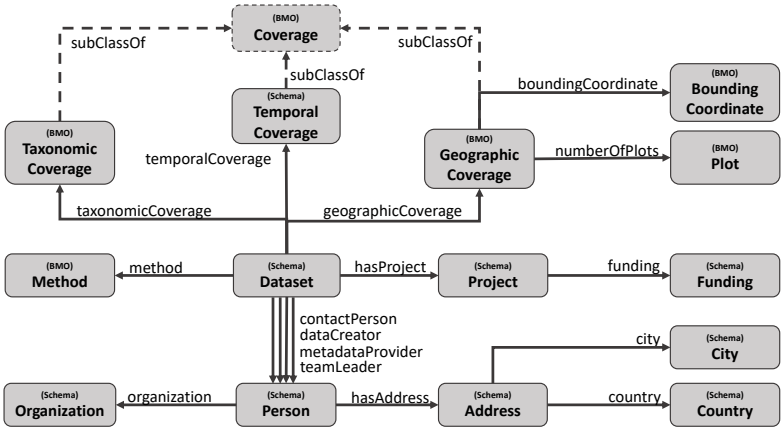


Figure 3: Biodiversity Metadata Ontology (BMO) concepts and relations.

other hand, for the pre-trained embeddings[13], we used the publicly available Wikipedia-based embeddings. We used these resources to generate both ontological embeddings (*BMOE*) and metadata embeddings (*MetaE*) for the unseen data. We compare both embedding sources during our experiments.

Generation Method The selected repositories use different keywords representing precisely the same thing. For example, BEF-China, GBIF, and BExIS use `geographicCoverage`, and DRYAD uses only `Coverage` to describe the geographical specs of a study. The same applies for `taxonomicCoverage` that BEF-China, GBIF, and BExIS use, whereas `Taxonomic_Scope` and `TaxonCoverage` are used by iDiv and PANGAEA, respectively. Thus, we used the list

[13] https://fasttext.cc/docs/en/english-vectors.html

Table 2
Properties of our main concept: `Dataset`. Short forms; SCH, DCT, and WD map to schema.org, Dublin Core Terms, and Wikidata respectively.

Type	Property	Match	Source	Meaning
datatype	title	name	SCH	The title of the dataset
	abstract	abstract	SCH	The short text that summarizes a dataset
	description	description	SCH	The summary of the dataset
	language	inLanguage	SCH	The language of the data provided
	intellectualRights	*accessRights*	DCT	Specify if the data is public or private
	license	*license*	DCT	The license of the dataset
	citation	citation	SCH	How to cite the dataset
	dataFormat		BMO	The dataset format, e.g., delimiter
	version	version	SCH	The version of the dataset
	keywordsSet	keywords	SCH	Dataset tags
	doi	P356	WD	Dataset DOI
	alternateIdentifier	identifier	SCH	Dataset download or home URL
	publicationDate	datePublished	SCH	When the dataset is published
	numberOfRecords	P4876	WD	How many records in the dataset
object	contactPerson		BMO	The contact person of the dataset
	metadataProvider		BMO	Who provided the metadata
	dataCreator		BMO	Who created the data
	project		BMO	The associated project of the dataset
	geographicCoverage		BMO	The geo specs of the study
	temporalCoverage		BMO	The time duration of the study
	taxonomicCoverage		BMO	The included taxons of the study

of synonyms for each selected key that was created during the BMO development. We aim to obtain embedding vectors of BMO relations that encode information from synonyms. For example, a vector for "version", represents the version of the dataset, would be a function of all its synonyms: "version", "versionID".

We developed two methods for approaching such an idea: 1) *Mean*: An embedding vector of a given key e_{key} is determined as the mean vector of all its synonyms set as defined as SE in Equation 1. 2) *Weighted Mean*: Similar to the Mean method and inspired from TF-IDF[14], we gave higher weight to the more specific words that form an entire key. For example, `temporalCoverage.startDate`, `startDate` would have double the weight of `temporalCoverage`. `temporalCoverage` is a less discriminative word since it would appear with another term like `endDate`. This method is described in Equation 2 where es_{ij} is the individual word vector of a given key of synonyms set SE, and we use the word position j as its weight. We use the embeddings generation methods to transform BMO ontology and the Unseen data keys into the embeddings space.

[14]https://en.wikipedia.org/wiki/Tf-idf

$$e_{key} = \frac{\sum_{i=1}^{|SE|} \sum_{j=1}^{|Words|} es_{ij}}{|SE|} \tag{1}$$

$$e_{key} = \frac{\sum_{i=1}^{|SE|} \sum_{j=1}^{|Words|} es_{ij} \times j}{|SE|} \tag{2}$$

3.5. Match

In this phase, we converted BMO into embeddings space using the equations above yielded *BMOE*. We also performed the same pre-processing procedure to obtain clean keys of the Unseen data. Then, we transformed the Unseen data keys into vector space as well yielded *MetaE*. One significant difference between this step and generating BMO embeddings is that the Unseen data have no synonyms; however, the mean-based operations are only done on the words of the key only. For matching, we used cosine similarity in the embedding space between the ontological embeddings, *BMOE*, and Unseen metadata embeddings, *MetaE*. For each *MetaE*, we retrieve the closest BMO vector that has $\geq 70\%$ similarity. We avoid the closest assignment for better recall. We chose such a threshold to balance the precision and recall. We tried higher thresholds; however, it misses a lot of true matches. This makes sense since the target ontological embeddings are created using a mean or weighted mean operation; thus, a 100% similarity will never be achieved. This step matches the unseen data to the ontology concepts and properties; however, it lacks the instances.

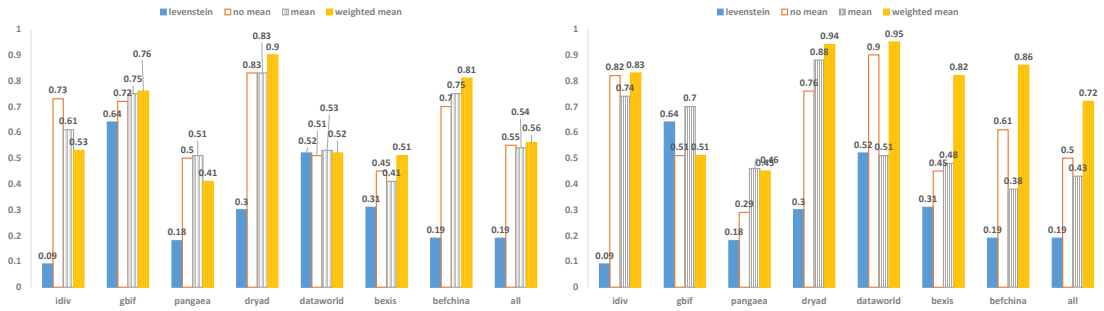
3.6. Validate & Populate

To populate the BMO with instances, we rely on the "flat dictionary". In that sense, the key has mapped to, e.g., ontology property, and its value represents the instance we add to the ontology. Auto-populating such ontology given only matches from the step above is not accurate for two reasons: 1) invalid entries in the metadata fields, and 2) miss-classification that yields datatype violations. We allow the population of a triple if and only if its value has the expected datatype. For example, we populate `dataCreator_Phone` if the corresponding value is a phone. We cover basic datatype validations using regular expressions for the following datatypes: *Phone*, *Email*, *Coordinate*, *URL*, *Decimal*, and *Date*. In addition, we validate the resultant KG using the W3C RDF Validation Service[15].

3.7. Release

Resources should be easily accessible to allow replication and reuse. We follow the FAIR (Findable, Accessible, Interoperable, and Reusable) guidelines [7] to release our contributions. We release our ground truth [20], ontological embeddings [21], BMO [22], and BMKG [23] in Turtle, N-triples, and RDF-XML format in Zenodo, so researchers in the community can benefit from them. We published our resources and code under the Creative Commons Attribution 4.0 International (CC BY 4.0) and Apache License 2.0, respectively.

[15] https://www.w3.org/RDF/Validator/



(a) Our Embeddings (b) Wiki Embeddings

Figure 4: Matching F1-score on Unseen data using our and Wiki-based embeddings.

4. Results

We conducted several experiments to demonstrate the effectiveness of the generated embeddings. Besides the two mean-based methods (mean and weighted mean) for embeddings generation, we developed a two baseline approaches. On one hand, to test the effectiveness of embeddings we developed a base line approach based on string similarity using Levenstein distance (levenstein). On other hand, to test the effectiveness of mean operations, we handled each key in the Unseen data as a single word without any kind of spliting, then, transform that word to a vector (key to vec). For evaluation, we manually annotated the cleaned Unseen Data with the correct match from the ontology. We use such ground truth to evaluate our matching technique. We considered the value of the auto-generated key to classify it. In the following, we show our matching results and give insights about the resultant auto-generated KG.

4.1. Matching Results

Since we have two embedding sources (our custom embeddings and pre-trained Wikipedia-based embeddings) and three techniques (baseline (no mean), mean, and weighted mean) to obtain an embedding vector with an additional lexical baseline (levenstein), we conducted seven experiments to cover all combinations. Figure 4 shows the F1-score for all experimental settings. We calculated the scores per data repository and the accumulated them as well (all). We found that the Weighted Mean approach combined with the pre-trained Wikipedia-based embeddings yielded the best scores. This proves that our developed mean-based method successfully captured a wide range of syntactic representations from metadata keywords. Since we used synthetic sentences that are derived from a combination of metadata key and value, “flat dictionary” items combined, we lacked proper natural text during the training. Thus, it justifies the lower scores with our custom embeddings. From the repositories perspective, our approach gained the lowest scores on PANGAEA due to the lack of proper metadata fields, thus, confusing our matching procedure. However, our method reaches, at some times, 100% precision on Dryad due to its relatively more straightforward fields to match, e.g., “title”.

4.2. Resultant Knowledge Graph

Our resultant BMKG represents BMO with instances. It contains those instances from the Unseen data. Figure 5 represents the frequency of triples in the BMKG. Darker colors depict higher field frequency. `Dataset` datatype properties, e.g., keywordSet, citation, and description, are the most occurred fields in the graph from the Unseen metadata files. They are auto-populated correctly with valid instances. The data properties are followed by the `DataCreator` and `ContactPerson`. The `NumberOfPlots` seems to be more frequently used than `BoundingCoordinates` under `GeographicCoverage`. The `MetadataProvider` is frequently incomplete compared to both `DataCreator` and `ContactPerson` since it is usually described by `givenName` and `phone` only.

We gave a closer look to the BMKG where we manually investigated the populated `Dataset` instances using Protégé[16]. Figure 6 shows a snippet of the automatically generated KG. We picked a random instance of the core concept "Dataset" and investigated the auto-populated triples manually using Protégé with its original corresponding metadata file. In the figure, "numberOfRecords", "temporalCoverage", "title", "geographicCoverage", "dataCreator" are correctly matched and populated (green rectangles 1, 4, 5, 6, 7). However, our technique mismatches the "Project" and the "startDate" under the "TemporalCoverage" triple (red rectangles 2 and 3). From the original file, the former is just a "description", and the latter should be "endDate". In addition, we identified missing triples under the "dataCreator", e.g., phone value. This means that our validation layer failed to validate a phone value.

5. Discussion & Limitations

In this section, we give the first answers to our previously expressed research questions. ***RQ1****: Is it possible to construct a KG using metadata as the only data source?* our conducted experiments in this paper show that metadata are a promising data source since we managed to create a KG from them in a fully automated way. However, the resultant KG suffers from quality issues as shown in Figure 6. This needs multiples revisions and human intervention to ensure a higher quality level of the restultant KG. In this paper, we developed a fully automated unsupervised approach based on embeddings to transform raw metadata files into an ontology and populate it with instances to generate a final KG. Thus, our approach initially answers ***RQ2****: How can we automate the transformation of metadata to KG?*. We pose the last research question, ***RQ3****: What are the challenges facing such automated transformation?* to discuss the common obstacles with our provided solutions as follows: 1) A resultant triple might violate datatype constraints due to a mismatch by our approach or originally filed with a wrong datatype. We proposed validations that are based on regular expressions for several datatypes. E.g., a triple like: *(dataCreator, phone, X)* is considered valid if and only if *X* is a valid *phone* value. 2) Inconsistent value format of metadata attributes. *Keywords* are used either in a word-by-word form or a list separated by a delimiter like commas and semicolons. We set the granularity to a word level for consistency, thus, we split any given list by its delimiter. 3) Embeddings failed to differentiate between values like *surName* and *givenName* since both are names. Thus, we consider the actual string value

[16] https://protege.stanford.edu/

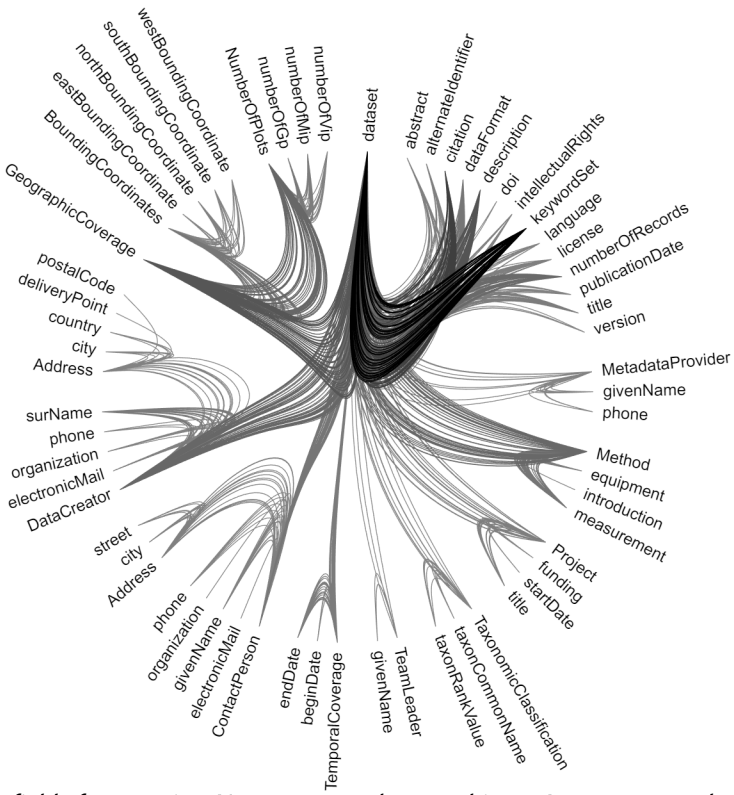


Figure 5: BMKG: fields frequencies. Upper case nodes are objects. Lower case nodes show literals.

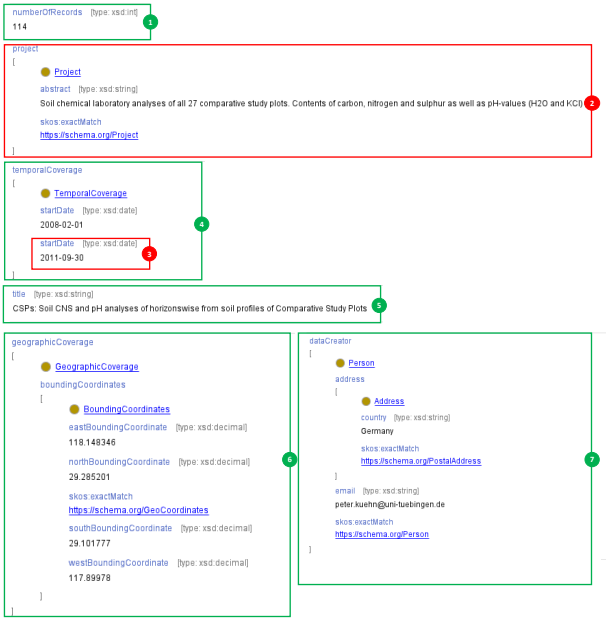


Figure 6: BMKG: Instance analysis. Green rectangles are correctly matched, red ones are otherwise.

to obtain the correct match for such cases. 4) Some repositories provide weak and incomplete metadata fields like PANGAEA. Such repositories introduce noise that we omit as much as

possible to generate a clean KG.

We list the following limitations that are not solved yet. In our future work, we will consider the sketched solutions: 1) We discovered more inconsistencies regarding some metadata fields. Currently, *license* and *intellectualRights* properties accept a literal as a range. However, Dublin Core defines both of them where the expected range is an actual "license" and "right statement" objects, respectively. We plan to change that to follow the Dublin Core definitions where we support entity linking. 2) Currently, *citation* is a data property that accepts a string as a range. We chose based on the options commonly used in the selected repositories. However, a typical citation contains more fine-grained data like authors, volume, and issue, which PANGAEA partially adopts. Thus, we consider a further analysis of the *citation* field by recognizing its individual parts. By this means, it would yield into more fine-grained KG and better description. 3) Metadata fields might contain several (semi)redundant information across various fields, e.g., BEF-China might have these duplicates under *description*, *abstract*, *introduction*, *measurement*. A semi-automated approach could overcome this issue. 4) We found complex fields that have multiple semantic concepts. E.g., the *description* that is used in data.world often contain information about *citation* or *license*. So, detecting those nested entities would yield concrete information.

6. Conclusions & Future Work

We investigated the construction of a Knowledge Graph (KG) using metadata as the only source of data. Our pipeline is tested on, but not limited to, a biodiversity domain use case. We demonstrated our used data repositories: seven biodiversity data portals. We manually collected the metadata files from them. We divide them into Seen and Unseen data. We used the Seen data to construct the underlying data model that aligns the selected data portals. In addition, we used them to transform the constructed ontology into the embedding space. We used the Unseen data to evaluate our unsupervised matching techniques and auto-populate the BMO with instances. Such embeddings-based techniques are based on the mean operation where the similarity measure is the cosine similarity. We demonstrated the effectiveness of the developed matching and population techniques. In addition, we showed the current limitations of the methodology, and we pointed out possible solutions for them. Besides the transformation pipeline, we presented the Biodiversity Metadata Ontology (BMO) and Biodiversity Metadata Knowledge Graph (BMKG) as byproducts of this work. We made our resources and code publicly under our GitHub repository. In addition, we released our ground truth [20], ontological embeddings [21], BMO [22], and BMKG [23] in Turtle, n-triples, and RDF-XML format in Zenodo.

We see multiple areas to extend this work. First, we plan to enhance our matching technique by using an ensemble-based method that relies on both embeddings and string similarity. In addition, we explore more options to close the open issues we have discussed. For example, we parse complex fields into more fine-grained pieces for better representation. Moreover, we explore triple verification approaches for more trusted KG. Finally, we expose the BMKG via a SPARQL endpoint to achieve better data re-usability.

Acknowledgment

The authors thank the Carl Zeiss Foundation for the financial support of the project "A Virtual Werkstatt for Digitization in the Sciences (P5)" within the scope of the program line "Breakthroughs: Exploring

Intelligent Systems" for "Digitization - explore the basics, use applications".

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, ACM Comput. Surv. 54 (2022) 71:1–71:37.

[2] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, number 22 in Synthesis Lectures on Data, Semantics, and Knowledge, Morgan & Claypool, 2021. doi:10.2200/S01125ED1V01Y202109DSK022.

[3] S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, M. Vidal, Towards a Knowledge Graph for Science, in: Proceedings of the 8th International Conference on Web Intelligence, Mining and Semantics, WIMS 2018, Novi Sad, Serbia, June 25-27, 2018, ACM, 2018, pp. 1:1–1:6. doi:10.1145/3227609.3227689.

[4] L. M. R. Gadelha, et al., A survey of biodiversity informatics: Concepts, practices, and challenges, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 11 (2021). doi:10.1002/widm.1394.

[5] F. Löffler, V. Wesp, B. König-Ries, F. Klan, Dataset search in biodiversity research: Do metadata in data repositories reflect scholarly information needs?, PloS one 16 (2021) e0246099. doi:10.1371/journal.pone.0246099.

[6] R. Page, Towards a biodiversity knowledge graph, Research Ideas and Outcomes 2 (2016) e8767. doi:10.3897/rio.2.e8767.

[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 (2016). doi:10.1038/sdata.2016.18.

[8] Y. Goldberg, O. Levy, Word2Vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method, CoRR abs/1402.3722 (2014). URL: http://arxiv.org/abs/1402.3722.

[9] J. Pennington, R. Socher, C. D. Manning, Glove: Global Vectors for Word Representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL, 2014, pp. 1532–1543. doi:10.3115/v1/d14-1162.

[10] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguistics 5 (2017) 135–146. doi:10.1162/tacl_a_00051.

[11] N. Abdelmageed, B. König-Ries, Meta2KG: transforming metadata to knowledge graphs, in: Proceedings of the 17th International Workshop on Ontology Matching (OM 2022) co-located with the 21th International Semantic Web Conference (ISWC 2022), Hangzhou, China, held as a virtual conference, October 23, 2022, volume 3324 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 226–228. URL: http://ceur-ws.org/Vol-3324/om2022_poster3.pdf.

[12] A. Sadeghi, C. Lange, M. Vidal, S. Auer, Integration of scholarly communication metadata using knowledge graphs, in: Research and Advanced Technology for Digital Libraries - 21st International Conference on Theory and Practice of Digital Libraries, TPDL 2017, Thessaloniki, Greece, September 18-21, 2017, Proceedings, volume 10450 of *Lecture Notes in Computer Science*, Springer, 2017, pp. 328–341. doi:10.1007/978-3-319-67008-9_26.

[13] S. Fathalla, C. Lange, EVENTS: A dataset on the history of top-prestigious events in five computer science communities, in: A. N. González-Beltrán, F. Osborne, S. Peroni, S. Vahdati (Eds.), Semantics, Analytics, Visualization - 3rd International Workshop, SAVE-SD 2017, Perth, Australia, April 3, 2017, and 4th International Workshop, SAVE-SD 2018, Lyon, France, April 24, 2018, Revised Selected Papers, volume 10959 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 110–120. doi:10.1007/978-3-030-01379-0_8.

[14] S. Fathalla, C. Lange, EVENTSKG: A knowledge graph representation for top-prestigious computer science events metadata, in: N. T. Nguyen, E. Pimenidis, Z. Khan, B. Trawinski (Eds.), Computational Collective Intelligence - 10th International Conference, ICCCI 2018, Bristol, UK, September 5-7, 2018, Proceedings, Part I, volume 11055 of *Lecture Notes in Computer Science*, Springer, 2018, pp. 53–63. doi:10.1007/978-3-319-98443-8_6.

[15] S. Fathalla, C. Lange, S. Auer, EVENTSKG: A 5-star dataset of top-ranked events in eight computer science communities, in: P. Hitzler, M. Fernández, K. Janowicz, A. Zaveri, A. J. G. Gray, V. López, A. Haller, K. Hammar (Eds.), The Semantic Web - 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2-6, 2019, Proceedings, volume 11503 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 427–442. doi:10.1007/978-3-030-21348-0_28.

[16] S. Fathalla, S. Vahdati, S. Auer, C. Lange, The scientific events ontology of the openresearch.org curation platform, in: C. Hung, G. A. Papadopoulos (Eds.), Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, SAC 2019, Limassol, Cyprus, April 8-12, 2019, ACM, 2019, pp. 2311–2313. doi:10.1145/3297280.3297631.

[17] M. Schröder, C. Jilek, A. Dengel, A human-in-the-loop approach for personal knowledge graph construction from file names, in: Proceedings of the 3rd International Workshop on Knowledge Graph Construction (KGCW 2022) co-located with 19th Extended Semantic Web Conference (ESWC 2022), Hersonissos, Greek, May 30, 2022, volume 3141 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022.

[18] N. Abdelmageed, A. Algergawy, S. Samuel, B. König-Ries, BiodivOnto: Towards a Core Ontology for Biodiversity, in: The Semantic Web: ESWC 2021 Satellite Events, volume 12739, Springer, 2021, pp. 3–8. doi:10.1007/978-3-030-80418-3_1.

[19] D. Vrandecic, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. doi:10.1145/2629489.

[20] N. Abdelmageed, B. König-Ries, Biodiversity Metadata Ground Truth, 2022. doi:10.5281/zenodo.6951623, version: v1.0.0, Zenodo.

[21] N. Abdelmageed, B. König-Ries, Biodiversity Metadata Ontology Embeddings (BMOE), 2022. doi:10.5281/zenodo.6951658, version: v1.0.0, Zenodo.

[22] N. Abdelmageed, B. König-Ries, Biodiversity Metadata Ontology (BMO), 2022. doi:10.5281/zenodo.6948519, version: v1.0.0, Zenodo.

[23] N. Abdelmageed, B. König-Ries, Biodiversity Metadata Knowledge Graph (BMKG), 2022. doi:10.5281/zenodo.6948573.