Semantic Web
Vol. 17(1) 1–24
© The Author(s) 2025
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/09217134251389825
journals.sagepub.com/home/swj

Analysis of the Performance of Representation Learning Methods for Entity Alignment: Benchmark Versus Real-World Data

Ensiyeh Raoufi[1], Bill Gates Happi Happi[2], Pierre Larmande[2], François Scharffe[1] and Konstantin Todorov[1]

Abstract

Representation learning for entity alignment (EA) aims to identify entities in different knowledge graphs (KGs) that refer to the same real-world object by comparing their embedding similarity. Although many EA models perform well on synthetic benchmark datasets, this performance does not always transfer to real-world, incomplete, and domain-specific data. A systematic comparison between synthetic benchmarks and original heterogeneous datasets is still limited. Many EA models also restrict the alignment search space to validation entities, limiting coverage of real KG content. Within this setting, our results show that embedding-based EA models continue to face generalization challenges in realistic large-scale KG search spaces. We evaluate several competitive EA models-commonly tested on benchmarks such as DBP15K-on multiple real-world heterogeneous datasets. The experiments reveal a performance decrease when moving beyond synthetic benchmarks, indicating that current models do not fully capture the characteristics of real data. We also analyze semantic similarity and profiling features of the datasets to help explain these differences. This study outlines practical limitations of embedding-based EA methods and provides insights for developing approaches that better handle the variability and complexity found in real-world KG alignment tasks.

Keywords

entity alignment, knowledge graphs, representation learning, knowledge graph heterogeneity, EA benchmarks

Received: February 19, 2024; accepted: September 3, 2025

Editor: Guest Editors OM-ML 2024
Solicited reviews: Vasilis Efthymiou, Assistant Professor, Department of Informatics and Telematics of Harokopio University of Athens, Tavros, Greece; Three anonymous reviews

[1]LIRMM, University of Montpellier, CNRS, France
[2]DIADE, IRD, CIRAD, University of Montpellier, France

Corresponding Author:
Ensiyeh Raoufi, LIRMM, University of Montpellier, CNRS, France.
Email: ensiyeh.raoufi@umontpellier.fr

1 Introduction

Knowledge graphs (KGs) have emerged as powerful tools for a range of applications, including information retrieval, question answering, and data federation (Ji et al., 2021). An entity in a KG refers to a distinct and identifiable concept, which can be, for example, a concrete object, an abstract idea, or an event. Entities are represented as nodes, forming the building blocks of the graph. The relationships between entities are modeled as edges, serving as connections that establish associations or interactions between the nodes. These edges convey various meanings, representing attribute properties or relation properties of the entities. Examples of attribute properties include "birth date," "genre," or "description," and examples of relation properties include "located in," "established by," or "worked with." Based on the given definition (Person1, birth date, "1989-09-30") indicates an attribute triple, while (City1, located in, Country1) indicates a relation triple.[1] Essentially, the combination of entities and their interconnecting relationships forms a structured representation of knowledge. Hence, KGs are designed in a way that facilitates storage, access, semantic understanding of data and reasoning over it and are widely used in a variety of domains, including the Semantic Web in general (Bonatti et al., 2019; Ryen et al., 2022; Villazón-Terrazas et al., 2020), cultural heritage (Achichi et al., 2018; Carriero et al., 2019; Dou et al., 2018; Marchand et al., 2020), biomedicine (Ernst et al., 2015; Nicholson & Greene, 2020; Sanou et al., 2022; Unni et al., 2022), sociology (Cao et al., 2020; Tchechmedjiev et al., 2019; Wang, Chen, et al., 2018), and data-driven industries (Bader et al., 2020; Kejriwal et al., 2019).

As data comes from different sources, it is often scattered across multiple KGs, even if it conveys the same information, leading to various challenges. One such challenge is identifying and matching entities from a source KG to their equivalents in a target KG that represent the same real-world object (Ferrara et al., 2011)—a task known as entity alignment (EA). EA in turn facilitates data integration, information retrieval, and entity disambiguation across diverse knowledge sources (Achichi et al., 2019; Beretta et al., 2020; Saha et al., 2018; Zou, 2020).

We start by providing some *terminology*. We use the term KG heterogeneity in the sense of Achichi et al. (2019), common in the fields of ontology and EA. In a nutshell, it concerns any difference in the expression of a given piece of knowledge across two KGs (be it structural, syntactical, terminological, or other). Sticking to the data heterogeneity concept provided in Achichi et al. (2019) and algebraic properties of the graphs, in this paper, we are interested in differences in value and structural levels of the KGs in each dataset. The pair of KGs in the datasets might differ in the graph's structural properties, such as size, degree distribution, etc. Also, each pair of aligned entities in two KGs may have descriptive and data quality heterogeneities, following Achichi et al. (2019).

By dataset, we mean an EA dataset, which consists of a pair of KGs: a source and a target KG to be interlinked, together with a reference alignment that helps evaluate or train the models. A reference (or seed) alignment is a manually curated set of correspondences or alignments (often together with a confidence score) between entities across the two different KGs. By unmatchable entities, we mean pairs of entities from the source and target KGs that are not to be aligned (i.e., they refer to different real-world entities).

We distinguish two main types of datasets. A synthetic benchmark dataset refers to a dataset that consists mostly of KGs sampled from larger ones, following a motivation of having smaller KGs that mimic certain characteristics of the real large KGs. Additionally, in machine learning applications, benchmark datasets could potentially be fully synthetic, that is, entirely generated from scratch, not using a real KG as a starting point. Often in benchmark datasets the source entities are matched with their corresponding counterparts in the target KG under the 1-to-1 assumption (meaning that each source entity has exactly one match in the target graph).[2] In this paper, we use the terms "benchmark dataset" and "synthetic benchmark dataset" interchangeably. A real-world dataset, on the other hand, is one that is issued from a real-world scenario and contains the unchanged KGs that are not sub-sampled from larger KGs under some conditions, such as being sparse or dense, or retaining a similar degree distribution as the KGs they are sampled from.

For the purposes of this study, we categorize EA techniques into two main groups: embedding and non-embedding-based methods. Non-embedding-based approaches apply user-crafted representations of entities and relations and align the entities across the KGs based on similarity measures or logic axioms. This group of approaches prioritizes symbolic reasoning, logical inferences, and linking specifications defined by domain experts to guide the alignment process (Zeng et al., 2021). Embedding-based approaches, in contrast, automatically represent entities in a feature space and predict alignments based on similarity metrics over the learned embeddings. Embedding refers to representing an object as a vector in a continuous space based on a given number of constraints (e.g., close in meaning entities should have vectors that live close to one another in the embedding space; Zeng et al., 2021). Following this paradigm, embedding-based EA models commonly use an embedding and an alignment module. While the embedding module represents each KG entity as a vector in a low-dimensional embedding space, the alignment module ensures that aligned entities are close together in a unique embedding space or learns a mapping between KGs with respect to the reference alignment. During the model's training phase, through iterative interactions between the embedding and alignment modules, all entities from both KGs are embedded, and predictions are made regarding which entities are most likely to align.

Relying on non-embedding-based methods may be more suitable in scenarios dealing with sparse or small datasets, or in situations where there is not a high variety of heterogeneities, such as predicate, class, or graph linking problems (as defined by Salazar et al., 2023). Real-world datasets often do not meet these ideal conditions, and therefore, embedding-based EA methods promise more efficiency, as they are flexible in cross-lingual scenarios, scalable to large KGs, and globally consistent in representations across KGs.

This paper's focus is on analyzing specifically the embedding-based EA models, with respect precisely to both synthetic benchmark datasets and real-world datasets, as well as considering the different training and evaluation strategies and model types. Hence, we add to several recent surveys and studies that investigate that question from a critical viewpoint (Ardjani et al., 2015; Shvaiko & Euzenat, 2011; Zeng et al., 2021). For example, Bengio et al. (2013), Hamilton et al. (2017), and Sun et al. (2017) analyze the performance of embeddings-based EA models and compare them regarding their performance on benchmark datasets (Euzenat et al., 2013; Fanourakis, Efthymiou, Kotzinos, & Christophides, 2023; Huang et al., 2022; Jiang et al., 2023; Sun et al., 2020; Zeng et al., 2021; Zhang et al., 2022). Previous research has extracted more realistic EA benchmark datasets (Sun et al., 2020) from large knowledge bases such as DBpedia (Lehmann et al., 2015). We enhance the work of the studies cited above by expanding the benchmarks with datasets containing a low percentage of matchable entities to better reflect real-world scenarios, and evaluating the performance of two of the leading embedding models (RDGCN and BERT-INT) when we include all entities in the target KG as alignment candidates during the model evaluation. Furthermore, we include an in-depth discussion on the evaluation metrics that are commonly used for the EA task, building on preliminary remarks found in Leone et al. (2022). As an overarching question, we consider several recent EA embedding-based models having state-of-the-art performance on synthetic benchmark datasets and analyze their capacities when they deal with heterogeneous real-world data. We show a considerable drop in performance in the latter scenarios. To help understand this observation, on the one hand, we analyze and compare the real-world and synthetic benchmark datasets with respect to a set of dataset profiling features (Ben Ellefi et al., 2018) studied and applied for the EA task. On the other hand, we pair these observations with a look into the underlying nature of the embedding-based models. Todorov (2019) observes that cutting-edge EA models did not address the particular properties of data well because they prioritized genericity and automation. Indeed, the results of our study demonstrate that while embedding-based models perform well on certain synthetic benchmark datasets, they struggle in real-world scenarios due to insufficient consideration of the inherent characteristics and nature of the data. Finally, in order to be able to compare the embedding-based models to methods from the non-embeddings group, we include in our analyses the DLinker system (Happi Happi et al., 2022), for reasons explained in Section 3.

The main contributions of this analysis paper are:

- A novel look into and comparison of the frameworks of established EA methods having different embedding bases: we propose a novel categorization of embedding-based EA methods based on their embedding approaches.
- A comparison of the features of synthetic benchmark and real-world datasets from aspects related to EA: although it appears difficult to isolate a structure-related meta-feature which explains the performance of all methods on the different datasets (because each method embeds the structure from a different aspect), we find that the semantic similarity is the dataset meta-feature that correlates at the strongest with the performance of embedding-based EA methods.
- A discussion of the commonly used evaluation metrics for the EA task: we explain how Hit@1 is equivalent to precision and recall when under the 1-to-1 assumption in the validation set, and when and why each evaluation metric should be applied.
- An analysis of the performance drop of EA methods on real-world datasets in comparison to their performance on established synthetic benchmarks: we present shreds of evidence and probable reasons to explain the observed drop in performance; we go beyond the 1-to-1 assumption during the model evaluation and investigate the performance drop of the EA models using both Hit@1 and F_1-score.
- Analysis of the different categories of embedding models with respect to synthetic and real-world datasets: we find out interaction training models to be the best-performing category of EA methods on real-world large-scale data.

The paper is organized as follows: Section 2 summarizes related surveys and empirical studies on EA methods, positioning our work within that context. Section 3 outlines the key features of the embedding-based EA methods selected for performance analysis. In Section 4, we compare several benchmark and real-world datasets, highlighting the greater heterogeneity and complexity of the latter. Finally, Section 5 examines the performance of EA models on heterogeneous real-world data and explores the reasons for their performance decline on these datasets.

2 Related Work

This section provides an overview of surveys and related analytical studies that critically examine existing EA approaches, with a particular focus on embedding-based methods. In line with the scope of the paper, specific alignment methods are not discussed here.

Several studies have contributed to the understanding and advancement of KG embeddings (KGEs) and their applications, such as link prediction, KG completion and reasoning, and EA (Choudhary et al., 2021; Ji et al., 2021; Lu et al., 2020; Sharma & Talukdar, 2018; Wang et al., 2017). While Sharma and Talukdar (2018) reveal sharp differences in the geometry of embeddings produced by various KGE methods, Tran and Takasu (2019) introduce a multi-embedding interaction mechanism for analyzing KGE models such as DistMult (Zhang et al., 2019) and ComplEx (Trouillon et al., 2016). The latter study unifies and generalizes these models, offering an intuitive perspective for their effective use. Luo et al. (2022) introduce a scalable and open-source Python library for multisource KG embeddings. Supporting joint representation learning, it implements 26 KGE models and 16 benchmark datasets. Moreover, Cao et al. (2024) categorize the existing KGE models based on representation spaces and discusses whether they have algebraic, geometric, or analytical structures.

Several surveys and experimental studies have been conducted on methods for EA across KGs (Fanourakis, Efthymiou, Kotzinos, & Christophides, 2023; Zhao, Jia, et al., 2020; Zhao, Zeng, et al., 2020). Broadly, these studies categorize EA techniques into two main groups: embedding-based methods and traditional approaches (Sun et al., 2020; Zeng et al., 2021; Zhang et al., 2022). Traditional EA methods rely on user-defined rules, Web Ontology Language reasoning, and/or similarity computations based on symbolic features of entities. We refer to these as non-embedding-based methods.

Moving now the focus toward embedding-based methods, Sun et al. (2020) created an open-source toolkit, named OpenEA. The authors discuss the characteristics and functionalities of embedding-based methods, highlighting how they predict matching entities through nearest-neighbor searches among target entity embeddings. Two combination paradigms are outlined: one encoding KGs in independent spaces and learning a mapping using seed alignment, and another representing KGs in a unified space, considering highly similar embeddings for aligned entities. The study underscores the incorporation of entity relations and attribute properties into embedding modules to enhance accuracy, categorizing relation embeddings into triple-based, path-based, and neighborhood-based groups. Attribute embedding, achieved through correlation or literal methods, is also explored for improving entity similarity assessment. Fanourakis, Efthymiou, Kotzinos, and Christophides (2023) present the meta-features of the OpenEA datasets, which adhere to the 1-to-1 assumption, and explain the technical details of several embedding-based EA models. However, they do not include details regarding the generation of the dataset's meta-features, such as description similarity. In this work, we provide formulas to compute the extracted meta-features, and we analyze the performance of EA models on both benchmark and real-world datasets that do not follow the 1-to-1 assumption.

Zhang et al. (2022) analyze the performance of translational and graph neural network (GNN)-based EA methods with respect to the seed alignment and dataset sizes, the use (or not) of attribute triples, the presence of multilingual data, and the embedding size. They propose new benchmark datasets sampled from large-scale KGs such as Wikidata (Vrandečić & Krötzsch, 2014) and Freebase (Bollacker et al., 2008) that do not fulfill the 1-to-1 assumption (40% and 75% of the entities in every pair of KGs combined in the datasets do not have matches). The authors then tested several EA models on the newly sampled dataset. However, one of the issues with this approach is the fact that the 1-to-1 assumption is not a condition that only holds for the datasets, but also many EA models consider that constraint during the model evaluation. Hence, even though Zhang et al. generate new data including not only $1:1$ mappings, none of the non-matchable entities are considered during the evaluation for some of the models they applied, such as MultiKE (Zhang et al., 2019) and TransEdge (Sun et al., 2019), due to the nature of these models that only consider the alignment candidates from the reference alignment. Similarly, Jiang et al. (2023) evaluate the performance of EA methods on newly generated, highly heterogeneous KGs that differ in scale and structure, with fewer overlapping entities than benchmark datasets (sharing 60% and 25% of the entities). They introduced two more realistic datasets, ICEWS-WIKI and ICEWS-YAGO, which were derived from knowledge bases with significantly different degree distributions. Although these new datasets deviate from the typical 1-to-1 assumption, they tested EA methods such as graph convolutional network (GCN)-Align (Wang, Lv, et al. 2018), RDGCN, and FuAlign (Wang et al., 2023) on sub-sampled heterogeneous datasets, overlooking the fact that these methods do not account for non-matchable entities as candidates in the evaluation phase. In contrast to these studies and building on these observations, we discuss in detail this search-space-related issue in our experiments in Section 5.3.2.

Zeng et al. (2021) provide a brief overview of research in EA, covering traditional methods, knowledge representation learning, and alignment based on representation learning in KGs. They conducted their research only on one single dataset—DBP15K, which is a synthetic benchmark dataset holding the $1:1$ assumption. They tested only two categories of models: translational and GNN-based in the context of multilingualism. In contrast, our work covers a wider variety of both datasets and models that differ in nature.

Fanourakis, Efthymiou, Christophides, et al. (2023) explore indirect biases of EA methods due to structural diversity in the KGs and introduce a sampling algorithm to generate challenging benchmark datasets by changing the properties of the KGs. In that way, the authors assess EA methods' robustness against such diversity. Modifications include changing connectivity metrics such as "average node degree," "max component size," and "ratio of weakly connected components" to control the level of structural heterogeneity of the generated datasets. In our work, we do not use a sampling algorithm; instead, we experiment with EA methods having different design bases on widely used benchmark datasets and real-world datasets.

Leone et al. (2022), provide a discussion on the evaluation metrics for EA for datasets that do not follow the 1-to-1 assumption. To go beyond this assumption, the authors generate sub-sampled datasets whose KG sizes are different, where each dataset variant includes about 30% unmatchable entities. In comparison to Leone's study, in addition to the fact that our real-world datasets are original and not obtained by sampling, the proportion of unmatchable entities for each of our real-world datasets is more than 80% (KGs in Doremus and AgroLD on average have more than 87% and 82% unmatchable entities, respectively). Furthermore, we report the Hit@k measures for the case that a 1-to-1 assumption does not hold on our datasets to compare the model's performance with the one reported in previous studies.

To sum up, our work builds on previous research by extending and refining key aspects of EA evaluation. In particular, we study the performance of embedding-based EA methods with distinct representation learning principles on real-world and benchmark datasets. In that, our study stands out for its attention to data quality considerations (Ben Ellefi et al., 2018). While prior studies provide valuable insights into meta-features and sampling methods, we advance this by offering explicit formulas for meta-feature extraction and testing EA models on both benchmark and real-world datasets that do not follow the 1-to-1 assumption. Instead of generating sampled datasets, we focus on real-world original data with over 80% unmatchable entities, providing a more rigorous evaluation. In this way, our work continues and enhances existing research, bringing new perspectives to real-world EA challenges.

3 Methods for EA via Representation Learning

Certain studies categorize embedding-based methods according to their use of semantic information to represent the KGs (Wang et al., 2023), while others categorize them according to whether they use attribute or relation predicates for embedding learning, their alignment modules (i.e., whether they embed both KGs in the same space or separately), or their learning strategy (supervised, semi-supervised, or unsupervised; Fanourakis, Efthymiou, Kotzinos, & Christophides, 2023; Sun et al., 2020). Based on recent studies (Fanourakis, Efthymiou, Kotzinos, & Christophides, 2023; Jiang et al., 2023; Sun et al., 2020; Wang et al., 2023; Zeng et al., 2021; Zhang et al., 2022) and our analysis, we propose to classify the embedding-based EA models into four groups: (1) translational, (2) GNN-based, (3) graph transformers (GTs)-based, and (4) interaction training models.

Several EA models, such as MTransE (Chen et al., 2017) and IPTransE (Zhu et al., 2017) have been designed by using translational techniques such as TransE (Bordes et al., 2013) for KGE and EA across KGs (Sun et al., 2018). A KG is usually represented as a directed graph, in which nodes refer to entities and edges refer to relations between entities, or simply by a set of triples of the type ⟨*head entity, relation, tail entity*⟩. The translational model's framework embeds a relation predicate as a translation vector from a head to a tail entity. GNN-based methods, such as GCN-Align (Wang, Lv, et al., 2018), RREA (Mao et al., 2020), and GMNN (Xu et al., 2019a) employ GNNs (Scarselli et al., 2008; Wu et al., 2021; Zhou et al., 2020) to represent the graphs and link them. These models rely on the signature GNN message-passing system to integrate the information of each entity's neighbors. Inspired by the successful application of the Transformer (Vaswani et al., 2017) model in representing sequences for the automatic translation task (Lin et al., 2022), several works developed and applied the GTs model for representing graphs and their entities (Dwivedi & Bresson, 2020; Li et al., 2018; Müller et al., 2023; Nguyen et al., 2022; J. Zhang et al., 2020). Recently, models built on top of Transformers have been developed specifically for the EA task (Cai et al., 2022; Trisedya et al., 2023), adopting the self-attention mechanism from Transformers to represent entities. As a point of comparison between GNNs and Transformers, we underline that Transformers use multi-head attention, treating the entire sequence as a local neighborhood, whereas standard GNNs aggregate features from local nodes (Hussain et al., 2022). Applying Transformer architecture to GNNs, as in the GTs approaches, is motivated by the need to overcome the issue of information dispersion between distant elements in structural data (Chen et al., 2022). GTs address the limitations of traditional GNNs by leveraging Transformers' ability to capture long-term dependencies. By integrating GNNs and Transformers, GTs expand the receptive field of GNNs, effectively utilize graph structure information, and establish a collaborative framework where each module reinforces the other's strengths (Min et al., 2022).

The final group of models we introduce (and that prior works categorize as "others"; Jiang et al., 2023) has a common important characteristic: learning the embeddings of the two KGs simultaneously (Tang et al., 2020; Wang et al., 2023;

Yang et al., 2020; Zeng et al., 2020). We refer to this group as the interaction training group. Unlike other methods that embed entire KGs independently and then align entities, interaction training models do not need to embed entire KGs, which makes their inference more adaptable to unseen data. Instead, these models embed pairs of entities from both source and target KGs, simultaneously capturing interactions between the entities. This is done by comparing the entities' features—using techniques such as aggregation or averaging—to generate interaction vectors, which are then embedded through neural networks or similar techniques. The final predictions are based on a distance margin or threshold. If the interaction embedding of a pair of entities belonging to the source and target KGs is measured to be above the threshold (measured using the vector norms), then the entities are aligned together. The aim is to keep a marginal distance between aligned entities with non-aligned ones. Interaction training methods might use translational or GNN or any other basic model to initially embed the entities, but in contrast to the three other groups, these models can provide insights into the correlation between the features of entities belonging to two KGs, whether they are aligned or not.

After analyzing many comparative studies on benchmark datasets, we decided to focus on this study on the following recently proposed embedding-based EA methods, representative of each of the four groups outlined above: MultiKE (translational model; Zhang et al., 2019), RDGCN (GNN-based model; Wu et al., 2019), i-Align (GT-based model; Trisedya et al., 2023), and BERT-INT (interaction training model). We choose these established models because they are scalable to run on real-world large KGs and have state-of-the-art performance on well-known benchmark datasets (Jiang et al., 2023). We give more details on each of them in the following.

MultiKE considers the two distinct KGs to be aligned as one large KG. To connect these two KGs and augment the number of relation triples, the method connects each entity in the source KG to the neighbors of its counterpart entity in the target KG and vice versa by replacing the head and tail entities of each relation triple with their counterparts in the reference alignment. To further enhance the relation triples, the method identifies matching relation and attribute predicates by comparing their literal or relation embeddings and selecting those that exceed a similarity threshold. Once the predicates are matched across the KGs, each relation is replaced by its counterpart, augmenting the relation triples accordingly. Then, it represents each entity and relation using a variant of TransE. To generate the final EA predictions, the model combines these representations with encoded local names of entities and predicates, which are then fed into a convolutional neural network (Zhang et al., 2019).

BERT-INT begins by generating initial entity embeddings using a pre-trained BERT-based model, leveraging the entities' descriptions or names/labels. It then constructs a similarity matrix based on the initial embeddings for each pair of training entities. Next, the method creates a neighborhood similarity matrix to co-train each entity pair in the candidate set. For training the interactions of the KGs' structural embeddings, BERT-INT relies exclusively on the direct neighbors of the entities. It is worth noticing that BERT-INT computes interactions between the attributes of entities being compared, rather than simply aggregating attribute information. This approach can reduce the impact of noisy or irrelevant attribute matches. The attribute–view interactions are processed in a unified way along with name/description and neighbor interactions, contributing to the overall alignment decision. It then aggregates all the vectors obtained by the similarity matrices to represent each pair of entities and finalizes the entity pair representations using a multi-layer perceptron.

RDGCN leverages the information of relations into entity representations employing a two-step process. First, a dual relation graph is constructed based on the input KG (the context graph), which is nothing but the line graph of the context graph.[3] In the dual graph, each node represents a type of relation, and two nodes are connected together if they have a common head or tail in the main KG. Then, a graph attention mechanism is applied to arouse reciprocal actions between the two graphs. The resulting vertex representations in the context graph are fed to GCNs (Kipf & Welling, 2017) layers to capture the graph's structural information through a message-passing system. In the last step, the obtained entity representations are used for aligning pairs of entities.

i-Align uses two transformer-based architectures to represent the entities based on their graph structures and textual attribute values. The model uses a graph encoder to aggregate the entities' structural information which can also effectively handle large KGs. The model's other transformer obtains the interconnection between the entity attributes using the embeddings of attribute keys and values as inputs. i-Align provides explanations of the alignment results in the form of a set of the most influential attribute predicates and entity neighbors based on the attention weights of its two transformers.

We summarize the main properties of these four methods in Table 1, including their respective results in terms of Hit@1 measure as reported in the original papers introducing these methods. As we can see, all methods report a Hit@1 of more than 88%, exceeding competing methods in the respective studies. MultiKE and i-Align have been evaluated on the DBP_WD_100K (Sun et al., 2018) and DBP_YG_15K (Zhang et al., 2022) benchmark datasets, respectively, while BERT-INT and RDGCN—on the DBP15K (Sun et al., 2017) dataset. All four methods use entity names for embedding as an extra; i-Align utilizes the attribute predicate's names and values in its embedding procedure. To use the maximum descriptive information of entities, BERT-INT employs the entity's descriptions instead of their names when such descriptions exist.

Table 1. Comparison of Embedding-Based EA Methods.

Method	KG Embedding Approach	Best-Evaluated Benchmark Dataset	Hit@1 on Benchmark Dataset	Input Features: Relation Predicate	Input Features: Attribute Predicate	Input Features: Entity Name
MultiKE	Translational	DBP_WD_100K	0.918	Relation name	Attr name/value	Entity name
RDGCN	GNN	DBP15K	0.886 (on FR–EN)	–	–	Entity name
i-Align	Graph Transformer	DBP_YG_15K	0.912	–	Attr name/value	Entity name
BERT-INT	Interaction training	DBP15K	0.992 (on FR–EN)	Relation name	Attr name/value	Entity name/ description

Note. EA = entity alignment; KG = knowledge graph; GNN=graph neural network; FR–EN = French–English.

Finally, to be able to compare between non-embedding-based and embedding-based methods, we include in our analyses DLinker (Happi Happi et al., 2022) as a representative method of the non-embedding-based group. The method applies an average aggregation between the similarity measures derived from the instance objects calculated by the longest common subsequence algorithm. DLinker has a performance that is close to that of the best-performing system LogMap (Jiménez-Ruiz & Cuenca Grau, 2011), on several OAEI[4] entity linking tracks. Furthermore, because it is developed in this paper's authors team, having full control of the tool facilitates further experiments.

In the sequel, we move on to describing and comparing the datasets that we consider in this study, specifically in terms of their different heterogeneity aspects.

4 Datasets and Their Heterogeneity Aspects

In this section, after giving a summary of the datasets we consider and motivating their choice, we study the degrees of their heterogeneities using specific metrics, introduced below. The study showed these datasets to be diverse and highly heterogeneous. Hence, we believe the analysis of the performance of the four chosen models (described above) on this particular collection of datasets would give us adequate insights beyond the specific choice of datasets and models, and in particular, for a better understanding of the challenges for the EA task when dealing with real-world, highly diverse datasets.

4.1 Datasets

We proceed to present and analyze the chosen datasets, coming from the two groups identified in the introduction: synthetic benchmark datasets and real-world datasets.

Benchmark Datasets. We consider DBP15K (Tang et al., 2020), which is a benchmark dataset that a significant number of state-of-the-art methods report their results on (Berrendorf et al., 2020; Zeng et al., 2021). DBP15K consists of three pairs of KGs that differ in the used language (French, Japanese, and Chinese). We pick the French–English dataset ($\text{DBP15K}_{\text{FR–EN}}$) as a sample of the whole. Furthermore, we consider SPIMBENCH,[5] the OAEI reference instance matching dataset, which is much smaller than DBP15K but similar to it in several heterogeneity aspects that we will discuss below. Additionally, we consider the ICEWS-WIKI and ICEWS-YAGO proposed in Jiang et al. (2023). They have the particularity of having been generated from KGs having highly different degree distributions, hence mimicking graphs from real-world scenarios. Hence, these two datasets are highly heterogeneous in structure as compared to $\text{DBP15K}_{\text{FR–EN}}$ (for full comparison, we refer to Jiang et al., 2023), where the source and target KGs in each of these two datasets have very different scales.

Real-World Datasets. Because heterogeneity of KGs has a broader meaning than linguistic differences (Achichi et al., 2019) and also, benchmarks often present idealized scenarios with a limited set of relationships, controlled noise, and specific characteristics (Fundulaki & Ngomo, 2016), we added to our investigation two real-world datasets: DOREMUS (Achichi et al., 2018) and AgroLD (Larmande & Todorov, 2021) that differ from benchmarks in terms of the types of their heterogeneity. DOREMUS is a real-world music-related dataset consisting of three interconnected datasets that describe classical music works and the related events and entities. The data is multilingual with a majority of French text and comes from catalogs and archives of three major French cultural institutions (Radio France, La Philharmonie de Paris, and the French National Library; Lisena et al., 2018). AgroLD consolidates data relevant to the plant science community, including crops such as rice, wheat, and Arabidopsis (Venkatesan et al., 2018). With approximately 900 million triples, AgroLD is the result of annotating and integrating over 100 datasets from 15 diverse sources (Larmande & Todorov, 2021).

Table 2. Comparing Each Two KGs for Each Dataset (All Numbers Indicate Percentages Except for KG Sizes, Which Indicates the Number of Entities).

					Reference Alignment	
Dataset	JS Divergence	Max Difference in Percentage of Nodes	KG Sizes	Size Similarity	Levenshtein Normalized Similarity	EA Semantic Similarity
DBP15K$_{FR-EN}$	5.55	1.87	#S 19,661 #T 19,993	98.3	60.1	90.5
SPIMBENCH	4.41	2.45	#S 2,966 #T 3,082	96.2	36.6	66.2
ICEWS-WIKI	36.1	6.21	#S 11,047 #T 15,831	69.78	–	–
ICEWS-YAGO	43.0	10.37	#S 26,863 #T 22,555	83.9	–	–
DOREMUS	16.8	14.0	#S 2,057 #T 1,889	92.8	30.3	46.6
AgroLD	6.84	6.22	#S 96,117 #T 51,488	53.6	19.6	56.6

Note. KG = knowledge graph; JS = Jensen–Shannon; EA = entity alignment.

To get an idea of the extent to which the KGs in each dataset differ in scale, we show in Table 2 the sizes of the source and target KG for each dataset (denoted by #S and #T, respectively). The remaining columns of the table will be introduced and explained as we proceed in this section.

4.2 *Evaluating the Heterogeneity of the Datasets*

We start by taking a bird's view on the datasets and showing the degree distribution of the underlying KGs, that is, the undirected graphs in these datasets in Figure 1. We count the number of entities relative to their degrees and visualize this for degrees up to the point where 90% of the nodes have a degree below that threshold. We also considered visualizing up to the median or median unique degree. However, we believe it is not appropriate to plot up to these values, as the median only represents the point at which 50% of the entities are below or equal, and fewer than five entities have the median unique degree.

The figure shows that the number of nodes having the same degrees is similar in the pair of KGs in both DBP15K and SPIMBENCH datasets, indicating similar degree distributions in these two datasets. However, this is not the case for DOREMUS, AgroLD, ICEWS-WIKI, and ICEWS-YAGO, where we can see that the number of nodes having the same degree is different across the KGs in each dataset.

To get a more in-depth understanding of the dataset heterogeneities, we proceed to compute three statistical and two qualitative metrics for each pair of KGs in our datasets.

4.2.1 Jensen–Shannon (JS) Divergence. To statistically analyze the underlying distribution of degree sequences in each pair of KGs, we opt for applying the JS divergence test. We found JS divergence or JS distance (Endres & Schindelin, 2003), a suitable statistical metric that captures the amount of overlap between two distributions by using a bi-directional Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951). KL divergence, defined in equation (1), measures how one reference probability distribution P is different from a second probability distribution Q.

$$D_{\mathrm{KL}}(P||Q) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right). \tag{1}$$

The JS divergence is a symmetrized and smoothed version of the KL divergence $D_{\mathrm{KL}}(P||Q)$, defined as follows:

$$D_{\mathrm{JS}}(P||Q) = \frac{1}{2}D_{\mathrm{KL}}(P||M) + \frac{1}{2}D_{\mathrm{KL}}(Q||M), \tag{2}$$

where $M = (1/2)(P+Q)$ is a mixture distribution of P and Q. JS divergence gives us a number in the range [0, 1], where the higher the number, the more divergent the distributions. The results of our study on JS divergence are given in Table 2.[6] For each KG, we construct the empirical degree distribution by counting the frequency of each node degree and normalizing

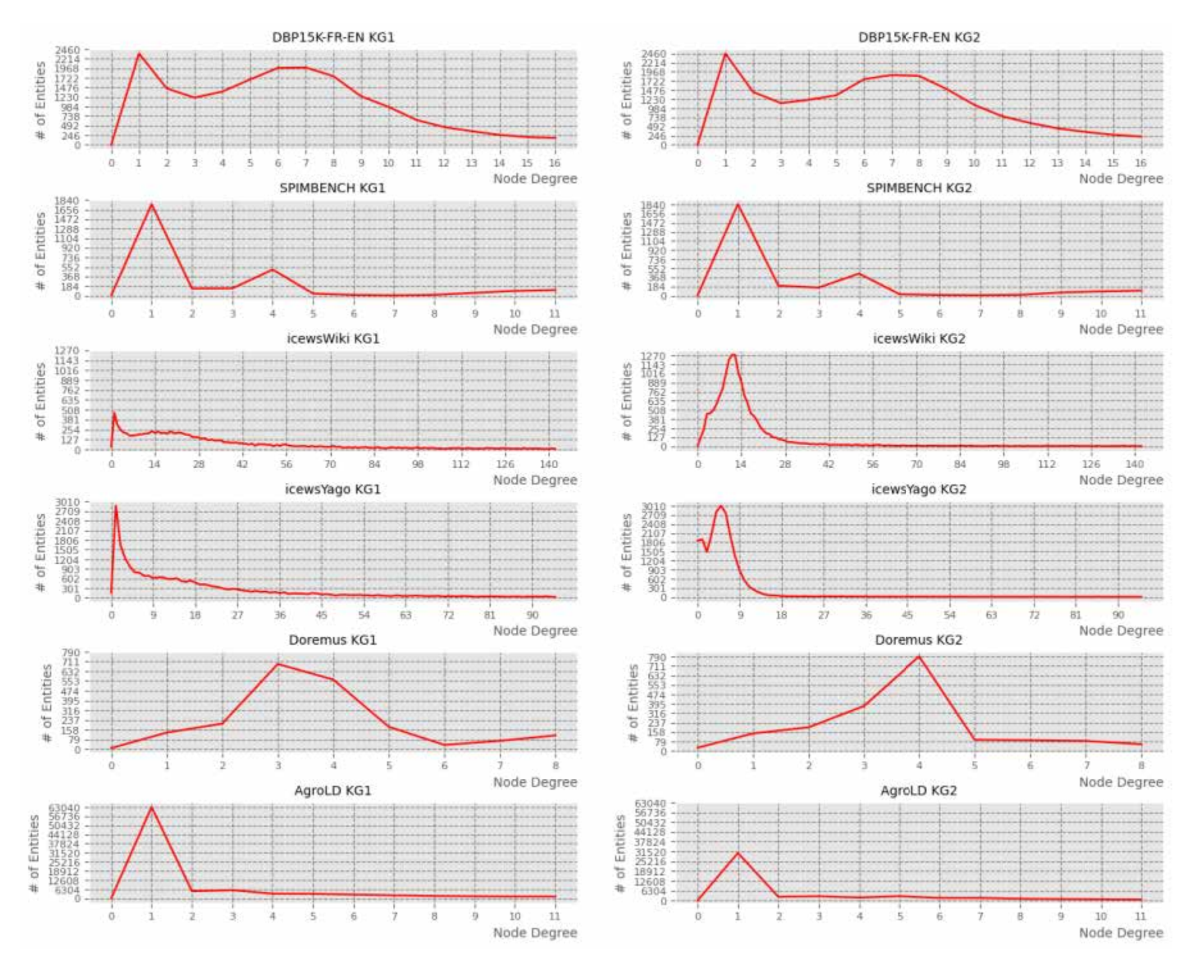

Figure 1. Degree distribution of each two knowledge graphs (KGs) for each dataset.

these counts into a probability vector. To make the two distributions comparable, we align them over the union of their degree supports, filling missing degrees with zeros. We truncate at the smallest maximum degree for which only a single node occurs in either KG to reduce noise from extreme tails. We then compute the JS distance between the two normalized distributions. We notice that the degree distributions of KGs in DBP15K$_{\mathrm{FR-EN}}$ and SPIMBENCH are less divergent than their counterparts in DOREMUS and AgroLD, as well as that there is a much higher level of heterogeneity in the degree distributions of the ICEWS-WIKI and ICEWS-YAGO datasets than in the other datasets. This is not surprising, since these two datasets have been generated to mimic the JS ratios of ICEWS, YAGO, and WIKI KGs, which have very different degree distributions.

4.2.2 Maximum Difference in Percentage of Nodes w.r.t. Node Degrees. To better recognize the differences in the KGs' degree distributions, we calculate our second statistical metric, which measures the maximum difference in the percentage of entities with respect to the degrees in each pair of KGs. By looking at the second column of Table 2, we can see that the maximum difference in the percentage of the nodes across the KGs (w.r.t. the node degrees) in DOREMUS is much higher than in all other datasets. This confirms the observation in Figure 1, showing that the percentage of entities having the same degree in the two KGs varies less in the benchmark datasets (DBP15K$_{\mathrm{FR-EN}}$ and SPIMBENCH), as compared to the other datasets.

4.2.3 Size Similarity. As a third statistical metric, we calculate the normalized difference between the number of entities in every pair of KGs, so we can compute the similarity in the size of KGs using equation (3):

$$\text{Size\_similarity} = 1 - \frac{|s(\text{KG}_1) - s(\text{KG}_2)|}{\max(s(\text{KG}_1), s(\text{KG}_2))}, \tag{3}$$

where $s(\text{KG})$ returns the size (i.e., number of nodes/entities) of a given KG. We divide the absolute value of the difference in sizes of the KGs by the size of the larger KG. Table 2 shows that the size of the KGs in the benchmark datasets is almost the same, while in real-world cases, the two KGs might include very different numbers of entities, as is the case for the AgroLD dataset. The size similarity of 53.6% for AgroLD indicates that the size of one of its KGs is almost twice as big as that of the other KGs. The two KGs in the AgroLD real-world dataset differ more strongly in scale even than their counterparts in the two highly heterogeneous synthetically generated datasets of ICEWS-WIKI and ICEWS-YAGO.

Analyzing the results of the three statistical features (Table 2), in combination with observing the degree distributions (Figure 1), reveals the higher level of structural heterogeneity in KGs of DOREMUS and AgroLD as compared to the two synthetic benchmark datasets. Moreover, there is less JS distance between the degree distribution of KGs in benchmark datasets in comparison with the other datasets. Furthermore, in all non-benchmark datasets, the ICEWS-YAGO dataset contains KGs having the least overlaps in their degree distributions, and AgroLD is the dataset that includes KGs having the most difference in scales.

Nevertheless, none of these three statistical metrics are indicators of the textual/lingual properties of the entities. We therefore turn our attention to string-level features.

4.2.4 Normalized Levenshtein Similarity. In order to get a more enhanced understanding of how dataset heterogeneities affect each model's performance, we need a qualitative heterogeneity metric, especially for approaches such as BERT-INT, MultiKE, and i-Align that use the textual attribute values of the entities. Even RDGCN, instead of random initialization, uses a representation vector of the entity name as the entity's initial embedding. In addition, since all four of the analyzed methods have been trained in a supervised manner, they all use some part (30%) of the reference alignment as their training data. Hence, the quality of data of the reference alignment affects the model's performance directly. Hence, we explore two qualitative metrics—one based on the Levenshtein similarity (in this subsection) and one based on the embeddings' semantic similarity (in the following subsection).

As a first qualitative metric, we get an average over the Levenshtein normalized similarity of the attribute values for all pairs of aligned entities in the reference alignment. Levenshtein, or edit distance, is originally a measure of the closeness of two strings. It quantifies the minimum number of single-character edits (insertions, deletions, or substitutions) required to transform one string into the other (Yujian & Bo, 2007). The resulting distance is always in the interval [0, 1]. The similarity is computed by subtracting the normalized distance by 1. To analyze the quality of the data, we focus on the reference alignment, and for the first step, we compute the average over the Levenshtein normalized similarity of the attribute values of each pair of aligned entities (cf. results in Table 2).

Because minor variations in the input data do not affect the performance of the language models in embedding the texts (Elekes et al., 2017; Heylen et al., 2008) and regarding the fact that in EA methods that utilize the attribute values of entities (including three of our employed methods), a language model is used for initial embeddings of the entities (Shen et al., 2022; Z. Zhang et al., 2020), we first lemmatized and stemmed all the words in each attribute value. Then, we compared all attribute values for each pair of entities in the reference alignment, and we computed the maximum Levenshtein similarity between each pair of attribute values and averaged all. Due to the multilingual nature of DBP15K$_{\text{FR-EN}}$, we used Google Translate to get the English version of the French KG.

The results of the Levenshtein measurements reported in Table 2 indicate that, despite possible translation errors in the DBP15K French KG (Wu et al., 2019), the normalized Levenshtein similarity of aligned entities in the benchmark datasets is higher than in the real-world datasets. This suggests that EA methods, particularly those relying on textual descriptions of entities, may perform better on benchmark datasets. Since attribute triples are not included in the ICEWS-WIKI and ICEWS-YAGO datasets, calculating the Levenshtein measure based on attribute values is not possible for these datasets.

4.2.5 Semantic Similarity. While the normalized Levenshtein similarity offers insights into the textual closeness of aligned entities in each dataset, it primarily focuses on character-level differences and does not capture the semantic or contextual similarities between entity pairs. Therefore, we further investigate the semantic similarity between the aligned entities in the KGs.

As mentioned, approaches using the entity or predicate features as input usually utilize a language model to embed the entities. The studies show that the performance of these methods is relevant to the quality of the initial embeddings (Sun

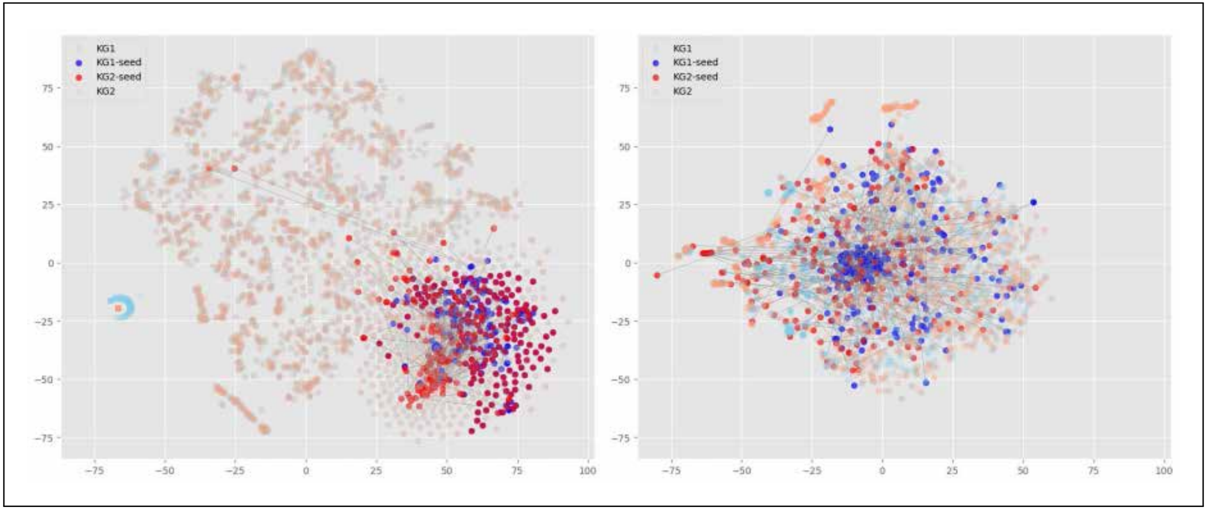

Figure 2. Reduced-dimension BERT-based initial entity embeddings of SPIMBENCH (to the left) and DOREMUS (to the right).

et al., 2020; Wang et al., 2019). Hence, we want to measure the similarity of two aligned KGs (Traverso et al., 2016; Zhu & Iglesias, 2016) based on initial embeddings of entities in the reference alignment. Because language models capture the semantic similarity of words, we rely on the entity embeddings. We apply the well-known normalized Euclidean relative distance over the pairs of entities of the reference alignment, which is a common choice (Fanourakis, Efthymiou, Kotzinos, & Christophides, 2023), given as follows:

$$\text{Semantic\_similarity}(\text{KG}_1, \text{KG}_2) = 1 - \frac{1}{s}\sum_{i=0}^{s} f(d_i), \tag{4}$$

where

$$d_i = \frac{\|v(\mathbf{e}_i) - v(e'_i)\|}{\|\sum_{j=0}^{s} v(\mathbf{e}_i) - v(e'_j)\|}, \tag{5}$$

with s being the size of the set of seed alignments $S = \{(e_0, e'_0), \ldots, (e_s, e'_s)\}$, and $v(\mathbf{e}_i)$—the initial representation vector of entity $\mathbf{e}_i$ that has been obtained by a pre-trained multilingual BERT model over entity's descriptions.[7] The relative distances d_i have been normalized using the MinMax scaler function f:

$$f(d_i) = \frac{d_i - \text{Min}(d_i)}{\text{Max}(d_i) - Min(d_i)}, \quad i = 1, \ldots, s. \tag{6}$$

Notice that by definition, having a lower value of the EA semantic similarity for a dataset indicates a higher level of semantic heterogeneity. The corresponding semantic similarities on our pairs of KGs in each dataset are reported in the last column of Table 2. The similarities are calculated based on the initial embeddings of attribute values of the aligned entities using the pre-trained BERT model. Note that this measurement is important when we analyze the performance of the embedding-based EA models, which use the entities' attribute values in their frameworks. As reflected by the semantic similarity results, the real-world datasets have a higher degree of semantic heterogeneity in comparison to the benchmark datasets. This confirms what we already observed with the help of the Levenshtein similarity, showing that from both character-level and conceptual perspectives, aligned entities from the benchmark datasets have more similar textual descriptions than those in real-world datasets.

To visualize how the semantic similarity in different synthetic benchmark and real-world datasets differs, we applied t-distributed stochastic neighbor embedding (t-SNE; Van der Maaten & Hinton, 2008). t-SNE is a dimensionality reduction technique commonly used for visualizing high-dimensional data in lower-dimensional space, typically 2D or 3D. In Figure 2, we visualize the entity embedding spaces of the SPIMBENCH and DOREMUS datasets that, according to Table 2, are datasets with a low and a high level of EA semantic similarity, respectively.[8]

The dark blue and red points represent the seed alignment of the KGs, while each entity in the seed alignment is connected to its counterpart using grey lines. Looking at the grey lines that show the distance between the initial embeddings of the entities in the reference alignment, one can easily recognize how far the entities are located in the DOREMUS real-world dataset. In SPIMBENCH, only two aligned entities have a long distance, and for the other aligned samples, the distance is much shorter than what we can see for the DOREMUS dataset. It is important to note that in Figure 2, for the SPIMBENCH dataset, several red dots appear in the bottom-right corner, seemingly unlinked. In reality, they are connected to nearly overlapping dark blue dots. The line between them is not visible because the initial embeddings of the source and target entities are so similar that the connection line becomes imperceptible. Additionally, there are some red dots located in the middle-left of the figure, surrounded by light-blue dots. These light-blue dots represent entities in the source KG that have very similar initial embeddings but are not part of the seed alignment, meaning they do not have a corresponding match in the target KG. The plots confirm the higher level of semantic heterogeneity in the real-world DOREMUS dataset as compared to the benchmark dataset SPIMBENCH.

5 Comparative Analysis of the Embedding-Based Methods

In this section, we present the results of implementing and applying the selected EA models on the chosen datasets. We first explain the challenges of using the models on real-world and less well-known benchmark datasets and how we overcome these issues, this being part of the lessons learnt in this work. Next, we discuss the evaluation metrics employed by the models and present the results of our experiments. Further on, we provide an overview of how the models perform on both benchmark and real-world datasets. We also investigate how these performances relate to the dataset features in light of the discussion in Section 4. Finally, we look into the inference capacities of the models when facing the full-scale graphs (instead of their corresponding validation sets).

5.1 Datasets Preparation for Applying the EA Models

For each dataset, we have a file of the source KG, a file of the target KG, and a file containing the reference alignments in XML, turtle, or Ntriples format. To feed the data to each model, we prepare a series of files following the naming convention and formats required by each model, for example, json, pkl, txt, or other. In this process, we confronted issues either related to the dataset design itself, for example, using blank nodes, or related to the model input's design, for example, when there is no instruction about the proper model input. Even with correctly formatted inputs, runtime errors can still occur unexpectedly due to minor changes in the input data. This necessitates a thorough process of data validation to ensure the models function correctly. Data validation in an ML pipeline ensures that training data is error-free and accurate, preventing issues that could degrade model performance during deployment and safeguarding against errors introduced during data processing (Polyzotis et al., 2018). Hence, we need to handle the data lifecycle of inputs to each embedding-based EA model (Gudivada et al., 2017), and address as many problems as we face to prepare suitable data. After writing the codes to prepare the proper input files for the four models that we use and validating them on the different benchmark and real-world datasets, we share the codes on a GitHub page[9] to pave the way for researchers to benefit from employing these methods on their specific datasets. We included all the links to the original models on our GitHub repository.

Note that, despite the high heterogeneity aspects of ICEWS-WIKI and ICEWS-YAGO, which make them more similar to the real-world KGs, the only model that we employ for them is RDGCN. The reason is that these two datasets lack the attribute triples, which are the essential features utilized by DLinker and the other three methods, and they only contain the relation triples. We opted not to use additional datasets due to significant structural differences and limited accessibility. The OpenEA datasets, for instance, were generated under the 1-to-1 assumption, omitting unmatched entities. In response to this limitation, Leone et al. (2022) introduced new, more realistic datasets that do not follow this assumption, which we could not access on their repository. While the sampling algorithm code is provided to regenerate the datasets, doing so would result in datasets that differ from DOREMUS and AgroLD, as Leone's datasets are derived from larger KGs, whereas DOREMUS and AgroLD are not. Therefore, we chose not to use additional datasets to maintain consistency in the real-world data analysis.

5.2 Evaluation Metrics

There are two commonly used families of evaluation metrics: (1) precision and recall (and the resulting F_1-score) and (2) Hit@k or, often, Hit@1. Traditionally, EA methods rely on metrics in (1), but the advent of embedding-based methods has moved the focus to metrics of type (2), as they follow the common tradition in link prediction settings. We provide definitions of the two types of metrics and discuss their commonalities and differences in what follows.

We define precision, recall, and F_1-score in the context of a classification problem: precision is the ratio of true positive predictions to the total number of positive predictions made by the model. It measures the accuracy of the positive predictions made by the model. Recall is the ratio of true positive predictions to the total number of actual positive instances in the dataset. It measures the model's ability to identify all relevant instances. The F_1-score is the harmonic mean of precision and recall. It balances both precision and recall in one metric. The respective formulas are given as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \quad F_1 - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{7}$$

where TP stands for true positive, that is, instances for which the model correctly predicted a positive label, FP stands for false positive, and FN stands for false negative, that is, instances for which the model incorrectly predicted a negative label. Hit@1, on the other side, is defined as:

$$\text{Hit@1} = \frac{\text{Number of times the top-ranked prediction is correct}}{\text{Total number of predictions}}. \tag{8}$$

A Hit@K, analogically, is recorded when the correct entity matching the ground truth appears within the top K positions in the ranked list of predictions.

In this work, we make use of both types of metrics: Hit@k allows us to compare our results with the state-of-the-art EA embedding-based methods, which mainly use this metric to report their performance. However, Leone et al. (2022) argue that precision, recall, and F_1-score better represent the performance of EA models. For that reason, we also report these metrics regarding the models we employed on our datasets. We consider it worthwhile to go into more depth regarding the similarities and differences of these two evaluation paradigms. The following discussion contributes to shedding light on the cases when they are considered equivalent, and on those when one of the two types is to be used by preference. During the validation phase of embedding-based EA methods, since the model generates a ranked list of candidates for each entity, and given that the validation set adheres to the 1-to-1 assumption, the precision, recall, and F_1-score are equivalent to Hit@1 (Sun et al., 2020). Leone et al. (2022) mention that using the Hit@k metric for evaluating the model's performance is based on the unfair 1-to-1 assumption. However, this assumption does not hold in real-world EA datasets. Therefore, for EA, evaluation metrics should be used that can take into account the case where no aligned entity is predicted for the given input entity. Based on the similarity measures (or the predicted alignment probabilities) that the EA methods provide at their last evaluation step, Leone et al. (2022) define an assignment module that matches a given entity in the source KG to at most one entity in the target KG (do note that there could be no match for the given source entity). Then, precision and recall are calculated based on their definitions in the field of information retrieval.

A question that might come to mind is why, under the 1-to-1 assumption, Hit@1 is equivalent to precision, recall, and F_1-score. While a few hints have been proposed in Sun et al. (2020) and in Appendix C of the technical report of Leone et al. (2022), we believe that detailing into this claim contributes further to the discussion. Current EA methods evaluate model performance on validation sets that include only positive samples, meaning pairs of corresponding entities, which, as a direct consequence, satisfy the 1-to-1 assumption, that is, for every given entity $e_l \in$ Source_KG included in the validation set, there exists exactly one entity $e'_l \in$ Target_KG, where $e_l \equiv e'_l$ (e_l is the same as e'_l). Suppose n is the size of the validation set (by size, we mean the number of samples included in the validation set), and suppose $\forall i, i = 1, \ldots, n$, in the validation set we have $\mathbf{e}_i \equiv e'_i$. Hence, because the 1-to-1 assumption holds in the validation set, for entity $e_l \in \mathit{Source\_KG}$ we have: $\forall i | i \neq l, e_l \not\equiv e'_i, e'_i \in$ Target_KG.

To show that the precision and recall (and the F_1-score) are equal, based on the definitions, it is enough to show FP = FN. Afterwards, if we show that the total number of the predictions (the denominator in the Hit@1 formula) equals TP + FP, we have shown that Hit@1 equals both precision and recall. Suppose that the class of aligned and non-aligned entities is the positive and negative classes, respectively. In Table 3, we formally define TP, FP, FN, and true negative (TN) in the context of EA models that predict a ranked list for evaluating the model's performance under the 1-to-1 assumption in their validation set.

Considering Hit@1 as the final prediction by EA models, for each entity $\mathbf{e}_i \in$ Source_KG included in the validation set, the model prediction is either its correct match (TP), which is e'_i, or a wrong match (FP), such as, $e'_j, j \neq i$ in Target_KG, and $i, j \in 1, \ldots, n$. Hence, it is trivial that TP + FP equals the total number of predictions. Furthermore, while we don't have any trivial non-aligned predictions by the model, each Hit@1 implicitly gives us some non-aligned predictions. For example, for a false negative, based on the definition, the model should predict that $\mathbf{e}_i \not\equiv e'_i$ for some $i \in 1, \ldots, n$. While the model never predicts a false negative explicitly, but each time that the model predicts that $\mathbf{e}_i \equiv e'_j$ where j indicates any index other than i, the 1-to-1 assumption which has been met in the validation set implies that $\mathbf{e}_i$ in Source_KG is not aligned with any other entity than e'_j in the Target_KG including the e'_i. Hence, whenever the model predicts $\mathbf{e}_i \equiv e'_j | i \neq j$,

Table 3. Defining TP, FP, TN, and FN for EA Models Based on Hit@1 Predictions, When the Validation Set of Size n is Free of the Negative Samples and Meets the 1-to-1 Assumption, where $i, j \in 1, \dots, n$.

Actual/Predicted	Aligned (P)	Non-aligned (N)	Implicitly Non-aligned (IN)
Aligned (P)	TP: $\mathbf{e}_i \equiv e'_i$	FN: $\mathbf{e}_i \not\equiv e'_i$	$\mathbf{e}_i \equiv e'_j$, $\lvert i \neq j$
Non-aligned (N)	FP: $\mathbf{e}_i \equiv e'_j$, $\lvert i \neq j$	TN: $\mathbf{e}_i \not\equiv e'_j$, $\lvert i \neq j$	$\mathbf{e}_i \equiv e'_i$

Note. TP = true positive; FP = false positive; TN = true negative; FN = false negative; EA = entity alignment.

Table 4. Measured Evaluation Metrics of Analyzed EA Models on the Datasets (all Numbers Indicate Percentages). Following the 1-to-1 Assumption During the Models' Evaluation, Hit@1 Equals the Precision, Recall, and F_1-Score for Each Model.

		Methods								
		BERT-INT		RDGCN		MultiKE		i-Align		DLinker
		Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	Hit@1	Hit@10	F_1-score
Datasets	DBP15K$_{FR-EN}$	**99.3**	99.8	88.6*	95.7*	37.5	43.6	26.6	43.2	–
	SPIMBENCH	**82.4**	82.4	77.7	94.7	57.1	57.1	75.0	86.5	70.2
	ICEWS-WIKI	–	–	**75.1**	84.2	–	–	–	–	–
	ICEWS-YAGO	–	–	**68.3**	80.8	–	–	–	–	–
	DOREMUS	47.9	64.1	1.33	5.92	2.70	8.70	53.1	68.0	**95.6**
	AgroLD	21.1	33.2	0.02	0.3	2.30	5.7	4.4	12.1	**59.0**

*The reported numbers are derived from the original study.

which is a false positive, following the 1-to-1 assumption, it implicitly has predicted that $\mathbf{e}_i \not\equiv e'_i$, which is a false negative. Accordingly, FP = FN, and we have shown that precision equals the recall (and the F_1-score). Furthermore, because the number of times the top-ranked prediction is correct equals the TP, and as we have earlier shown, the denominator of the precision and Hit@1 are equal, we can see that precision equals the Hit@1. As a result, all the aforementioned metrics are equivalent whenever the 1-to-1 assumption has been met during the evaluation.

5.3 Evaluation Tasks and Analyses

5.3.1 Real-World Versus Benchmark Datasets. The comparative performance of the selected models on the collection of chosen datasets is given in Table 4. For each dataset, the highest Hit@1 score is indicated in bold. Following the discussion presented in Section 5.2, we compare the Hit@1 results of embedding-based models with DLinker, considering cases where the 1-to-1 assumption was met during the evaluation of embedding-based EA models on each dataset. The best-performing method for each dataset is highlighted in bold. We can observe an overall drop in the performance of embedding-based models when tested on real-world datasets, as compared to benchmark ones. In what follows, we discuss the results of each of the models of interest in light of that global observation, while also considering the internal mechanisms of each of the models that differentiate them from one another and could provide insights into these observations.

The performance of the BERT-INT model is strong on datasets such as DBP15K and SPIMBENCH, achieving high Hit@1 rates (99.3% and 82.4%, respectively). However, its performance drops significantly on our real-world datasets DOREMUS and AgroLD (Hit@1 rates of 47.9% and 21.1%, respectively). The reason for this drop is that BERT-INT relies heavily on the quality and amount of textual information (entity descriptions), as observed in Table 1. Hence, datasets with less textual and semantic similarity and fewer descriptive features, such as DOREMUS and AgroLD (Table 2), lead to a decrease in its performance. This emphasizes the importance of high-quality data descriptions for BERT-INT's success.

Observing the results of RDGCN (Table 4), we can see that its performance is more than 88% and 77% on DBP15K and SPIMBENCH, respectively, and it also drops significantly in the real-world scenarios (close to 0% Hit@1). RDGCN relies solely on graph structure and a GloVe word embedding[10] on entity names (see Table 1). This, along with the statistical metrics from Table 2, which highlight the greater structural heterogeneity of the real-world dataset, helps explain this outcome. Since RDGCN is adaptable to different initial embeddings, we modified the initial embedding of the model to use a multilingual pre-trained BERT model to generate the initial embeddings for entity names. This adjustment improved the Hit@1 and Hit@10 scores on the dataset by only 0.4% and 6%, respectively. This modest improvement is likely due to the fact that most entity names are represented by IDs rather than meaningful text, which limits the impact of the embeddings. We have uploaded the code for generating the initial embeddings with BERT using graphics processing units to our GitHub repository. We also experimented with using BERT-based embeddings of the entities' descriptions (used in BERT-int) as the initial embeddings in RDGCN. To facilitate this, we implemented a method to create a dictionary of entities' descriptions for any pair of given RDF graphs, which is available in our repository. This approach led to improvements in the percentage of the Hit@1 scores across several datasets: 93.70 on DBP15k, 99.53 on Spimbench, 22.49 on DOREMUS, and 7.21 on AgroLD. These represent enhancement percentages of 5.1%, 21.83%, 21.16%, and 7.19%, respectively. This demonstrates that more informative initial embeddings significantly boost RDGCN's performance.

Looking at Figure 1, we can see the long-tailed issue of AgroLD's KGs. The long-tailed problem in graphs (Malekzadeh Hamedani & Kaedi, 2019; Shi, 2013) is described as an issue where a small number of nodes have a substantial number of neighbors, while the majority (referred to as tail nodes) have only a few neighbors (Liu et al., 2021). GNNs used in RDGCN under-represent tail nodes during the training of the model and lead to a low-quality KGE (Liang et al., 2024), and this can explain the drop in performance on this dataset for RDGCN. However, as we observe in Figure 1, SPIMBENCH also has a long tail problem, which does not appear to be an issue for RDGCN. We found two main differences between SPIMBENCH and AgroLD that could explain the drop in performance from one to another of these methods. (1) The number of common neighbors: In the SPIMBENCH dataset, many entities in the reference alignment share common neighbors. On average, 48% of the entities in the reference alignment, across its two KGs, have at least one common neighbor with their linked entities. According to the results presented in Wang et al. (2024), a higher number of common neighbors improves the efficiency of embeddings generated by GCNs for these entities. However, in the AgroLD KGs, none of the linked entity pairs share a common neighbor. As Wang et al. (2024) demonstrated, the performance of GNN-based graph embedding models, including GCNs, correlates more strongly with the number of common neighbors than with node degrees. This suggests that the lack of common neighbors in AgroLD could negatively impact the performance of the RDGCN model. (2) The KGs in AgroLD are bipartite: Giamphy et al. (2023) discuss how GNN-based graph embedding of a large bipartite graph is difficult due to the challenge of merging heterogeneous node and graph-level information while ensuring scalability to handle the graph's increasing size. They also propose a list of available resources that perform better on bipartite graph embedding (but unfortunately, we found none of them working on the multi-relational graph embedding, which is our case). Moreover, RDGCN uses word embedding models to produce the initial embeddings of the entities using entity names. Because the names (which are the last part of the entity URIs by the model's default) of the musical works and the proteins/genes in DOREMUS and AgroLD have been defined by IDs in their respective ontologies, the initial entity embeddings would not be able to guide the embedding module to a better result. All the aforementioned observations can explain the low performance of RDGCN on DOREMUS (1.2% of Hit@1) and AgroLD (<1% of Hit@1). Furthermore, the results of RDGCN on the ICEWS-WIKI and ICEWS-YAGO datasets suggest again the contribution of the high-quality entity names to improving the model's performance. As a result, these two datasets are structurally less complex for the model compared to real-world datasets.

Although MultiKE outperforms several EA translational-based methods (Zhang et al., 2019) using a multi-view KGE technique, this model overall performs the weakest among the employed embedding- and non-embedding-based models on the selected datasets. Similar to its predecessors (Table 4), MultiKE's performance also drops for DOREMUS and AgroLD. Recall that we observe a higher level of structural and qualitative heterogeneities in these two real-world datasets than in the benchmark datasets (Table 2). Hence, the fact that MultiKE relies on both the graph structure and textual information of entities and their attributes (Table 1) can explain the gap in the performance of this model. Furthermore, while DBP15K is less heterogeneous than the SPIMBENCH, we suspect the reason for the worse performance of MultiKE on this dataset is the multilinguality that could not be handled using a pre-trained English word2vec model[11] that MultiKE is employing for the entities' local name embeddings. To embed the French language in Doremus and DBP15K using MultiKE, we were unable to use a pre-trained multilingual BERT model due to the method's strong reliance on word2vec. Instead, we added a French word2vec dictionary to the existing English one, which led to significant improvements in MultiKE's performance on the DOREMUS dataset. Specifically, Hit@1 increased to 30.7% and Hit@10 rose to 34.4%, which seems reasonable given the predominance of French text in DOREMUS. However, the inclusion of this multilingual collection significantly reduces MultiKE's performance on DBP15K, with Hit@1 and Hit@10 dropping to 0.53% and 3.18%, respectively. This represents a decrease of 37% in Hit@1 and 40.4% in Hit@10. We believe this decline is due to a lack of mappings between the French and English word vectors, which causes conflicts in the embeddings of the two languages.

Finally, i-Align uses two transformer encoders for text and graph embeddings. As Table 4 shows, it performs better on SPIMBENCH and DOREMUS (75.0% and 53.1% of Hit@1, respectively) as compared to DBP15K (26.6% of Hit@1), and its performance drops significantly when it comes to AgroLD (4.4% of Hit@1). We suspect that the reason for the model performing worse on DBP15K as compared to DOREMUS is the fact that only the first 10 characters of the attribute values were considered, while the rest of the sequence was ignored by the textual transformer-based encoder. Due to the curse of multilinguality issue by transformers (Blevins et al., 2024; Pfeiffer et al., 2022) and inter-language competition for the model parameters, it seems this limited amount of data may not suffice to train the same text transformer's parameters. Additionally, during our experiments, we discovered that reducing the length of textual properties of the entities in the BERT-INT model can result in a significant reduction in performance by as much as 19%. This again illustrates the importance of retaining the informative attribute descriptions included in the values.

As a baseline of non-embedding-based approaches, we used the DLinker method. Because this model fundamentally finds the longest common subsequence in the descriptions of a pair of entities belonging to two different KGs, it does not support EA on the multilingual dataset of DBP15K$_{FR-EN}$. Moreover, by comparing the Hit@1 of the embedding-based EA models (Table 4), DLinker is not the best-performing method on the SPIMBENCH, but it shows the top performance on the real-world DOREMUS and AgroLD datasets.

Furthermore, since DLinker finds the alignments based on the greedy strategy of finding the longest common subsequence and ignoring the rest of structural or literal information, and since DLinker's performance is significantly better than the embedding-based methods, we can conclude that in cases where we have real-world data, taking the extra volume of information into account does not help the quality of the embeddings but enforces more noise to them. Although this conclusion holds for all categories of embedding-based EA methods, the translational and GNN-based methods, which rely primarily on graph structure, introduce more noise into the embeddings. However, i-Align also embeds the graph structures using a graph transformer to embed the local subgraphs containing the nodes that have high interconnectivity with the given entities in two KGs. By comparing the performance of i-Align with RDGCN and MultiKE, we recognize that using the transformer attention mechanism for local subgraph embedding propagates noise less than the GNN message passing and translational systems in the more heterogeneous real-world cases (see results of these three methods on and AgroLD datasets in Table 4). The other reason that i-Align has a better performance than the RDGCN and MultiKE seems to be the fact that it relies more than those on the literals and textual properties of the entities, as we see that both i-Align and BERT-INT perform better than the other methods for the real-world cases. Furthermore, by comparing the performance of i-Align and BERT-INT on AgroLD, we can conclude that an interaction training method that uses almost all textual properties of the entities is the best choice for the case where we have structurally and semantically heterogeneous large-scale KGs. Because the interaction training methods mostly rely on the comparison between the properties of pairs of entities belonging to two KGs rather than comparing them as particles of the large KGs they belong to, we can conclude that for doing an embedding-based EA task on real-world datasets, a local comparison of the given entities in two KGs will guide the model to predict higher quality alignments.

Generalizability Assessment. Inspired by prior work on domain generalization (Fan et al., 2024; Gulrajani & Lopez-Paz, 2021), we examine the robustness of existing EA models by evaluating their performance on datasets that differ from synthetic benchmarks in structure and semantic similarity levels. However, unlike these studies that focus on training on one domain and testing on a different one, we investigate how well-established EA methods perform when applied to more heterogeneous and less curated datasets, without altering their training or optimization procedures. To quantify this, we compute the average Hit@1 on the real-world datasets DOREMUS and AgroLD. As shown in Table 4, BERT-INT reaches an average Hit@1 of 34.5%, while i-Align, MultiKE, and RDGCN score 28.75%, 2.5%, and 0.675%, respectively. This superior performance suggests that interaction-based models such as BERT-INT generalize more effectively to heterogeneous real-world scenarios compared to structure-dependent embedding models, although further investigation would be needed to confirm their generalizability across other domains. Overall, our results show that even though embedding-based models perform very well on some benchmark datasets (e.g., 99.3% of Hit@1 for BERT-INT on DBP15K$_{FR-EN}$), it seems that they overfit on the benchmark data.[12] Consequently, using these models could lead to errors in producing alignments in heterogeneous real-world datasets. Hence, we observed how dataset features presented in Table 2 together with the results of our experiments in this section can justify the gap between the performance of the selected methods on benchmark and real-world datasets.

5.3.2 Analyzing the Models' Effectiveness of Inference. In this section, we focus on the capabilities of models in the inference phase. As a common practice, under the 1-to-1 assumption, models such as RDGCN and BERT-INT consider only a subset of the reference alignment as a validation set during the model evaluation, ignoring the rest of the space. That corresponds to the subspace of dark-colored points in Figure 2 (reference alignments). Such under-representation of the search space undermines the reliability of the reported results, as well as the efficiency of these methods in predicting correct alignments beyond the validation set. Indeed, some real-world studies have removed the 1-to-1 assumption in dataset generation, allowing for more complex scenarios with non-matchable entities. However, many EA models still only focus on data that involves ground truth entities, sometimes even using only ground truth for training. As a result, these models fail to

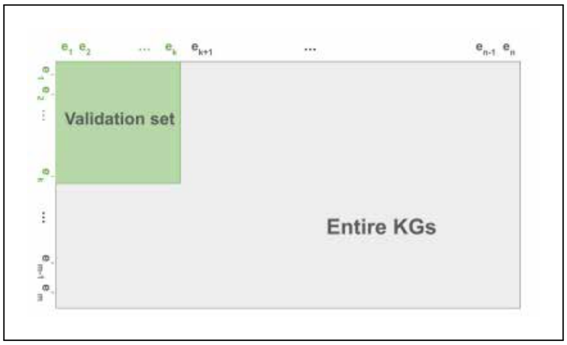
e_1 e_2 … e_k e_{k+1} … e_{n-1} e_n
e'_1 e'_2 … e'_k … e'_{m-1} e'_m
Validation set
Entire KGs

Figure 3. Illustration of the search (comparison) spaces of the EA models in the limited case (imposing the 1-to-1 assumption) and in the extended case (entire graphs). *n* and *m* are the sizes of the source and target KGs, respectively. *Note*. EA = entity alignment; KG=knowledge graph.

consider the non-matchable entities added to the dataset as candidates. This means the model's performance remains the same as if it were working under the 1-to-1 assumption, because it doesn't effectively handle the additional non-matchable data. In Figure 3, we illustrate the comparison space of EA models that impose the 1-to-1 assumption during evaluation. In this context, the similarity matrix used to identify the top-ranked predictions is a square matrix, as depicted by the green one in Figure 3. This matrix excludes comparisons between entities in the validation set and other entities in the KGs, such as non-matchable entities and those utilized during training. Later in this section, when we extend the comparison space—first to compare source-to-target and then to compare target-to-source, the results show a decrease in Hit@1. This suggests that the best-predicted match in the extended comparison is not always the same as in the more restricted case. Moreover, for certain entities in the validation set, their most likely alignment may actually be outside the validation set, highlighting a lack of efficient embedding for these entities.

Therefore, in what follows, we assess the models' performance on two versions of the datasets: (1) a limited validation set scenario, and (2) an extended scenario where each source entity's candidate search space includes a broader set of entities from the target KG, rather than being restricted to only those in the validation set. In Figure 4, we visualized the results of our experiments on the performance of BERT-INT and RDGCN. We utilized the repository provided by Leone et al. (2022) to measure F_1-score in the extended case. However, we encountered difficulties in replicating its original performance when applied to the RDGCN method. As a result, we re-implemented its assignment module to ensure accurate computation of these metrics. It is important to note that the module presented in Leone et al. (2022) excludes the portion of the ground truth that had been seen during training as alignment candidates for the entities in the validation set. In contrast, we applied a stricter condition: we considered all entities from the target KG (including the ground truth entities used during training) as candidates for aligning with a given entity from the source KG in the validation set. We make two main observations: although the Hit@1 measure decreases from the limited to the extended search space case, this decrease is not significant. However, if we look at the F_1-score, the situation is drastically different, where an important drop in performance of the two models can be observed from the limited to the extended search space scenario.

As illustrated by the comparison between the blue and dark-green bars in Figure 4, there is a performance decline of 5.66% and 45.15%, as measured by Hit@1, for RDGCN on the DBP15K$_{\text{FR–EN}}$ and SPIMBENCH datasets, respectively, when the search space is extended. This drop is of 0.5% and 4.3% for the same datasets, respectively, when it comes to using BERT-INT. Due to RDGCN's very low performance in the limited scenario, this is not surprising that extending the candidates' search space of DOREMUS and AgroLD causes Hit@1 of RDGCN to drop to zero. Furthermore, extending this space on DOREMUS and AgroLD causes Hit@1 of BERT-INT to drop by 10.8% and 2.6%, respectively. However, as Leone et al. (2022) pointed out, precision, recall, and F_1-score are fairer metrics for evaluating EA tasks. To assess this, we used their repository[13] to test BERT-INT, and we re-implemented the assignment module to calculate these metrics for RDGCN, by expanding the search space and removing the 1-to-1 assumption. Note that in Leone et al. (2022), the candidates are chosen from both the validation set and non-matchable entities, not the entire KGs. The results of this experiment are visualized in light-green bars of Figure 4.

Recall that Hit@1, as shown in Table 4, is equivalent to F_1-score under the 1-to-1 assumption, that is, in the limited search space case, while this equivalence does not hold anymore in the extended case. Comparing the blue and light-green bars in Figure 4 shows that extending the candidates' search space to include also the non-matchable entities in the KGs causes the F_1-score of RDGCN to drop by 12.6%, 72.6%, 1.33%, and 0.02% on DBP15K$_{\text{FR–EN}}$, SPIMBENCH, DOREMUS, and AgroLD, respectively. Similarly, BERT-INT's performance is decreased by 27.9%, 77.4%, 46.2%, and

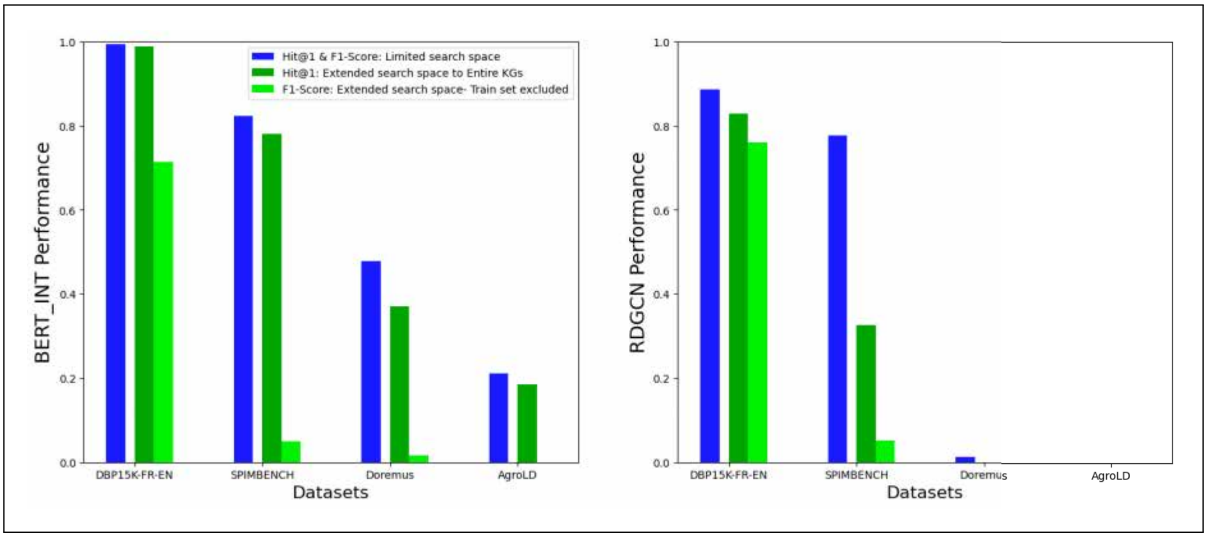

Figure 4. Hit@1 and F_1-score of BERT-INT and RDGCN models on the validation data, considering the candidate search space limited to the reference alignment (shown in blue) or extended to the whole knowledge graph (KG) space (shown in green).

21.1% for the same datasets, respectively. Although we applied a stricter extension condition, looking at Figure 4, we can see that for the vast majority of cases, Hit@1 is still higher than the F_1-score of the models in the extended case. Since the reduction in F_1-score is more notable than the Hit@1, and since the assignment module in Leone et al. (2022) predicts a pair of entities as an alignment only in the case that those entities are predicted as counterparts in source-to-target and target-to-source predictions, our results suggest that these models have difficulty making symmetric predictions. These results also show that embedding-based EA models still encounter generalizability issues and need improvements in order to be able to find alignments in the realistic search space of the KGs.

6 Conclusion and Future Work

The objective of this work is to build upon and complement recent empirical studies in the field of embedding-based EA (Fanourakis, Efthymiou, Christophides, et al., 2023; Fanourakis, Efthymiou, Kotzinos, & Christophides, 2023; Leone et al., 2022; Sun et al., 2020; Zhang et al., 2022), offering a critical perspective on the different models and their limitations, particularly in relation to the challenges posed by various types of datasets and the evaluation process. Therefore, we aim for this study to open new methodological avenues, without focusing on proposing a new model.

We conducted an in-depth analysis of the features of several real-world datasets compared to popular benchmark datasets. Also, we presented an empirical study analyzing the performance of embedding-based EA models beyond test data and on real-world heterogeneous data. We observed that a number of EA embedding-based models, such as BERT-INT and RDGCN with very strong performance for the task of EA on the well-known DBP15K dataset, suffer a drop in performance on real-world data with heterogeneous textual properties. Hence, the results of our study shed light on the benchmark overfitting issue of EA methods discussed in Roelofs (2019) and Todorov (2019), that is, the scenario where the model is tuned excessively to perform well on specific benchmark datasets or evaluation metrics, at the expense of its generalization ability to new, unseen real-world data.

It appears challenging to identify a single structure-related meta-feature that accounts for the performance drops of all methods across different datasets, as each method captures the structure from a different perspective. Since, however, heterogeneity is not just limited to diversity in size and degree distribution, we observed the semantic similarity over the reference alignments to be well-correlated with the performance of EA models that employ a language model, helping explain the performance issues. Then, by investigating the reasons for performance fluctuations of EA models regarding the heterogeneities of real-world datasets, we found interaction training models better fit for driving the EA task in real-world, especially large-scale scenarios. Although interaction training models showed promise on real-world data, we could not conduct a deeper analysis of how they handle noise and ambiguity, due to time and resource constraints. We leave this as an important direction for future work.

Most of the existing embedding-based EA methods simplify the inference process by considering the 1-to-1 assumption (Zeng et al., 2021) and use just a limited portion of the embedding space for the evaluation. This seems to be neither fair nor practical when it comes to using them to discover unseen alignments. As a result, there is a need to go toward an inductive learning EA approach, in which models are trained on pairs of entities from two aligned KGs to predict alignments between unseen entities belonging to the same KGs, as well as matches between entities in other unseen KGs. By addressing this challenge, we believe that EA models will be able to uncover a significantly larger number of alignments across different pairs of KGs.

Acknowledgments

Our team was granted access to the HPC resources of IDRIS under the allocation 2024-AD010315119R1 made by GENCI. The authors acknowledge the ISO 9001 certified IRD i-Trop HPC (South Green Platform) at IRD Montpellier for providing HPC resources that have contributed to the research results reported within this paper.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is partially supported by the French National Research Agency under the ANR DACE-DL project, grant number ANR-21-CE23-0019.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

ORCID iDs

Ensiyeh Raoufi https://orcid.org/0000-0003-0744-6876
Bill Gates Happi Happi https://orcid.org/0000-0003-1199-7726
Pierre Larmande https://orcid.org/0000-0002-2923-9790
François Scharffe https://orcid.org/0000-0002-0010-0058
Konstantin Todorov https://orcid.org/0000-0002-9116-6692

Notes

1. While this work was conducted with KGs represented in the resource description format (RDF) format, the research is not restricted to this paradigm and applies to property graphs as well.
2. Note that the 1 : 1 assumption is sometimes “imposed” by the models, while in many cases it is inherent to the benchmark datasets, as is the case in the OpenEA datasets, discussed in detail below.
3. The line graph of an undirected graph G is another graph that represents the adjacencies between edges of G. The line graph of the given graph G is constructed by making a node (vertex) instead of each edge in G. Then, for every two edges in G that have a vertex in common, we make an edge between their corresponding vertices in the line graph of G. See https://en.wikipedia.org/wiki/Line_graph.
4. https://oaei.ontologymatching.org/.
5. https://hobbit-project.github.io/OAEI_2022.html.
6. Note that the JS measurements have been multiplied by 100 to show the percentage.
7. https://huggingface.co/google-bert/bert-base-multilingual-cased.
8. Although the amount of semantic similarity for the DBP15K dataset is much higher than that of SPIMBENCH, we visualized SPIMBENCH because it has much fewer entities, and this facilitates the visualization.
9. https://github.com/dace-dl-anr/Create_Input_Data_to_EA_Models.
10. https://nlp.stanford.edu/projects/glove/.
11. https://fasttext.cc/docs/en/english-vectors.html.
12. Benchmark overfitting, meant as models being too good on benchmark datasets and less so on unseen real-world ones (Todorov, 2019), is not to be confused with data overfitting of models while training.
13. https://github.com/epfl-dlab/entity-matchers.

References

Achichi, M., Bellahsene, Z., Ellefi, M. B., & Todorov, K. (2019). Linking and disambiguating entities across heterogeneous RDF graphs. *Journal of Web Semantics*, *55*, 108–121. https://doi.org/10.1016/j.websem.2018.12.003

Achichi, M., Lisena, P., Todorov, K., Troncy, R., & Delahousse, J. (2018). Doremus: A graph of linked musical works. In *The semantic Web–ISWC 2018: 17th international semantic web conference, Monterey, CA, USA, October 8–12, 2018, proceedings, part II 17* (pp. 3–19). Springer. https://doi.org/10.1007/978-3-030-00668-6_1

Ardjani, F., Bouchiha, D., & Malki, M. (2015). Ontology-alignment techniques: Survey and analysis. *International Journal of Modern Education & Computer Science*, *7*(11), 67–78. https://doi.org/10.5815/ijmecs.2015.11.08

Bader, S. R., Grangel-Gonzalez, I., Nanjappa, P., Vidal, M. E., & Maleshkova, M. (2020). A knowledge graph for Industry 4.0. In *The semantic web: 17th international conference, ESWC 2020, Heraklion, Crete, Greece, May 31–June 4, 2020, proceedings 17* (pp. 465–480). Springer. https://doi.org/10.1007/978-3-030-49461-2_27

Ben Ellefi, M., Bellahsene, Z., Breslin, J. G., Demidova, E., Dietze, S., Szymański, J., & Todorov, K. (2018). RDF dataset profiling—a survey of features, methods, vocabularies and applications. *Semantic Web*, *9*(5), 677–705. https://doi.org/10.3233/SW-180294

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(8), 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

Beretta, V., Boland, K., Seen, L. L., Harispe, S., Todorov, K., & Tchechmedjiev, A. (2020). Can knowledge graph embeddings tell us what fact-checked claims are about? In *Workshop on insights from negative results in NLP* (pp. 71–75). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.insights-1.11

Berrendorf, M., Faerman, E., Melnychuk, V., Tresp, V., & Seidl, T. (2020). Knowledge graph entity alignment with graph convolutional networks: lessons learned. In *Advances in information retrieval: 42nd European conference on IR research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, proceedings, part II 42* (pp. 3–11). Springer. https://doi.org/10.1007/978-3-030-45442-5_1

Blevins, T., Limisiewicz, T., Gururangan, S., Li, M., Gonen, H., Smith, N. A., & Zettlemoyer, L. (2024). Breaking the curse of multilinguality with cross-lingual expert language models. In *Proceedings of the 2024 conference on empirical methods in natural language processing* (pp. 10822–10837). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.emnlp-main.604

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on management of data* (pp. 1247–1250). Association for Computing Machinery. https://doi.org/10.1145/1376616.1376746

Bonatti, P. A., Decker, S., Polleres, A., & Presutti, V. (2019). Knowledge graphs: New directions for knowledge representation on the semantic web (Dagstuhl seminar 18371). *Dagstuhl Reports 8*(9), 29–111. https://doi.org/10.4230/DagRep.8.9.29

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems* 26 (pp. 2787–2795). https://proceedings.neurips.cc/paper_files/paper/2013/file/1cecc7a77928ca8133fa24680a88d2f9-Paper.pdf

Cai, W., Ma, W., Zhan, J., & Jiang, Y. (2022). Entity alignment with reliable path reasoning and relation-aware heterogeneous graph transformer. In *Proceedings of the thirty-first international joint conference on artificial intelligence, IJCAI-22* (pp. 1930–1937). Association for Computing Machinery. https://doi.org/10.24963/ijcai.2022/268

Cao, J., Fang, J., Meng, Z., & Liang, S. (2024). Knowledge graph embedding: A survey from the perspective of representation spaces. *ACM Computing Surveys*, *56*(6), 1–42. https://doi.org/10.48550/arXiv.2211.03536

Cao, L., Zhang, H., & Feng, L. (2020). Building and using personal knowledge graph to improve suicidal ideation detection on social media. *IEEE Transactions on Multimedia*, *24*, 87–102. https://doi.org/10.1109/TMM.2020.3046867

Carriero, V. A., Gangemi, A., Mancinelli, M. L., Marinucci, L., Nuzzolese, A. G., Presutti, V., & Veninata, C. (2019). ArCo: The Italian cultural heritage knowledge graph. In *The semantic Web–ISWC 2019: 18th international semantic web conference, Auckland, New Zealand, October 26–30, 2019, proceedings, part II 18* (pp. 36–52). Springer. https://doi.org/10.1007/978-3-030-30796-7_3

Chen, D., O'Bray, L., & Borgwardt, K. (2022). Structure-aware transformer for graph representation learning. In *International conference on machine learning* (pp. 3469–3489). PMLR. https://proceedings.mlr.press/v162/chen22r.html

Chen, M., Tian, Y., Yang, M., & Zaniolo, C. (2017). Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. In *Proceedings of the 26th international joint conference on artificial intelligence, IJCAI'17* (pp. 1511–1517). AAAI Press. https://doi.org/10.48550/arXiv.1611.03954

Choudhary, S., Luthra, T., Mittal, A., & Singh, R. (2021). A survey of knowledge graph embedding and their applications. *arXiv preprint arXiv:2107.07842* https://doi.org/10.48550/arXiv.2107.07842

Dou, J., Qin, J., Jin, Z., & Li, Z. (2018). Knowledge graph based on domain ontology and natural language processing technology for Chinese intangible cultural heritage. *Journal of Visual Languages & Computing*, *48*, 19–28. https://doi.org/10.1016/j.jvlc.2018.06.005

Dwivedi, V. P., & Bresson, X. (2020). A generalization of transformer networks to graphs. *ArXiv* abs/2012.09699. https://doi.org/10.48550/arXiv.2012.09699

Elekes, A., Schaeler, M., & Boehm, K. (2017). On the various semantics of similarity in word embedding models. In *2017 ACM/IEEE joint conference on digital libraries (JCDL)* (pp. 1–10). IEEE. https://doi.org/10.1109/JCDL.2017.7991568

Endres, D. M., & Schindelin, J. E. (2003). A new metric for probability distributions. *IEEE Transactions on Information Theory*, *49*(7), 1858–1860. https://doi.org/10.1109/TIT.2003.813506

Ernst, P., Siu, A., & Weikum, G. (2015). Knowlife: A versatile approach for constructing a large knowledge graph for biomedical sciences. *BMC Bioinformatics*, *16*, 1–13. https://doi.org/10.1186/s12859-015-0549-5

Euzenat, J., Roşoiu, M. E., & Trojahn, C. (2013). Ontology matching benchmarks: Generation, stability, and discriminability. *Journal of web Semantics*, *21*, 30–48. http://dx.doi.org/10.2139/ssrn.3199066

Fan, S., Wang, X., Shi, C., Cui, P., & Wang, B. (2024). Generalizing graph neural networks on out-of-distribution graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(1), 322–337. https://doi.org/10.1109/TPAMI.2023.3321097

Fanourakis, N., Efthymiou, V., Christophides, V., Kotzinos, D., Pitoura, E., & Stefanidis, K. (2023). Structural bias in knowledge graphs for the entity alignment task. In *The semantic web: 20th international conference, ESWC 2023, Hersonissos, Crete, Greece, May 28–June 1, 2023, proceedings* (pp. 72–90). Springer-Verlag. https://doi.org/10.1007/978-3-031-33455-9_5

Fanourakis, N., Efthymiou, V., Kotzinos, D., & Christophides, V. (2023). Knowledge graph embedding methods for entity alignment: An experimental review. *Data Mining and Knowledge Discovery*, *37*(5), 2070–2137. https://doi.org/10.1007/s10618-023-00941-9

Ferrara, A., Nikolov, A., & Scharffe, F. (2011). Data linking for the semantic web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, *7*(3), 46–76. https://doi.org/10.4018/jswis.2011070103

Fundulaki, I., & Ngomo, A. C. N. (2016). Instance matching benchmark for spatial data: a challenge proposal to OAEI. In *Proceedings of the 11th International Workshop on Ontology Matching (OM 2016)* (pp. 233–234). https://ceur-ws.org/Vol-1766/om2016_poster4.pdf

Giamphy, E., Guillaume, J. L., Doucet, A., & Sanchis, K. (2023). A survey on bipartite graphs embedding. *Social Network Analysis and Mining*, *13*, 54. https://doi.org/10.1007/s13278-023-01058-z

Gudivada, V., Apon, A., & Ding, J. (2017). Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations. *International Journal on Advances in Software*, *10*(1), 1–20. https://api.semanticscholar.org/CorpusID:221131081

Gulrajani, I., & Lopez-Paz, D. (2021). In search of lost domain generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=lQdXeXDoWtI

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Representation learning on graphs: Methods and applications. *IEEE Data Engineering Bulletin*, *40*, 52–74. https://doi.org/

Happi Happi, B. G., Fokou Pelap, G., Symeonidou, D., & Larmande, P. (2022). DLinker results for OAEI 2022. In *17th international workshop on ontology matching, OM 2022, CEUR workshop proceedings, CEUR-WS* (Vol. 3324, pp. 166–173). CEUR-WS.org. https://ceur-ws.org/Vol-3324/oaei22_paper6.pdf

Heylen, K., Peirsman, Y., Geeraerts, D., & Speelman, D. (2008). Modelling word similarity: An evaluation of automatic synonymy extraction algorithms. In *Proceedings of the sixth international language resources and evaluation (LCER'08)* (pp. 3243–3249). European Language Resources Association (ELRA). https://aclanthology.org/L08-1204/

Huang, J., Wang, J., Li, Y., & Zhao, W. (2022). A survey of entity alignment of knowledge graph based on embedded representation. *Journal of Physics: Conference Series*, *2171*, 012050. https://doi.org/10.1088/1742-6596/2171/1/012050

Hussain, M. S., Zaki, M. J., & Subramanian, D. (2022). Global self-attention as a replacement for graph convolution. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining* (pp. 655–665). Association for Computing Machinery. https://doi.org/10.1145/3534678.3539296

Ji, S., Pan, S., Cambria, E., Marttinen, P., & Philip, S. Y. (2021). A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(2), 494–514. https://doi.org/10.1109/TNNLS.2021.3070843

Jiang, X., Xu, C., Shen, Y., Su, F., Wang, Y., Sun, F., Li, Z., & Shen, H. (2023). Rethinking GNN-based entity alignment on heterogeneous knowledge graphs: New datasets and a new method. *arXiv preprint arXiv:2304.03468* https://doi.org/10.48550/arXiv.2304.03468

Jiménez-Ruiz, E., & Cuenca Grau, B. (2011). Logmap: Logic-based and scalable ontology matching. In *The semantic Web–ISWC 2011: 10th international semantic web conference, Bonn, Germany, October 23–27, 2011, proceedings, part I 10* (pp. 273–288). Springer. https://doi.org/10.1007/978-3-642-25073-6_18

Kejriwal, M., Sequeda, J. F., & Lopez, V. (2019). Knowledge graphs: Construction, management and querying. *Semantic Web*, *10*(6), 961–962. https://doi.org/10.3233/SW-190370

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*. https://openreview.net/forum?id=SJU4ayYgl

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86. https://doi.org/10.1214/aoms/1177729694

Larmande, P., & Todorov, K. (2021). Agrold: A knowledge graph for the plant sciences. In *The semantic web–ISWC 2021: 20th international semantic web conference, ISWC 2021, Virtual Event, October 24–28, 2021, proceedings 20* (pp. 496–510). Springer. https://doi.org/10.1007/978-3-030-88361-4_29

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., & Bizer, C. (2015). Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, *6*(2), 167–195. https://doi.org/10.3233/SW-140134

Leone, M., Huber, S., Arora, A., García-Durán, A., & West, R. (2022). A critical re-evaluation of neural methods for entity alignment. *Proceeding of the VLDB Endowment*, *15*, 1712–1725. https://doi.org/10.14778/3529337.3529355

Li, Y., Liang, X., Hu, Z., Chen, Y., & Xing, E. P. (2018). Graph transformer. https://openreview.net/forum?id=HJei-2RcK7

Liang, L., Xu, Z., Song, Z., King, I., Qi, Y., & Ye, J. (2024). Tackling long-tailed distribution issue in graph neural networks via normalization. *IEEE Transactions on Knowledge and Data Engineering*, *36*(5), 2213–2223. https://doi.org/10.1109/TKDE.2023.3315284

Lin, T., Wang, Y., Liu, X., & Qiu, X. (2022). A survey of transformers. *AI Open*, *3*, 111–132. https://doi.org/10.1016/j.aiopen.2022.10.001

Lisena, P., Achichi, M., Choffé, P., Cecconi, C., Todorov, K., Jacquemin, B., & Troncy, R. (2018). Improving (re-)usability of musical datasets: An overview of the Doremus project. *Bibliothek Forschung und Praxis*, *42*(2), 194–205. https://doi.org/10.1515/bfp-2018-0023

Liu, Z., Nguyen, T. K., & Fang, Y. (2021). Tail-GNN: Tail-node graph neural networks. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining* (pp. 1109–1119). Association for Computing Machinery. https://doi.org/10.1145/3447548.3467276

Lu, F., Cong, P., & Huang, X. (2020). Utilizing textual information in knowledge graph embedding: A survey of methods and applications. *IEEE Access*, *8*, 92072–92088. https://doi.org/10.1109/ACCESS.2020.2995074

Luo, X., Sun, Z., & Hu, W. (2022). μKG: A library for multi-source knowledge graph embeddings and applications. In *International semantic web conference* (pp. 610–627). Springer. https://doi.org/10.1007/978-3-031-19433-7_35

Malekzadeh Hamedani, E., & Kaedi, M. (2019). Recommending the long tail items through personalized diversification. *Knowledge-Based Systems*, *164*, 348–357. https://doi.org/10.1016/j.knosys.2018.11.004

Mao, X., Wang, W., Xu, H., Wu, Y., & Lan, M. (2020). Relational reflection entity alignment. In *Proceedings of the 29th ACM international conference on information & knowledge management* (pp. 1095–1104). Association for Computing Machinery. https://doi.org/10.1145/3340531.3412001

Marchand, E., Gagnon, M., & Zouaq, A. (2020). Extraction of a knowledge graph from French cultural heritage documents. In *ADBIS, TPDL and EDA 2020 common workshops and doctoral consortium: International Workshops: DOING, MADEISD, SKG, BBIGAP, SIMPDA, AIMinScience 2020 and Doctoral Consortium, Lyon, France, August 25–27, 2020, proceedings 24* (pp. 23–35). Springer. https://doi.org/10.1007/978-3-030-55814-7_2

Min, E., Chen, R., Bian, Y., Xu, T., Zhao, K., Huang, W., Zhao, P., Huang, J., Ananiadou, S., & Rong, Y. (2022). Transformer for graphs: An overview from architecture perspective. *arXiv preprint arXiv:2202.08455* https://doi.org/10.48550/arXiv.2202.08455

Müller, L., Galkin, M., Morris, C., & Rampášek, L. (2023). Attending to graph transformers. *arXiv:2302.04181* https://openreview.net/forum?id=HhbqHBBrfZ

Nguyen, D. Q., Nguyen, T. D., & Phung, D. (2022). Universal graph transformer self-attention networks. In *Companion proceedings of the web conference 2022* (pp. 193–196). Association for Computing Machinery. https://doi.org/10.1145/3487553.3524258

Nicholson, D. N., & Greene, C. S. (2020). Constructing knowledge graphs and their biomedical applications. *Computational and Structural Biotechnology Journal*, *18*, 1414–1428. https://doi.org/10.1016/j.csbj.2020.05.017

Pfeiffer, J., Goyal, N., Lin, X., Li, X., Cross, J., Riedel, S., & Artetxe, M. (2022). Lifting the curse of multilinguality by pre-training modular transformers. In M. Carpuat, M. C. de Marneffe, & I. V. Meza Ruiz (Eds.), *Proceedings of the 2022 conference of the North American Chapter of the Association for Computational Linguistics: Human language technologies* (pp. 3479–3495). Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.naacl-main.255

Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data lifecycle challenges in production machine learning: A survey. *ACM SIGMOD Record*, *47*(2), 17–28. https://doi.org/10.1145/3299887.3299891

Roelofs, R. (2019). *Measuring generalization and overfitting in machine learning* [PhD thesis, University of California, Berkeley].

Ryen, V., Soylu, A., & Roman, D. (2022). Building semantic knowledge graphs from (semi-) structured data: A review. *Future Internet*, *14*(5), 129. https://doi.org/10.3390/fi14050129

Saha, A., Pahuja, V., Khapra, M., Sankaranarayanan, K., & Chandar, S. (2018). Complex sequential question answering: Towards learning to converse over linked question answer pairs with a knowledge graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, *32*(1), 11332. https://doi.org/10.1609/aaai.v32i1.11332

Salazar, R. C., Jonquet, C., & Symeonidou, D. (2023). Classification of linking problem types for linking semantic data. In *SEMANTICS 2023* (Vol. 56, pp. 194–209). https://doi.org/10.3233/SSW230014

Sanou, G., Giudicelli, V., Abdollahi, N., Kossida, S., Todorov, K., & Duroux, P. (2022). IMGT-KG: A knowledge graph for immunogenetics. In *International semantic web conference* (pp. 628–642). Springer. https://doi.org/10.1007/978-3-031-19433-7_36

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605

Sharma, A., & Talukdar, P. (2018). Towards understanding the geometry of knowledge graph embeddings. In *Proceedings of the 56th annual meeting of the Association for Computational Linguistics (Volume 1: Long papers)* (pp. 122–131). Association for Computational Linguistics. https://doi.org/10.18653/v1/P18-1012

Shen, J., Wang, C., Gong, L., & Song, D. (2022). Joint language semantic and structure embedding for knowledge graph completion. In *Proceedings of the 29th international conference on computational linguistics* (pp. 1965–1978). International Committee on Computational Linguistics. https://aclanthology.org/2022.coling-1.171/

Shi, L. (2013). Trading-off among accuracy, similarity, diversity, and long-tail: A graph-based recommendation approach. In *Proceedings of the 7th ACM conference on recommender systems* (pp. 57–64). Association for Computing Machinery. https://doi.org/10.1145/2507157.2507165

Shvaiko, P., & Euzenat, J. (2011). Ontology matching: State of the art and future challenges. *IEEE Transactions on Knowledge and Data Engineering*, *25*(1), 158–176. https://doi.org/10.1109/TKDE.2011.253

Sun, Z., Hu, W., & Li, C. (2017). Cross-lingual entity alignment via joint attribute-preserving embedding. In *The semantic web–ISWC 2017: 16th international semantic web conference, Vienna, Austria, October 21–25, 2017, proceedings, part I 16* (pp. 628–644). Springer. https://doi.org/10.1007/978-3-319-68288-4_37

Sun, Z., Hu, W., Zhang, Q., & Qu, Y. (2018). Bootstrapping entity alignment with knowledge graph embedding. In *Proceedings of the 27th international joint conference on artificial intelligence, IJCAI'18* (pp. 4396–4402). AAAI Press. https://doi.org/10.24963/ijcai.2018/611

Sun, Z., Huang, J., Hu, W., Chen, M., Guo, L., & Qu, Y. (2019). Transedge: Translating relation-contextualized embeddings for knowledge graphs. In C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, & F. Gandon (Eds.), *The semantic web—ISWC'2019* (pp. 612–629). Springer International Publishing. https://doi.org/10.1007/978-3-030-30793-6_35

Sun, Z., Zhang, Q., Hu, W., Wang, C., Chen, M., Akrami, F., & Li, C. (2020). A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceeding of the VLDB Endowment*, *13*(12), 2326–2340. https://doi.org/10.14778/3407790.3407828

Tang, X., Zhang, J., Chen, B., Yang, Y., Chen, H., & Li, C. (2020). BERT-INT: A BERT-based interaction model for knowledge graph alignment. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI'20* (pp. 3174–3180). AAAI Press. https://doi.org/10.24963/ijcai.2020/439

Tchechmedjiev, A., Fafalios, P., Boland, K., Gasquet, M., Zloch, M., Zapilko, B., Dietze, S., & Todorov, K. (2019). Claimskg: A knowledge graph of fact-checked claims. In *The semantic Web–ISWC 2019: 18th international semantic web conference, Auckland, New Zealand, October 26–30, 2019, proceedings, part II 18* (pp. 309–324). Springer. https://doi.org/10.1007/978-3-030-30796-7_20

Todorov, K. (2019). Datasets first! A bottom-up data linking paradigm. In *ISWC (satellites)* (pp. 338–342). CEUR-WS.org. https://ceur-ws.org/Vol-2456/paper95.pdf

Tran, H. N., & Takasu, A. (2019). Analyzing knowledge graph embedding methods from a multi-embedding interaction perspective. *arXiv preprint arXiv:1903.11406* https://doi.org/10.48550/arXiv.1903.11406

Traverso, I., Vidal, M. E., Kämpgen, B., & Sure-Vetter, Y. (2016). GADES: A graph-based semantic similarity measure. In *Proceedings of the 12th international conference on semantic systems* (pp. 101–104). Association for Computing Machinery. https://doi.org/10.1145/2993318.2993343

Trisedya, B. D., Salim, F. D., Chan, J., Spina, D., Scholer, F., & Sanderson, M. (2023). i-Align: An interpretable knowledge graph alignment model. *Data Mining and Knowledge Discovery*, *37*, 2494–2516. https://doi.org/10.1007/s10618-023-00963-3

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning* (pp. 2071–2080). PMLR. https://proceedings.mlr.press/v48/trouillon16.html

Unni, D. R., Moxon, S. A., Bada, M., Brush, M., Bruskiewich, R., Caufield, J. H., Clemons, P. A., Dancik, V., Dumontier, M., Fecho, K., Glusman, G., Hadlock, J. J., Harris, N. L., Joshi, A., Putman, T., Qin, G., Ramsey, S. A., Shefchek, K. A., & Solbrig, H., ... Mungal, C. J. (2022). Biolink model: A universal schema for knowledge graphs in clinical, biomedical, and translational science. *Clinical and Translational Science*, *15*(8), 1848–1855. https://doi.org/10.1111/cts.13302

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*(11), 2579–2605. https://www.jmlr.org/papers/v9/vandermaaten08a.html

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*, *NIPS'17* (Vol. 30, pp. 6000–6010). Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Venkatesan, A., Tagny Ngompé, G., Hassouni, N. E., Chentli, I., Guignon, V., Jonquet, C., Ruiz, M., & Larmande, P. (2018). Agronomic linked data (AgroLD): A knowledge-based system to enable integrative biology in agronomy. *PLoS One*, *13*(11), e0198270. https://doi.org/10.1371/journal.pone.0198270

Villazón-Terrazas, B., Ortiz-Rodríguez, F., Tiwari, S. M., & Shandilya, S. K. (2020). Knowledge graphs and semantic web. *Communications in Computer and Information Science*, *1232*, 1–225. https://doi.org/10.1007/978-3-030-65384-2

Vrandečić, D., & Krötzsch, M. (2014). Wikidata: A free collaborative knowledgebase. *Communications of the ACM*, *57*(10), 78–85. https://doi.org/10.1145/2629489

Wang, B., Wang, A., Chen, F., Wang, Y., & Kuo, C. C. J. (2019). Evaluating word embedding models: Methods and experimental results. *APSIPA Transactions on Signal and Information Processing*, *8*, e19. https://doi.org/10.1017/ATSIP.2019.12

Wang, C., Huang, Z., Wan, Y., Wei, J., Zhao, J., & Wang, P. (2023). Fualign: Cross-lingual entity alignment via multi-view representation learning of fused knowledge graphs. *Information Fusion*, *89*, 41–52. https://doi.org/10.1016/j.inffus.2022.08.002

Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743. https://doi.org/10.1109/TKDE.2017.2754499

Wang, Y., Wang, D., Liu, H., Hu, B., Yan, Y., Zhang, Q., & Zhang, Z. (2024). Optimizing long-tailed link prediction in graph neural networks through structure representation enhancement. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining, KDD '24* (pp. 3222–3232). Association for Computing Machinery. https://doi.org/10.1145/3637528.3671864

Wang, Z., Chen, T., Ren, J., Yu, W., Cheng, H., & Lin, L. (2018). Deep reasoning with knowledge graph for social relationship understanding. *arXiv preprint arXiv:1807.00504* https://doi.org/10.24963/ijcai.2018/142

Wang, Z., Lv, Q., Lan, X., & Zhang, Y. (2018). Cross-lingual knowledge graph alignment via graph convolutional networks. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 349–357). Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1032

Wu, Y., Liu, X., Feng, Y., Wang, Z., Yan, R., & Zhao, D. (2019). Relation-aware entity alignment for heterogeneous knowledge graphs. In *Proceedings of the Twenty-Eighth international joint conference on artificial intelligence, IJCAI'19* (pp. 5278–5284). Curran Associates, Inc. https://doi.org/10.24963/ijcai.2019/733

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Xu, K., Wang, L., Yu, M., Feng, Y., Song, Y., Wang, Z., & Yu, D. (2019a). Cross-lingual knowledge graph alignment via graph matching neural network. In A. Korhonen, D. Traum, & L. Màrquez (Eds.), *Proceedings of the 57th annual meeting of the Association for Computational Linguistics* (pp. 3156–3161). Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1304

Yang, K., Liu, S., Zhao, J., Wang, Y., & Xie, B. (2020). COTSAE: Co-training of structure and attribute embeddings for entity alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, *34*(3), 3025–3032. https://doi.org/10.1609/aaai.v34i03.5696

Yujian, L., & Bo, L. (2007). A normalized levenshtein distance metric. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *29*(6), 1091–1095. https://doi.org/10.1109/TPAMI.2007.1078

Zeng, K., Li, C., Hou, L., Li, J., & Feng, L. (2021). A comprehensive survey of entity alignment for knowledge graphs. *AI Open*, *2*, 1–13. https://doi.org/10.1016/j.aiopen.2021.02.002

Zeng, W., Zhao, X., Wang, W., Tang, J., & Tan, Z. (2020). Degree-aware alignment for entities in tail. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in information retrieval* (pp. 811–820). ACM. https://doi.org/10.1145/3397271.3401161

Zhang, J., Zhang, H., Xia, C., & Sun, L. (2020). Graph-BERT: Only attention is needed for learning graph representations. *arXiv preprint arXiv:2001.05140* https://doi.org/10.48550/arXiv.2001.05140

Zhang, Q., Sun, Z., Hu, W., Chen, M., Guo, L., & Qu, Y. (2019). Multi-view knowledge graph embedding for entity alignment. In *Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI'19* (pp. 5429-5435). AAAI Press. https://doi.org/10.24963/ijcai.2019/754

Zhang, R., Trisedya, B. D., Li, M., Jiang, Y., & Qi, J. (2022). A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. *The VLDB Journal*, *31*(5), 1143–1168. https://doi.org/10.1007/s00778-022-00747-z

Zhang, Z., Liu, X., Zhang, Y., Su, Q., Sun, X., & He, B. (2020). Pretrain-KGE: Learning knowledge representation from pretrained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020* (pp. 259–266). Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.findings-emnlp.25

Zhao, X., Jia, Y., Li, A., Jiang, R., & Song, Y. (2020). Multi-source knowledge fusion: A survey. *World Wide Web*, *23*, 2567–2592. https://doi.org/10.1007/s11280-020-00811-0

Zhao, X., Zeng, W., Tang, J., Wang, W., & Suchanek, F. M. (2020). An experimental study of state-of-the-art entity alignment approaches. *IEEE Transactions on Knowledge and Data Engineering*, *34*(6), 2610–2625. https://doi.org/10.1109/TKDE.2020.3018741

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI open*, *1*, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhu, G., & Iglesias, C. A. (2016). Computing semantic similarity of concepts in knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, *29*(1), 72–85. https://doi.org/10.1109/TKDE.2016.2610428

Zhu, H., Xie, R., Liu, Z., & Sun, M. (2017). Iterative entity alignment via joint knowledge embeddings. In *Proceedings of the 26th international joint conference on artificial intelligence, IJCAI'17* (pp. 4258–4264). AAAI Press. https://doi.org/10.24963/ijcai.2017/595

Zou, X. (2020). A survey on application of knowledge graph. *Journal of Physics: Conference Series*, *1487*, 012016. https://doi.org/10.1088/1742-6596/1487/1/012016