A Critical Re-evaluation of Neural Methods for Entity Alignment

Manuel Leone[*]
EPFL
manuel.leone@epfl.ch

Stefano Huber[*]
EPFL
stefano.huber@epfl.ch

Akhil Arora[*][†]
EPFL
akhil.arora@epfl.ch

Alberto García-Durán[*]
EPFL
agaduran@gmail.com

Robert West
EPFL
robert.west@epfl.ch

ABSTRACT

Neural methods have become the de-facto choice for the vast majority of data analysis tasks, and entity alignment (EA) is no exception. Not surprisingly, more than 50 different neural EA methods have been published since 2017. However, surprisingly, an analysis of the differences between neural and non-neural EA methods has been lacking. We bridge this gap by performing an in-depth comparison among five carefully chosen representative state-of-the-art methods from the pre-neural and neural era. We unravel, and consequently mitigate, the inherent deficiencies in the experimental setup utilized for evaluating neural EA methods. To ensure fairness in evaluation, we homogenize the entity matching modules of neural and non-neural methods. Additionally, for the first time, we draw a parallel between EA and record linkage (RL) by empirically showcasing the ability of RL methods to perform EA. Our results indicate that Paris, the state-of-the-art non-neural method, statistically significantly outperforms all the representative state-of-the-art neural methods in terms of both efficacy and efficiency across a wide variety of dataset types and scenarios, and is second only to BERT-INT for a specific scenario of cross-lingual EA. Our findings shed light on the potential problems resulting from an impulsive application of neural methods as a panacea for all data analytics tasks. Overall, our work results in two overarching conclusions: (1) Paris should be used as a baseline in every follow-up work on EA, and (2) neural methods need to be positioned better to showcase their true potential, for which we provide multiple recommendations.

PVLDB Reference Format:
Manuel Leone, Stefano Huber, Akhil Arora, Alberto García-Durán, and Robert West. A Critical Re-evaluation of Neural Methods for Entity Alignment. PVLDB, 15(8): 1712-1725, 2022.
doi:10.14778/3529337.3529355

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/epfl-dlab/entity-matchers.

1 INTRODUCTION

The past decade—a.k.a. the neural era—has witnessed a substantial amount of research on representation learning for Knowledge Graphs (KGs) [52]. KGs store knowledge in the form of the so-called facts or triples (e.g. (*head entity*, *relationship*, *tail entity*)) and have been shown to be essential for enhancing several downstream applications (e.g. question answering [42], intelligent conversational agents [57], and recommender systems [73]). A common characteristic across most, if not all, of the KGs is that they are seldom complete, i.e., they lack a large portion of knowledge. To this end, the vast majority of research over the past decade has focused on KG completion via embedding-based methods for link prediction [30, 48]. These techniques embed the entities and relationships of the KG as low-dimensional vectors, which are used in conjunction with a scoring function to predict unseen links between pairs of entities for which the obtained score is high. Another prevalent approach to increase the completeness of a KG is termed *entity alignment* (EA), a fundamental task in the broad field of data integration. Instead of predicting missing links within a KG, the EA task consists of integrating two or more KGs into the same source of knowledge by aligning nodes that refer to the same entity/concept. This resembles the join operation in relational databases (DBs), where records from two or more tables are combined based on some common attributes. Different to the join operation, in the EA problem the entities of two or more KGs are not represented by unique distinct identifiers that are shared across all the KGs. That said, given a way to handle possibly heterogeneous KG schemas, the EA task can be seen as a record linkage (RL) problem for integrating tables from two different relational DBs.

Motivated by the success of embedding-based methods for link prediction, researchers adapted these methods to address the EA problem. The motivation is sound: if the neighboring structure and the values of the attributes of counterpart entities from different KGs are similar, then their low-dimensional representations should also be similar. However, the EA problem did not emerge recently, rather it is a well-studied and fundamental task in the data management and semantic web communities (e.g. [8, 31, 35, 59], to name but a few) that predates the neural era. Unfortunately, most of the embedding-based (neural) EA methods lack a comparison to the methods from the pre-neural era, and only benchmark against other neural methods. This *endogamic behavior* impedes our understanding about the *true progress* achieved by neural EA methods with respect to the solid and vetted contributions made by methods from the pre-neural era and other communities.

[*] Equal contribution.
[†] Corresponding author.

Proceedings of the VLDB Endowment, Vol. 15, No. 8 ISSN 2150-8097.
doi:10.14778/3529337.3529355

Given the success of representation learning in multiple fields such as computer vision or natural language processing, one might expect substantial progress on account of neural EA methods, however, this has to be validated empirically. As a matter of fact, a recent work [18, 19] in the field of recommender systems has revealed the superiority of naïve baselines over the vast majority of neural approaches published at top research venues in the last years. In the same vein, a *critical re-evaluation* of neural EA methods is warranted. Given the increasing interest in the development of neural EA methods, an exhaustive comparison to methods from a different family will only help to provide a clearer picture of the progress made recently in this problem, thereby serving as a timely addition to research in the field. That said, the goal of this work is to shed light on three main research questions, which are stated as follows:

- **RQ1:** *Is the evaluation setup employed by neural EA methods meaningful?*
- **RQ2:** *What is the true progress achieved by neural EA methods?*
- **RQ3:** *What lies in the future for the field of neural EA?*

Scope of the present work. Before we list our contributions, it is important to explicitly clarify the scope of this work. The primary goal of this work is to assess and disseminate the *true progress* achieved in the field on account of neural EA methods. That said, *exhaustively benchmarking* recent EA methods as performed by related works [66, 81, 82], was neither originally in the scope of this work nor should it be perceived as such. Conversely, measuring true progress requires a systematic, solid, assumption-free, and realistic evaluation of the *representative* methods on a large collection of datasets with diverse and realistic characteristics. To this end, we redirect our focus on specific aspects of the EA task captured by the aforementioned three research questions (RQ1-3). Further, we make multiple nuanced contributions (detailed below), which, on their own, should only be perceived as by-products of this work, however, are crucial collectively to achieve our primary goal.

Key contributions. The main contributions of this work are:

- We extend Paris [59]—an established and vetted method from the pre-neural era—to incorporate supervised seed alignment information (§ 3.1). This extension is important for performing a fair comparison of Paris, which is originally unsupervised, with neural methods that are supervised by design.
- We answer RQ1 by identifying and rectifying unrealistic assumptions followed by methods from the neural era. Specifically, we create more than 10 novel benchmark datasets that closely mimic real-world EA scenarios, and introduce a homogenized evaluation protocol for a fair comparison across all methods (§ 3).
- We answer RQ2 by performing extensive empirical evaluations across a wide-range of dataset configurations and evaluation metrics, and successfully establish the strength of the methods from the pre-neural era. Additionally, for the first time, we draw a parallel between EA and RL by empirically showcasing the ability of RL methods to perform EA (§ 4, § 5, and § 6).
- We answer RQ3 by providing actionable recommendations to the community for enabling a streamlined advancement of research in neural EA. We lead by example and enhance a neural method with a feature identified from our analysis as fundamental to the success of non-neural methods (§ 7).

2 ENTITY ALIGNMENT: A PRIMER

Preliminaries. Following recent work [66], we refer to (*head*, *relation*, *tail*) and (*head*, *attribute*, *literal*) as *relation* and *attribute* triples, respectively. Instances of both types of triples are (Madrid, is_capital_of, Spain) and (Madrid, has_latitude, "40.4"), respectively. The arguments *head* and *tail* represent entities, *relation* is a relationship that holds between two entities, and *attribute* is a type of relationship that holds between an entity and a *literal.* As opposed to entities, which are language-agnostic representation of concepts, literals are used to identify values for strings, numbers or dates. Therefore, literals are written in a certain language or following a certain format, both defined by the schema of the KG. We note that the referred terminology is not shared by past works [35, 59], where, for instance, triples and attributes are termed statements and properties, respectively. Therefore, a KG is characterized with a number of triples $\mathcal{E} \times \mathcal{R} \times (\mathcal{E} \cup \mathcal{L})$, where $\mathcal{E}$, $\mathcal{R}$, and $\mathcal{L}$ indicate the set of entities, relationships, and literals, respectively. While attribute triples are a fundamental part of every KG, some recent entity alignment methods are built upon the assumption that the KG is solely characterized with relation triples. We coin the term *shallow KG* to refer to this type of KG. We sometimes use the shortcut $r(a, b)$ to indicate that a relationship r holds between an entity a and a second argument b (either an entity or a literal).

The entity alignment problem is typically defined between two KGs, $\mathcal{KG}_1$ and $\mathcal{KG}_2$, where the task consists of finding equivalences between the set of entities $\mathcal{E}_1$ and $\mathcal{E}_2$ of the two KGs. Sometimes there exists a set of known alignments that can be used as supervision. This set $\mathcal{S}$ is known as *seed alignment.* We always assume that there exists a ground truth $\mathcal{G} = \{(e, e') \in \mathcal{E}_1 \times \mathcal{E}_2 | e = e'\}$ that includes all possible equivalences between pairs of entities.

Related research problems. The success of the Semantic Web [24] was key to the proliferation of ontologies/schemas and knowledge graphs[1]. This motivated the interest in methods that helped to integrate independently designed ontologies and KGs, and led to the following three broad directions:

(a) *Schema Alignment* [6, 27, 29, 72] aims to identify equivalent classes between pairs of ontologies. Although it shares some similarities with the EA problem, one fundamental difference between both tasks relies on the number of elements to be matched. This becomes the main obstacle for transferring successful ontology alignment methods to the EA problem.

(b) *Ontology Alignment* [38, 59, 70] refers to holistic solutions that address the more general problem of both instance (i.e. entities and relationships) and schema alignment.

(c) *Entity Matching (a.k.a. record linkage or entity resolution)* [16, 33, 39, 46] is the counterpart problem to EA for relational DBs. It refers to the problem of determining whether data entries from two DBs refer to the same real-world entity. For the sake of clarity, we will use the term RL to refer to these methods in the remainder of the paper. We study the connection between RL and EA in detail in § 5.

[1] While the distinction between these two terms is sometimes blurry, one may think of a schema as a formal description of knowledge—a set of classes (e.g. the class of all movies, actors, etc.) within a domain and the relations that hold between them—which enables the generation of instantiations of knowledge—a knowledge graph.

It is important to highlight the Ontology Alignment Evaluation Initiative (OAEI) [2] that runs yearly campaigns—soliciting proposals across a variety of tracks followed by a publication of the findings [22, 40, 54]—to track the progress on schema and ontology alignment. Similar efforts should be explored by the EA community, primarily to ensure a homogeneous evaluation setup.

2.1 Pre-neural Era

Entity alignment for KGs has been an active research topic in the data mining and semantic web communities since 2000. These approaches have been typically benchmarked on real KGs such as DBpedia [5] or Freebase [9]. Popular EA methods for large-scale KGs included efficient solutions to the quadratic scaling cost of comparing all possible pairs of entities [8, 35]. Other popular solutions were based on rules [4], clustering techniques [88], or consisted of principled frameworks to scale any generic EA algorithm [56].

Similar to ontology alignment, there is an additional problem that solves EA as a by-product of a more general task, called instance alignment, where the goal is to align both entities and relationships between two ontologies. Some of the proposed instance alignment methods leveraged terminological structure [49], exploited a set of heuristics [69] or were based on relational clustering techniques [7]. Paris [59], one of the most popular ontology alignment methods, is a probabilistic method that neither requires any seed alignment (i.e. it is unsupervised) nor needs any parameter tuning. We introduce some technical concepts of Paris that are pivotal to some subsequent sections of this work.

Functionality is a key concept to understand Paris. A relationship is said to be (quasi) functional if, for a given entity, the expected number of entities or literals that complete the triple is (close to) one. Examples of (quasi) functional relationships are born_in or height. Formally, it is defined as follows:

$$fun(r) = \frac{\#x : \exists y : r(x, y)}{\#x, y : r(x, y)}. \qquad (1)$$

While we omit many technical details for the sake of simplicity, we note that the bulk of Paris boils down to estimating the probabilities of an entity e in $\mathcal{KG}_1$ being equivalent to another entity e' in $\mathcal{KG}_2$.

$$Pr(e \equiv e') = 1 - \prod_{\substack{r(e,y) \\ r(e',y')}} (1 - fun(r^{-1}) Pr(y \equiv y')). \qquad (2)$$

For the probability of $e \equiv e'$ to be large, the entities e and e' *only* need one (quasi) functional relationship that connects them to two entities (or literals) y and y', respectively, with a high equivalence probability. Therefore, Paris operates recursively: the equivalence probability of $e \equiv e'$ depends recursively on other equivalence probabilities. Initial equivalences are computed between literals based on a certain string distance.

2.2 Neural Era

After an intense period of research—with a peak around 2010—the topic became less active and did not experience any breakthrough for a period of time. However, a new wave of interest was initiated in 2017. As opposed to the previous wave, where we witnessed the development of a large variety of techniques, this era was dominated by neural approaches. A second key difference is that most approaches of this new era focus solely on the EA problem. A large number of these neural approaches are adaptations of successful embedding-based methods for link prediction. To our knowledge, the work by Chen et al. [15] pioneered this family of solutions. Their approach, referred to as MTransE, is inspired by translational models for link prediction such as TransE [11] and rTransE [26]. Surprisingly, the authors did not benchmark MTransE against any of the previous approaches that may work under the exact same setup, but their baselines consist of methods from the multilingual word embedding literature such as CCA [23], LM [45] and OT [76]. MTransE compared favorably with these baselines, and became the state-of-the art for successive neural approaches.

Motivated by the recent surge of neural EA methods at top-tier AI conferences (e.g. [12, 14, 15, 53, 63, 64, 68, 74, 75, 77, 80, 84]), some very recent works [66, 81, 82] have provided a benchmarking study of these methods. Such works [66, 82] also depicted that most of the neural approaches can be described as instances of a modular framework. The main modules of the framework are:

- The *embedding module* encodes the components (i.e. entities and relations) of the KG into a low-dimensional space. While for shallow KGs the encoding function typically consists of a simple lookup table, for normal KGs the encoding function learns low-dimensional representations for entities by exploiting their literal values and those of their neighbors.
- The *interaction module* leverages a supervised signal (i.e. the seed alignment) to guide the optimization of an objective function. The module ensures that all components of the two KGs are projected into the same latent space. The interaction module also computes alignment scores between pairs of entities based on an appropriate scoring/similarity/distance function.
- The alignment scores are used as input to the *matching module*, which addresses the combinatorial optimization problem known as the assignment problem [34]. While more sophisticated—and also more costly—solutions to this problem are possible, it very often follows a greedy strategy to output equivalences between entities. In addition, boostrapping is a practice followed by some works: The alignments outputted in an iteration are used as supervision by the interaction module in subsequent iterations.

The aforementioned modules are common to all the neural techniques, regardless of whether they are designed for normal or shallow KGs. With the exception of the *matching module*, which is also present in methods from the pre-neural era, the other two modules are only relevant to neural approaches. Thus, fundamental differences across neural methods emanate primarily from the embedding and interaction modules (cf. [66, 82] for details).

2.3 Recent Benchmarking Studies

The proliferation of neural EA methods led to multiple recent benchmarking studies—published in the later half of 2020—that systematically evaluate and compare their performance.

- To our knowledge, the work by Sun et al. [66] provided the first in-depth analysis and comparison of neural EA methods. Sun et al. also developed and released OpenEA, an open-source library [61] comprising an implementation of 13 recent EA methods. OpenEA is duly maintained and integrates the source code of additional EA methods to facilitate their evaluation. It also incorporates a

Table 1: A summarized comparison of the contributions of recent benchmarking studies on entity alignment.

Contribution Type	Description	Present work (Section)	Zhao et al. [82]	Sun et al. [66]
Evaluation	Realistic datasets: degree distribution	✓ (3.2)	✓	✓
	Realistic datasets: no 1-to-1 assumption	✓ (3.2)	✓*	
	Realistic datasets: obfuscated entity URIs	✓ (3.2)		
	Benchmark datasets for ablation analyses	✓ (3.2)		
	Homogenized matching module	✓ (3.3)		
	Realistic evaluation metrics	✓ (3.3)	✓*	
Empirical	In-depth comparison: Paris vs. neural methods	✓ (4)		
	Comparison to record linkage methods	✓ (5)	✓°	
Methodological★	Extending Paris to leverage seed alignment	✓ (3.1)		
	Extending neural methods with ideas from non-neural methods	✓ (7.2)		

* The discussion and analysis was carried out on only a single dataset

° The comparison included name-based record linkage heuristics

★ Existing benchmarking studies did not propose any methodological extensions

set of dedicated small-scale benchmark datasets (cf. § 3.2). These datasets are sub-sampled versions of well-known KGs designed to better reflect the properties of original KGs. We use the term OpenEA to refer to these datasets in the remainder of the work.

- Next, similar to [66], Zhao et al. [82] conducted a systematic and comprehensive comparison of neural methods.
- The deployment of neural EA methods in industrial setups was discussed by Zhang et al. in [81]. Seed alignments in the industrial setup are scarce and less biased when compared to the ideal academic setup. Not surprisingly, Zhang et al. report that the performance of neural methods declines drastically when the evaluation context is moved from the ideal to the industrial setup.

All these works are partially complementary with respect to the set of evaluated neural methods and the benchmark datasets, however collectively, they provide a clear picture of the best-performing neural approaches published until, at least, mid-2020. Interestingly, all these works *conclude with a very superficial comparison to Paris.*

Novelty of the present work. We share a similar goal with the aforementioned benchmarking studies (especially [66, 82]), namely, we want to gain a better understanding of the EA problem and better position the progress made in the neural era.

To achieve this goal, previous studies perform an exhaustive comparison across neural methods, but only a tangential and superficial analysis with respect to Paris. On the contrary, Paris is treated as a first-class citizen (along with other representative methods) in the analysis performed in this work. More specifically, the findings of previous studies simply serve as the starting point of the analysis conducted in this work, i.e., facilitating the selection of representative EA methods. That said, this paper goes beyond conventional benchmarking, and performs a wide-variety of in-depth analyses to unravel the technical and practical underpinnings of the EA problem. The key novel research directions, which unfortunately were not explored in the previous studies, are stated as follows:

- Achieving an evaluation setup that closely mimics real-world settings. This not only involves the creation of a wide-variety of realistic datasets but also the adoption of suitable evaluation metrics, and the homogenization of design choices (e.g. the matching module) that facilitate a fair comparison across methods.
- Ablations to analyze how specific characteristics of KGs (e.g. #attributes, KG sparsity, or amount of supervision) may affect methods from both the pre-neural and neural era.

Table 2: (Top) Representative methods benchmarked in this work, and (bottom) their key characteristics.

Era	KGs w/ attribute triples	KGs w/o attribute triples
Pre-neural	Paris+ (this work)	Paris+ (this work)
Neural	RDGCN [74], BERT-INT [67]	BootEA [64], TransEdge [65]

Method	Key characteristics
Paris [59]	Probabilistic approach relying on the functionality of the relationships.
Paris+ (this work)	Extension of Paris to leverage the seed alignment.
RDGCN [74]	Embedding module based on GCN [32].
BERT-INT [67]	Embedding module based on BERT [20]. Entities are characterized with descriptions.
BootEA [64]	Link prediction objective function. Bootstrapping procedure to iteratively extend the seed alignment.
TransEdge [65]	It extends BootEA to also optimize an entity alignment objective.

- Providing several recommendations to improve and/or better position future work on neural entity alignment. Importantly, we already empirically show the benefit obtained by incorporating one of the recommendations to extend neural EA methods.

Table 1 concludes this discussion by providing an actionable summary as well as a qualitative comparison of specific contributions made by this work and the previous benchmarking studies.

3 EXPERIMENTAL SETUP

Based on our careful review of the EA literature, we observed that methods belonging to the pre-neural and neural era differ not only at the technical level but also in their evaluation. In this section, we (i) provide a detailed description of multiple salient discrepancies in their evaluation setups, and (ii) discuss our contributions towards obtaining a realistic and homogenized evaluation setup, thereby mitigating the said discrepancies.

3.1 On the Choice of Representative Methods

We start by enumerating and justifying the selection of methods benchmarked in this work. Table 2 lists the chosen techniques, which are meant to enable a meaningful comparison in setups where (i) the KGs possess attribute triples, and (ii) the KGs lack attribute triples (shallow KGs).

Why Paris? To the best of our knowledge, there does not exist any work from the pre-neural era that significantly beats Paris on the datasets where Paris was benchmarked—YAGO, DBpedia and IMDb. However, if we only focus on the EA task—a subset of the outcome of Paris—there do exist works that report similar performance while being more efficient. One such example is SIGMa [35], which comes at the cost of requiring a (small) set of manually aligned relationships and attributes. Additional strengths of Paris are that it is publicly available [21] and is extremely easy to use.

What is Paris+? It is our variant of Paris that works even in the absence of attribute triples, as long as there exists a seed set. Paris exploits the information in the attribute triples to initially compute probabilities of pairs of literals being the same, which serve as the basis for estimating equivalence probabilities between entities. Thus, Paris is destined to fail in the absence of attribute triples. However, Paris+ processes the seed information to generate attribute triples such that for every pair of entities (*e*, *e*') that are part of the the seed alignment, it creates the attribute triples (*e*, *EA:label*, `string`(*e*)) and (*e*', *EA:label*, `string`(*e*)), where

EA:label is a synthetic relationship that connects an entity to a quoted literal—following the RDF format [1]–obtained through the function `string`. Thus, the relationship *EA:label* is designed to be *highly functional* (Eq. 1) and, consequently, the entity pair (*e*, *e*') will be deemed equivalent (Eq. 2) with a very high likelihood.

With this added ability to leverage seed alignments (if any), Paris+ becomes the *only representative method from the pre-neural era* in the remainder of the paper. A comparison between Paris and Paris+ can be found in the appendix of our technical report [36].

Why RDGCN, BERT-INT, BootEA and TransEdge? We carefully reviewed the benchmarking studies (cf. § 2.3) and other papers published until March 2021. We rely upon experiments reported in these works on sub-sampled versions of KGs derived from DBpedia, Yago and Wikidata. From the benchmarking studies [66, 82] published in the later half of 2020, we concluded that RDGCN was always very competitive—often the best performing neural technique—for KGs that possess attribute triples. Also from these works we concluded that for shallow KGs the best performing techniques are BootEA and TransEdge. We continued tracking subsequent papers published at top-tier conferences, and decided to include BERT-INT in our selection of representative neural methods. The reason is twofold: (i) to our knowledge, to date BERT-INT outperforms all existing neural methods on a collection of datasets [67] derived from DBpedia, (ii) it facilitates assessing the strength of the highly successful language model—BERT—on the EA task.

We follow the best practices in [66] to enhance the performance (e.g. using the cross-domain similarity local scaling (CSLS) [17] as a similarity metric in the interaction module) of these methods.

Why not other methods? To the best of our knowledge, there does not exist any method in the literature from either the pre-neural or neural era that is significantly better than the chosen techniques in datasets generated from the KGs considered in this work. Next, we provide specific reasons behind the omission of some recently published neural EA methods.

- While CEA [79] was shown to outperform other more elaborate methods [82], we excluded CEA from this study for two reasons: (i) it is outperformed by BERT-INT [67], and (ii) we hypothesized that the observed gains in performance over related works is due to a complex and expensive matching module. As we will discuss later in this section, for a fair comparison, we deliberately homogenize the matching module of the selected methods.
- MuGNN [12]: Graph neural networks (GNNs) are the main building block of many neural EA approaches. Despite the relevance of MuGNN for pioneering the GNN-based methods in the EA literature, its performance is far from the methods included in our selection of representative methods [82].
- CG-MuAlign [85]: This GNN-based approach was shown to scale to moderately-sized KGs (up to 2.6M entities) and outperform a number of other GNN-based variants and other methods. However, the authors reported performance on a set of KGs (e.g. IMDb, Amazon, etc.) that differs from the KGs prevalent in standardized benchmark datasets used in our study.
- HyperKA [62]: Based on the recent success of hyperbolic representations [47], HyperKA embeds shallow KGs in a hyperbolic space. We ran preliminary experiments with HyperKA and observed that it never outperformed comparable methods (i.e. BootEA and TransEdge).
- Other works such as RNM [86], DUAL-AMN [44], and JEANS [13] were excluded because they are outperformed by BERT-INT.

3.2 Datasets

Scale. One driving factor in the design of EA methods proposed in the pre-neural era was their application to real-world KGs such as Freebase [10] or Yago [60]. This fundamental characteristic is very often overlooked in the design of methods from the neural era, which, in their current state, lack the ability to scale to real-world KGs, and are therefore benchmarked in heavily sub-sampled ($\approx 50x$ smaller) datasets constructed from the original full KGs.

On the one hand, scaling up state of the art neural EA methods to large real-world KGs seems to be a possibility. This is primarily a consequence of the recent advancements in scaling up the training of KG embeddings for performing link prediction—a task that possesses some similarities with the EA task—in large real-world KGs via frameworks such as GraphVite [87], PyTorch-BigGraph [37], and DGL-KE [83]. On the other hand, on account of limited to no evidence of successfully scaling neural EA methods to large KGs, assumptions that straight-forward adaptations of the aforementioned frameworks would suffice are far-fetched. In our humble and honest opinion, non-trivial advancements of the aforementioned frameworks and a substantial amount of engineering efforts (e.g. asynchronous distributed training) and hardware resources (e.g. a large number of GPUs) would be required to enable the neural approaches to run on the original full KGs.

1-to-1 assumption. An important characteristic of *most* of the datasets employed for benchmarking entity alignment methods from the neural era is what we refer to as the **1-to-1 assumption**: *each entity in a KG has a counterpart in the second KG.* The 1-to-1 assumption, also referred to as the closed-domain scenario and discussed independently in a very recent benchmarking study [82], is never observed in real-world KGs. This is especially true when the alignment is performed between pairs of KGs that are fed from different information sources (e.g. the movie-related IMDb and the general KG Wikidata [71]). Another important characteristic is discussed in [66]: the KG properties—e.g. entity degree distribution or KG clustering coefficient—of the sub-sampled datasets generated for benchmarking neural methods are quite different to those of the original KGs. To address this issue, Sun et al. [66] proposed an iterative degree-based sampling (IDS) algorithm to obtain sub-sampled KGs—the OpenEA datasets—that closely approximate the degree distribution of the original KGs while achieving a desired amount of aligned entities. However, we note that the OpenEA datasets are still generated under the *unrealistic* 1-to-1 assumption.

Leakage. By convention, identifiers for ontology terms should be semantics-free or meaningless [3]. Contrary to this general wisdom and convention, we observed that the unique resource identifiers (URIs) of entities for some of the KGs are semantically meaningful. For example, consider the URIs—https://dbpedia.org/page/Barack_Obama, https://yago-knowledge.org/resource/Barack_Obama, and https://www.wikidata.org/wiki/Q76—of the entity Barack Obama in the DBpedia, Yago, and Wikidata KGs, respectively. Specifically,

Table 3: **Datasets benchmarked in this work. While OpenEA datasets were introduced in [61], the remaining datasets have been introduced in this work. All datasets closely approximate the degree distribution of the original KGs.**

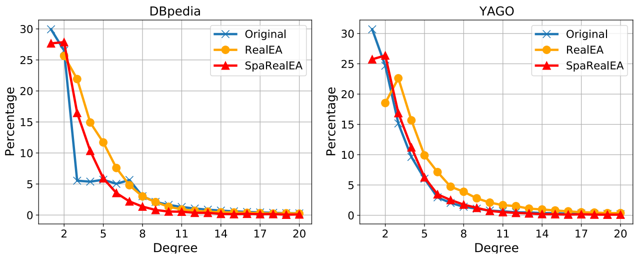
Type	Scope	Main Characteristics
OpenEA	Primary	1-to-1 assumption.
RealEA	Primary	no 1-to-1 assumption.
XRealEA	Primary	no 1-to-1 assumption, cross-lingual.
SupRealEA	Ablation	no 1-to-1 assumption, varying amount of supervision.
AttRealEA	Ablation	no 1-to-1 assumption, varying amount of attributes.
SpaRealEA	Ablation	no 1-to-1 assumption, sparser KG.
RealEA_NoObfs	Ablation	no 1-to-1 assumption, non-obfuscated URIs.
XRealEA_Pure	Ablation	no 1-to-1 assumption, purely cross-lingual.

DBpedia and Yago use a canonicalized—thereby unique—variant of the entity name—thereby meaningful—as a part of the URI, whereas Wikidata uses meaningless identifiers. Meaningful URIs carry information, thereby leading to a problem that we refer to as **leakage**. In fact, a careful analysis of the technical description and the code of RDGCN—a representative neural method for KGs possessing attribute triples—revealed that it leverages this information as if it was a proper literal, which, in hour humble opinion, is an unfair trick. To ensure fairness and consistency across methods, we fix the leakage problem by *obfuscating* entity URIs in all the KGs. Obfuscated URIs are semantics-free or meaningless, thereby preventing any method from unfairly or unintentionally leveraging information that was not meant to be used, as well as conforming to the general wisdom and conventions in the literature.

Towards realistic datasets. An ideal comparison should include KGs as they are. However, as previously argued, neural methods, in their current state, lack the ability to scale to original full KGs containing millions of entities and triples. For this reason and following convention in the literature [66, 82], we proceed to perform the comparison in datasets at a smaller scale. We generate a wide variety of entity alignment datasets constructed from real-world KGs such as Wikidata, DBpedia, and Yago. For a more realistic entity alignment scenario, unless stated otherwise, we impose the following three constraints on the generated datasets:

(1) Similar to [66], the sub-sampled KGs must approximate the degree distribution of the original KGs.
(2) The sub-sampled KG pairs should not follow the unrealistic 1-to-1 assumption.
(3) Entity URIs should be semantics-free and are therefore obfuscated.

Given that the IDS algorithm proposed by Sun et al. [66] only addresses the first requirement, we propose a simple modification to IDS to also address the second requirement. At a high-level, IDS proceeds in two stages. Firstly, it takes as input two KGs and a reference alignment and filters out all the entities that do not have a counterpart entity in the other KG according to the reference alignment. In a second stage, IDS iteratively removes pairs of aligned entities in order to adjust possible discrepancies in the degree distribution between the sampled and original KGs. We refer the reader to [66] for a more detailed technical description. For all practical purposes, our modification—denoted as IDS*—circumvents the first stage of IDS to not enforce the 1-to-1 assumption. The pseudocode for IDS* is presented in the appendix of our technical report [36].



(a) (b)

Figure 1: Comparing the degree distribution of RealEA with SpaRealEA in the sub-sampled KGs generated with IDS* for the (a) DBpedia and (b) Yago datasets, respectively. All the remaining dataset types have the same distribution as that of RealEA, and are therefore omitted in the plot.

We use IDS* to generate a wide variety of dataset types that fulfill the three aforementioned constraints, viz. (1) they closely approximate the degree distributions of the original KGs, (2) do not follow the unrealistic 1-to-1 assumption, and (3) possess obfuscated entity URIs. That said, our primary benchmarking datasets correspond to three broad types: (i) OpenEA, which was generated in [61, 66]; (ii) RealEA, and (iii) XRealEA, which we generate in this work using the IDS* algorithm. Specifically, RealEA and XRealEA can be seen as the realistic and assumption-free counterparts to the mono-lingual and cross-lingual OpenEA datasets, respectively. Moreover, our analysis goes much further and also explores other characteristics that might affect the comparison between neural and non-neural EA methods. These characteristics are investigated with variants of RealEA and XRealEA, which serve the purpose of ablation studies. Primarily, the variants are designed to study the *robustness* of the methods to, among others, the amount of supervision (SupRealEA) or attributes (AttRealEA). Table 3 provides a summary of all the dataset types benchmarked in this work.

Fig. 1 illustrates that the datasets generated by IDS* closely approximate the degree distribution of the original KGs. While RealEA better mimics the original KGs in the high-degree (i.e., > 8) range, SpaRealEA does a better reconstruction in the low-range, thereby becoming relatively more sparse—hence its name. The dataset statistics are provided in Table 4. The following shorthand codes are used: DBpedia (DB), Yago (YG), Wikidata (WD), English (EN), French (FR), Japanese (JA)—DBpedia is the underlying KG of the last three datasets. For all datasets except SupRealEA, the seed alignment set consists of 20% of overall matchable entities, which was also found to conform to the real world [66].

3.3 Evaluation

Historically, EA strategies [35, 59] have been evaluated using information classification-based evaluation metrics such as precision, recall, and F_1-score [28]. They are computed by comparing the set of alignments output by a system to the ground truth. The system aims to retrieve as many correct alignments as possible while keeping the number of incorrect alignments as low as possible.

On the other hand, neural EA approaches adopted ranking-based evaluation metrics such as mean reciprocal rank or precision at k [43]. While it is not clear why these metrics were chosen, one may intuit that this was motivated by the recent literature on neural approaches for link prediction, which have largely influenced many neural EA methods. Link prediction methods are evaluated based

Table 4: **Dataset statistics. Every dataset type contains pairs of KGs to be aligned. The dataset types SupRealEA and AttRealEA modify RealEA, while XRealEA_Pure modifies XRealEA for performing ablation studies. Therefore, their statistics only vary for #Attributes or #Att. Triples, and are presented in the appendix in our technical report [36].**

	Dataset Type											
	OpenEA				RealEA				XRealEA			SpaRealEA
Dataset	DB-YG-15K	DB-WD-15K	DB-YG-100K	DB-WD-100K	DB-YG-15K	DB-WD-15K	DB-YG-100K	DB-WD-100K	EN-FR-15K	EN-DE-15K	EN-JA-15K	DB-YG-15K
#Entities	15,000 - 15,000	15,000 - 15,000	100,000 - 100,000	100,000 - 100,000	19,865 - 21,050	20,038 - 19,581	126,145 - 136,211	129,847 - 137,721	20,473 - 19,922	19,680 - 19,740	21,537 - 19,751	20,399 - 21,019
#Relations	165 - 28	248 - 169	287 - 32	413 - 261	290 - 32	306 - 214	386 - 32	456 - 329	303 - 254	287 - 155	341 - 219	291 - 32
#Attributes	257 - 35	342 - 649	379 - 38	493 - 874	247 - 34	307 - 490	366 - 38	478 - 785	303 - 451	366 - 217	335 - 186	241 - 34
#Rel. Triples	30,291 - 26,638	38,265 - 42,746	294,188 - 400,518	293,990 - 251,708	60,329 - 82,109	50,007 - 65,017	479,510 - 653,261	399,061 - 489,698	57,224 - 54,366	46,432 - 35,808	87,662 - 65,769	36,282 - 46,241
#Att. Triples	71,716 - 132,114	68,258 - 138,246	523,062 - 749,787	451,011 - 687,860	129,330 - 392,845	85,331 - 112,786	677,721 - 1,427,545	566,073 - 668,925	109,141 - 85,283	111,557-108,760	99,812 - 68,389	123,822 - 326,896
#Matchable Ent.	15,000	15,000	100,000	100,000	15,000	15,000	100,000	100,000	15,000	15,000	15,000	15,000

on their ability to complete queries. A query (*head*, *relation*, ?) is an entity-relation pair where a second entity is missing to form a valid triple. This unrealistic evaluation requires a partial knowledge of the ground truth, and has been very recently shown not to translate to good performance on the actual KG completion task [58].

A similar concern might be raised when ranking-based metrics are used for EA. The counterpart query in this problem is (*head*, *sameAs*, ?), which assumes partial knowledge of the ground truth: the set of *matchable* entities. In practice, this knowledge is reflected in the 1-to-1 assumption—every entity in one KG has a counterpart in the other—that serves as the basis for the datasets used by the neural EA community. It is up for discussion whether the choice of metrics influenced the dataset construction process, or vice versa.

Towards a realistic evaluation. As previously argued, in a realistic setting, we will not possess the apriori knowledge of which entities in a KG are matchable (i.e. have a counterpart entity in the other KG). For this reason, we proceed with a homogeneous evaluation where each EA method is validated and evaluated based on standard classification-based metrics, i.e., precision, recall, and F_1-score [28]. Let $\mathcal{M}$ be the set of entity pairs outputted by a system, the aforementioned metrics are formalized as follows. Note that F_1 conveys a balance between precision and recall, and is usually used as an indicator of the overall system performance [28, 43].

$$\text{Prec.} = \frac{|\mathcal{M} \cap \mathcal{G}|}{|\mathcal{M}|} \quad \text{Recall} = \frac{|\mathcal{M} \cap \mathcal{G}|}{|\mathcal{G}|} \quad F_1 = 2\frac{Prec. * Recall}{Prec. + Recall}$$

The set $\mathcal{M}$ is outputted by the matching module. For a fair comparison across methods, whose main contributions are independent to the matching module, we deliberately homogenize this module for all benchmarked methods. The input to this module is a weighted bipartite graph where every node in one graph is connected to every node in the other graph, and the output is a pruned version of the same graph, where every node may keep at most one single edge. This pruning procedure is known as the assignment problem, for which there exists a number of solutions that unfortunately scale quadratically or cubically with the number of nodes [25]. Given the scale of the KGs, we explore simpler greedy solutions. We empirically find that the matching strategy implemented in Paris works the best: two entities (e, e') are matched if $e' \leftarrow \arg\max_{x \in \mathcal{KG}_2} f(e \equiv x)$ and $e \leftarrow \arg\max_{x \in \mathcal{KG}_1} f(e' \equiv x)$, where f indicates either a probability or a similarity metric. The pseudocode is presented in the appendix of our technical report [36].

4 RESULTS: ENTITY ALIGNMENT

In this section, we assess the efficacy and efficiency of five representative methods—Paris+ from the pre-neural era, and BootEA, TransEdge, RDGCN and BERT-INT from the neural-era—for performing entity alignment. While BootEA and TransEdge use only the signals manifested in the relationships between entities (*shallow KG*), Paris+, RDGCN, and BERT-INT leverage the information manifested in both attributes and relation triples. We report the average and standard deviation of the results obtained via 5-fold cross-validation. For additional details about the experiment setup or results, please see the appendix of our technical report [36].

4.1 OpenEA Datasets

Table 5a presents the first set of results, which are based on the OpenEA datasets provided by Sun et al. [66]. Before discussing the results, it is important to highlight the following noteworthy traits of the experiment: (1) these datasets are generated under the unrealistic 1-to-1 assumption (§ 3.2), (2) neural methods leverage the said 1-to-1 assumption and use the entity matching module prescribed by the OpenEA library [61], thereby being at an advantageous position when compared to Paris+, (3) Paris+ operates oblivious to the 1-to-1 assumption and uses the general bidirectional entity matching algorithm (cf. appendix in [36]), (4) neural methods use CSLS [17, 66] as it consistently improves their efficacy, and (5) owing to multiple flaws (details in the appendix of our technical report [36]) in the implementation of Paris exposed by the OpenEA library [61], the reported results for Paris+ in Table 5a are substantially different (and better) from those reported in [66].

It is evident from Table 5a that Paris+ significantly outperforms all the neural methods on each evaluation metric across all datasets, with improvements ranging from 3%–7% and 40%–50% (absolute difference) on the DB-YG and DB-WD datasets, respectively. Additionally, the stability of all the methods to varying testing portions of the datasets is established by the consistently low standard deviations observed for each method across dataset types and metrics.

Note that the observed performance for all the methods is generally lower on the DB-WD datasets than the DB-YG datasets. This is explained in [66] as a consequence of the existence of symbolic heterogeneity of attributes in Wikidata: Attributes are encoded using numeric identifiers. While the attribute heterogeneity adversely affects attribute embedding based neural methods (RDGCN), the consistently strong performance of Paris+ in this scenario is noteworthy. Overall, this result establishes the strength of non-neural (Paris+) EA methods and raises concerns around the omission of such a comparison from the literature on neural EA.

4.2 RealEA Datasets

The next set of results are based on the RealEA datasets, which are generated using the IDS* algorithm (cf. appendix in [36]) proposed

Table 5: Entity alignment quality measured using precision, recall, and F_1-score on the (a) OpenEA, (b) RealEA, (c) XRealEA, and (d) AttRealEA datasets. For each method, we perform a 5-fold cross-validation and report the mean and standard deviation. The best performance (higher numbers indicate better performance) is shown in bold.

(a) OpenEA Datasets

		DB-YG-15K (OpenEA)			DB-WD-15K (OpenEA)			DB-YG-100K (OpenEA)			DB-WD-100K (OpenEA)		
Category	**Method**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**
Neural	BootEA	0.926 ± 0.002	0.675 ± 0.004	0.781 ± 0.002	0.806 ± 0.003	0.547 ± 0.005	0.651 ± 0.003	0.898 ± 0.002	0.625 ± 0.001	0.737 ± 0.001	0.763 ± 0.003	0.452 ± 0.004	0.568 ± 0.003
Neural	RDGCN	0.984 ± 0.001	0.855 ± 0.003	0.915 ± 0.002	0.799 ± 0.004	0.384 ± 0.005	0.519 ± 0.005	0.972 ± 0.008	0.815 ± 0.060	0.886 ± 0.038	0.713 ± 0.005	0.261 ± 0.005	0.382 ± 0.001
Neural	BERT-INT	0.875 ± 0.001	**0.969 ± 0.002**	0.920 ± 0.001	0.834 ± 0.009	0.197 ± 0.008	0.319 ± 0.011	0.874 ± 0.000	**0.965 ± 0.001**	0.918 ± 0.000	0.819 ± 0.003	0.128 ± 0.004	0.221 ± 0.006
Neural	TransEdge	0.367 ± 0.085	0.212 ± 0.056	0.268 ± 0.068	0.743 ± 0.014	0.453 ± 0.018	0.562 ± 0.018	0.730 ± 0.017	0.481 ± 0.011	0.579 ± 0.006	0.687 ± 0.051	0.436 ± 0.027	0.533 ± 0.035
Non-neural	Paris+	**0.998 ± 0.000** †	0.961 ± 0.001	**0.979 ± 0.001**	**0.970 ± 0.001**†	**0.743 ± 0.005**†	**0.842 ± 0.003**†	**0.998 ± 0.000**†	0.957 ± 0.001	**0.977 ± 0.000**†	**0.963 ± 0.001**†	**0.709 ± 0.002**†	**0.817 ± 0.001**†

(b) RealEA Datasets

		DB-YG-15K (RealEA)			DB-WD-15K (RealEA)			DB-YG-100K (RealEA)			DB-WD-100K (RealEA)		
Category	**Method**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**
Neural	BootEA	0.459 ± 0.008	0.313 ± 0.009	0.372 ± 0.007	0.609 ± 0.007	0.280 ± 0.009	0.383 ± 0.008	0.671 ± 0.005	0.487 ± 0.004	0.565 ± 0.003	0.548 ± 0.008	0.272 ± 0.007	0.363 ± 0.006
Neural	RDGCN	0.822 ± 0.003	0.709 ± 0.004	0.761 ± 0.003	0.583 ± 0.012	0.242 ± 0.009	0.342 ± 0.011	0.846 ± 0.001	0.708 ± 0.002	0.771 ± 0.001	0.538 ± 0.003	0.203 ± 0.001	0.295 ± 0.001
Neural	BERT-INT	0.817 ± 0.001	0.827 ± 0.004	0.822 ± 0.002	0.604 ± 0.030	0.075 ± 0.006	0.134 ± 0.010	0.841 ± 0.001	0.865 ± 0.006	0.853 ± 0.003	0.698 ± 0.009	0.120 ± 0.002	0.206 ± 0.003
Neural	TransEdge	0.335 ± 0.025	0.203 ± 0.017	0.253 ± 0.020	0.589 ± 0.126	0.183 ± 0.034	0.279 ± 0.054	0.566 ± 0.011	0.438 ± 0.018	0.494 ± 0.016	0.339 ± 0.041	0.147 ± 0.012	0.205 ± 0.018
Non-neural	Paris+	**0.906 ± 0.000** †	**0.931 ± 0.001**†	**0.918 ± 0.001**	**0.928 ± 0.002**†	**0.551 ± 0.004**†	**0.691 ± 0.003**†	**0.923 ± 0.000**†	**0.939 ± 0.000**†	**0.931 ± 0.000**†	**0.927 ± 0.001**†	**0.615 ± 0.001**†	**0.740 ± 0.001**†

(c) XRealEA Datasets

		EN-FR-15K (XRealEA)			EN-DE-15K (XRealEA)			EN-JA-15K (XRealEA)			EN-JA-15K (XRealEA_Pure)		
Category	**Method**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**
Neural	BootEA	0.528 ± 0.013	0.312 ± 0.007	0.392 ± 0.005	0.556 ± 0.011	0.222 ± 0.018	0.317 ± 0.017	0.448 ± 0.002	0.294 ± 0.006	0.355 ± 0.005	0.445 ± 0.009	0.291 ± 0.027	0.351 ± 0.022
Neural	RDGCN	0.755 ± 0.004	0.532 ± 0.004	0.624 ± 0.004	0.736 ± 0.003	0.484 ± 0.002	0.584 ± 0.002	0.461 ± 0.002	0.161 ± 0.001	0.238 ± 0.001	0.212 ± 0.009	0.037 ± 0.002	0.063 ± 0.004
Neural	BERT-INT (desc)	0.836 ± 0.003	**0.970 ± 0.002** †	**0.898 ± 0.002** †	0.842 ± 0.004	**0.977 ± 0.003** †	**0.905 ± 0.003** †	**0.835 ± 0.003**	**0.960 ± 0.006** †	**0.893 ± 0.003** †	**0.835 ± 0.004** †	**0.958 ± 0.004** †	**0.892 ± 0.002** †
Neural	BERT-INT (no desc)	0.806 ± 0.002	0.636 ± 0.003	0.711 ± 0.002	0.800 ± 0.002	0.558 ± 0.007	0.658 ± 0.005	0.765 ± 0.022	0.225 ± 0.020	0.347 ± 0.025	0.125 ± 0.250	0.000 ± 0.001	0.000 ± 0.001
Neural	TransEdge	0.479 ± 0.013	0.219 ± 0.030	0.299 ± 0.028	0.478 ± 0.023	0.197 ± 0.013	0.278 ± 0.010	0.384 ± 0.024	0.174 ± 0.016	0.239 ± 0.019	0.386 ± 0.026	0.176 ± 0.023	0.242 ± 0.027
Non-neural	Paris+	**0.902 ± 0.001** †	0.800 ± 0.002	0.848 ± 0.001	**0.910 ± 0.001** †	0.795 ± 0.003	0.849 ± 0.002	0.827 ± 0.002	0.624 ± 0.005	0.712 ± 0.004	0.704 ± 0.007	0.309 ± 0.006	0.430 ± 0.007

(d) AttRealEA Datasets

		DB-YG-15K (AttRealEA_All)			DB-WD-15K (AttRealEA_All)			DB-YG-15K (AttRealEA_None)			DB-WD-15K (AttRealEA_None)		
Category	**Method**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**	**Precision**	**Recall**	**F_1-score**
Neural	BootEA	0.471 ± 0.006	0.314 ± 0.005	0.377 ± 0.003	0.603 ± 0.003	0.273 ± 0.013	0.376 ± 0.012	0.455 ± 0.011	0.311 ± 0.006	0.370 ± 0.007	0.607 ± 0.015	0.274 ± 0.010	0.378 ± 0.009
Neural	RDGCN	0.824 ± 0.003	0.713 ± 0.003	0.764 ± 0.003	0.871 ± 0.002	0.757 ± 0.004	0.810 ± 0.003	**1.000 ± 0.000**	0.000 ± 0.000	0.000 ± 0.000	**1.000 ± 0.000**	0.000 ± 0.000	0.000 ± 0.000
Neural	BERT-INT	0.818 ± 0.004	0.827 ± 0.003	0.822 ± 0.002	0.837 ± 0.003	0.837 ± 0.008	0.837 ± 0.005	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000	0.000 ± 0.000
Neural	TransEdge	0.510 ± 0.021	0.199 ± 0.021	0.286 ± 0.025	0.583 ± 0.121	0.176 ± 0.046	0.271 ± 0.067	0.373 ± 0.009	0.236 ± 0.012	0.289 ± 0.011	0.631 ± 0.056	0.192 ± 0.020	0.294 ± 0.030
Non-neural	Paris+	**0.893 ± 0.001** †	**0.924 ± 0.001**†	**0.908 ± 0.000**	**0.970 ± 0.000**†	**0.909 ± 0.002**†	**0.939 ± 0.001**†	0.736 ± 0.004	**0.352 ± 0.002**†	**0.476 ± 0.002**†	0.858 ± 0.010	**0.322 ± 0.003**†	**0.468 ± 0.002**†

† Indicates statistical significance ($p < 0.01$) between the best and the second-best method using the Student's paired t-test.

in this work. This experiment is different from the one discussed in § 4.1 in the following ways: (1) the generated datasets do not follow the unrealistic 1-to-1 assumption, (2) consequently, all the methods—both neural and non-neural—use the general bidirectional entity matching algorithm (cf. appendix in [36]), and (3) the CSLS trick does not consistently improve the efficacy of the neural methods, and thus, the reported results correspond to the setting (with or without CSLS) that achieves the best F_1-score.

Table 5b shows that Paris+ significantly outperforms all the neural methods on each evaluation metric and across all datasets. More specifically, Paris+ obtains improvements ranging 2%–5% and 60%–80% on the DB-YG and DB-WD datasets, respectively.

This result further substantiates the strength of non-neural methods over their neural counterparts, as Paris+ outperforms all the methods not only on the OpenEA datasets, which are generated based on the unrealistic 1-to-1 assumption, but also on the RealEA datasets that alleviate the aforementioned unrealistic assumption.

4.3 XRealEA Datasets

We also perform an experiment using the XRealEA datasets to assess the efficacy of the considered methods for cross-lingual entity alignment. We follow the exact same setup as described in § 4.2. It was claimed by Sun et al. [66] that non-English KGs were required to be translated to English to mitigate the language barrier for Paris. However, our experiments revealed its futility, as Paris+ was able to successfully operate on the XRealEA datasets without requiring such a preprocessing step. That said, we use the original KGs without performing any translations of the non-English literals.

Table 5c shows that yet again Paris+ significantly outperforms all (but one) neural methods (TransEdge, BootEA, and RDGCN) on each evaluation metric across all datasets, with improvements ranging from 15%–45%. Specifically, Paris+ is second only to BERT-INT. However, it should be noted that only BERT-INT (with descriptions) is better than Paris+, whereas BERT-INT (without descriptions) is still outperformed by Paris+. Note that DBpedia article descriptions constitute additional information, which is only leveraged by BERT-INT, thereby making this comparison slightly unfair.

Importantly, this result eradicates the misconception about Paris—i.e., Paris lacks the ability to perform EA in multi-lingual KGs without data preprocessing—plaguing the existing EA literature.

4.4 Analysis: RealEA Datasets

To better understand the efficacy of all the considered methods in different evaluation scenarios, we analyze the effect of three important dataset characteristics, viz., the number of attributes, the number of relations, and the amount of supervision. The analyses presented in this section are performed on the 15K datasets. Moreover, we exclude the OpenEA datasets from this analysis because:

(1) RealEA datasets closely mimic (§ 3.2) the real-world entity alignment scenarios, and are therefore, more relevant, and (2) similar outcomes were observed in the analysis conducted on the OpenEA datasets, and hence, their results are omitted for the sake of brevity.

AttRealEA: Robustness to the number of attributes. While RealEA datasets use the exact same attributes as prescribed by Sun et al. [66], the original full-sized KGs possess many more attributes. Here, we construct two additional variants of the RealEA datasets: (1) AttRealEA_All: possessing all the attributes present in the original KG, and (2) AttRealEA_None: depicting an absence of attributes (a.k.a. *shallow KG*). Next, we analyze the impact of the number of attributes on the entity alignment performance of the considered methods. The results are presented in Table 5d. Since TransEdge and BootEA do not rely on attribute information, their performance remains stable across variants. On the contrary, RDGCN, BERT-INT, and Paris+ leverage both relation and attribute triples for entity alignment, thus, their performance improves considerably with the addition of attributes (AttRealEA_All) whereas deteriorates substantially with their removal (AttRealEA_None). Moreover, the performance of RDGCN and BERT-INT deteriorates more substantially (F_1-score of 0) than Paris+ for the AttRealEA_None variants, thereby showcasing the robustness of Paris+ to variations in the number of attributes.

Importantly, this analysis rectifies the incorrect claim in [66] about the inability of Paris to obtain any entity alignments using relation triples alone: Our simple variant Paris+ works even in the absence of attribute triples (AttRealEA_None). In fact, even for AttRealEA_None datasets Paris+ obtains a statistically significant improvement of around 25% over TransEdge and BootEA.

SpaRealEA: Robustness to the sparsity in the graph. Here, we analyze the impact of the number of relations on the entity alignment performance of the considered methods using the SpaRealEA dataset, which, as indicated in Table 4, possesses half the number of relation triples when compared to the RealEA dataset. The results are presented in the appendix of our technical report [36]. For RDGCN, BERT-INT, and Paris+, the obtained results are similar to that observed on RealEA datasets, thereby showcasing their robustness to the variation in the number of relations in the KG. On the contrary, the performance of TransEdge and BootEA deteriorates significantly when compared to RealEA (Table 5b) as it relies only on relation triples.

SupRealEA: Robustness to the amount of supervision. Lastly, we vary the amount of supervision from the set {1%, 5%, 10%, 20%, 30%}, and analyze its impact on the performance of the considered methods using the SupRealEA dataset. The results are presented in the appendix of our technical report [36]. We find that RDGCN, BERT-INT, and Paris+ are robust to the amount of supervision obtaining consistently strong performance even with a negligible number of seed alignments. On the contrary, the amount of supervision strongly affects TransEdge and BootEA, and it performs very poorly when the number of seed alignments are small.

Summarizing the outcomes from the aforementioned analyses, it is evident that Paris+ is robust to variations in different dataset characteristics, and consistently and significantly outperforms all the neural methods across dataset types and variants.

5 RESULTS: RECORD LINKAGE

As briefly discussed in § 2, RL and EA are counterpart problems. While the former addresses the task of finding identical records between relational DBs, the latter aims to finds identical entities between KGs. We note that tables in relational DBs play a role similar to the relationships in KGs, and thus, it is possible to represent KGs as a set of records distributed across a number of different tables. However, we also note that relational DBs are not specifically designed for this use case. Instead, graph DBs may store KGs in a more natural manner. Interestingly, it is still possible to represent the information encoded in KGs as tables in a relational DB by serializing entities as DB-style records, which captures the 1-hop neighborhood of the entity. This procedure enables standard RL methods to operate on tables that are meant to approximate KGs.

Specifically, we group KG attributes into aspects, namely—(1) names, (2) other attributes, and (3) relationships. We also capture the 1-hop neighborhood of each entity by extending the aspects with 1-hop names and other attributes. Having obtained the serialized DB-style record representation of KG entities, we leverage two state-of-the-art RL methods, namely—(1) DeepMatcher [46], and (2) Ditto [39] for performing EA. To obtain manageable and high-quality datasets for training RL methods, we perform meta-blocking (cf. the appendix in [36] for additional details about blocking) using the TF-IDF weighting scheme [50, 51]. Both DeepMatcher and Ditto are trained for 10 epochs using the recommended hyperparameters described in the respective papers [39, 46].

From an empirical standpoint, we extensively evaluate DeepMatcher and Ditto on RealEA and XRealEA (cf. the appendix in [36] for XRealEA results) datasets. It is evident from Table 6 that both DeepMatcher and Ditto portray competitive performance on the EA task. While Paris+ remains the best performing method, Ditto significantly outperforms both BERT-INT and BootEA, the best performing neural EA methods with and without attributes, respectively, on majority of the datasets. These results unravel the strength of RL methods to effectively address EA.

6 RESULTS: EFFICIENCY AND SCALABILITY

Moving beyond efficacy, we also evaluate the efficiency (measured using running time) and scalability (measured using memory footprint) of all the benchmarked methods across all dataset types. The results are presented in the appendix of our technical report [36].

The key finding is that Paris+ is orders of magnitude faster than all the neural methods. While Paris+ is around 100 and 1000 times faster for training and inference, respectively, on the 15K datasets, its performance improves further—1000 times faster training and 10000 times faster inference—on the 100K datasets. Moreover, even the memory footprint of Paris+ is 5–10 times smaller than the neural methods. Neural RL methods are second only to Paris+, and portray 3–4 times faster training, albeit a moderately slower inference performance, and 2–3 times smaller memory footprint when compared to all the neural EA methods. These results indicate that Paris+, DeepMatcher, and Ditto possess the ability to scale gracefully to large datasets, while neural EA methods do not.

The scalability aspect is further substantiated by performing experiments on the 500K datasets, and the original full-sized KGs: DBpedia (5.9M entities and 18.7M relations), Yago (4.3M entities

Table 6: A comparison of the best entity alignment (EA) methods (BootEA, BERT-INT, and Paris+) with state-of-the-art record linkage (RL) methods (DeepMatcher, and Ditto) using precision, recall, and F_1-score on the RealEA datasets.

		DB-YG-15K			DB-WD-15K			DB-YG-100K			DB-WD-100K		
Category	**Method**	**Prec.**	**Rec.**	F_1	**Prec.**	**Rec.**	F_1	**Prec.**	**Rec.**	F_1	**Prec.**	**Rec.**	F_1
RL	DMatch	0.85 ± 0.02	0.79 ± 0.01	0.82 ± 0.01	0.23 ± 0.01	0.16 ± 0.01	0.19 ± 0.01	0.88 ± 0.01	0.69 ± 0.01	0.77 ± 0.01	0.05 ± 0.02	0.34 ± 0.00	0.09 ± 0.01
RL	Ditto	0.87 ± 0.01	0.82 ± 0.01	0.84 ± 0.00	0.34 ± 0.02	0.21 ± 0.00	0.26 ± 0.00	**0.92 ± 0.01**	0.68 ± 0.00	0.78 ± 0.01	0.76 ± 0.01	0.25 ± 0.01	0.38 ± 0.01
EA	BootEA	0.46 ± 0.01	0.31 ± 0.01	0.37 ± 0.01	0.61 ± 0.01	0.28 ± 0.01	0.38 ± 0.01	0.67 ± 0.01	0.49 ± 0.00	0.57 ± 0.00	0.55 ± 0.01	0.27 ± 0.01	0.36 ± 0.01
EA	BERT-INT	0.82 ± 0.00	0.83 ± 0.00	0.82 ± 0.00	0.60 ± 0.03	0.08 ± 0.00	0.13 ± 0.01	0.84 ± 0.00	0.87 ± 0.01	0.85 ± 0.00	0.70 ± 0.01	0.12 ± 0.00	0.21 ± 0.00
EA	Paris+	**0.91 ± 0.00**†	**0.93 ± 0.00**†	**0.92 ± 0.00**	**0.93 ± 0.00**†	**0.55 ± 0.00**†	**0.69 ± 0.00**†	**0.92 ± 0.00**	**0.94 ± 0.00**†	**0.93 ± 0.00**†	**0.93 ± 0.00**†	**0.62 ± 0.00**†	**0.74 ± 0.00**†

† Indicates statistical significance ($p < 0.01$) between Paris+ and the second-best method using the Student's paired t-test.

and 12.4M relations), and Wikidata (19.6M entities and 64.5M relations). For the 500K datasets, all the methods apart from Paris+ and Ditto crashed owing to going out of memory. Note that DeepMatcher did not go out of GPU memory, but required more than 256 GB RAM. Paris+ obtained an F_1-score of 0.630 requiring 610 seconds for training and 0.94 seconds for inference on DB-WD-500K, and an F_1-score of 0.90 requiring 1320 seconds for training and 1.3 seconds for inference on DB-YG-500K. Ditto obtained an F_1-score of 0.31 requiring 12 hours for training and 5 hours for inference on DB-WD-500K, and an F_1-score of 0.71 requiring 19 hours for training and 8 hours for inference on DB-YG-500K. Moving further, Paris+ was the only technique to successfully obtain entity alignments on the full-sized DB-YG (training: 6 hours, inference: 7 seconds, F_1-score: 0.773) and DB-WD (training: 12 hours, inference: 10 seconds, F_1-score: 0.722) KG pairs.

7 DISCUSSION AND CONCLUDING INSIGHTS

The conclusions emanating from the in-depth analysis provided in this work are clear. With only one exception, Paris+ achieves the best performance across a wide-variety of datasets and evaluation metrics. Specifically, Paris+ is second only to BERT-INT for a particular scenario of cross-lingual EA. Moreover, Paris+ is several orders of magnitude faster than the best-performing neural methods, making it a "true" solution to the eventual task, viz., performing EA in large-scale KGs with millions of entities and relations. Finally, Paris+ does not require any parameter fine tuning, thereby being an extremely easy-to-use off-the-shelf tool with only a single requirement, that the ontology/KG be provided in a certain standard format. For all the aforementioned reasons, *we recommend that future work should always include Paris+ as a baseline.*

Moving beyond EA methods, we also highlight the strong and competitive performance obtained via RL methods for addressing EA. Similar to Paris+, Ditto should also be included as a baseline for future work on neural entity alignment.

That said, we do not want our findings to be perceived as discouraging judgments directed towards neural EA methods. Conversely, we think that neural methods have much more to offer and need to be positioned better (in terms of use cases or dataset types) to portray their true potential. Moreover, the goal of this work is not to serve as an impediment to the vibrant field of neural EA, rather facilitate its streamlined and meaningful advancement. To this end, the remainder of this section includes a comprehensive discussion providing substantive answers to the following three questions:

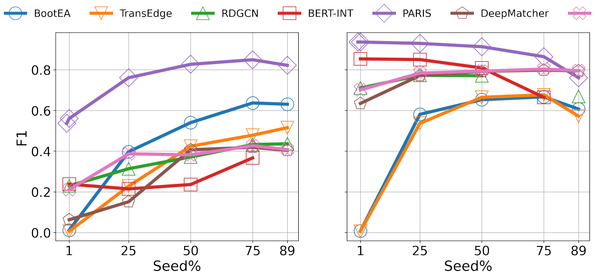
- Is the amount of supervision in our standard evaluation setup sufficient for data-intensive neural methods? (§ 7.1)
- Why does Paris+ outperform all the neural methods? (§ 7.2)
- What lies in the future for the field of neural EA? (§ 7.3)

7.1 Amount of Supervision

The goal of this section is to assess whether the main conclusion—the superiority of Paris+ over all the considered methods—drawn from the previously conducted experiments is a by-product of an experimental design choice that may be negatively biased towards neural approaches. The amount of training data for (almost) all the experiments corresponds to 20% of all existing alignments. This is justified because: i) it is the standard setup choice used by most (if not all) of the EA methods from the neural literature; and ii) it conforms to some of the real-word datasets [66]. Nevertheless, neural approaches, which are known to be data demanding, might be negatively affected by this setup. To this end, we vary the amount of supervision from the set $\{1\%, 25\%, 50\%, 75\%, 89\%\}$, and analyze its impact on the performance of the considered methods using the 100K versions of the RealEA datasets. Note that this analysis is complementary to that performed on SupRealEA, where we considered the lower end of the supervision spectrum as the focus was to assess the robustness to the amount of supervision.

Fig. 2 presents the results. Note that RDGCN and BERT-INT crashed for 75% and 89% supervision, respectively, and thus, their results are omitted. On the one hand, the performance of TransEdge and BootEA, which do not leverage attribute triples, improves substantially up to 75% of training data. On the other hand, for almost all other methods that leverage attribute information, the increase is considerable until 25%, whereas much less pronounced in the range [25%, 75%]. We note that neural methods are (partially) built upon heavy language models that are pre-trained on very large corpora. We argue that this characteristic, which reduces to a large extent their dependency on a supervised signal, may explain the early *plateau* observed for these methods. This argument is bolstered in the comparison to TransEdge and BootEA, where all the model weights are solely learned from the supervised seed alignments. Moving ahead, and with almost no exception, the improvement from 75% to 89% training data is little or none for all the methods. Interestingly, both BERT-INT and Paris+ suffer a consistent decrease in performance on DB-YG-100K with increasing amount of supervision, however, further investigation on this aspect is left for future work. Lastly, it is important to note that Paris+ significantly outperforms all methods across the full supervision spectrum, thereby, reinforcing its strength over all competing methods.

To conclude, we argue that the ensemble of the aforementioned observations indicate that the amount of supervision used in our standard setup is not the fundamental factor behind the differences in performance across methods, thereby conforming the generalizability of the conclusions drawn from the results obtained in § 4 and § 5 using the standard evaluation setup with 20% supervision.



(a) DB-WD-100K (b) DB-YG-100K

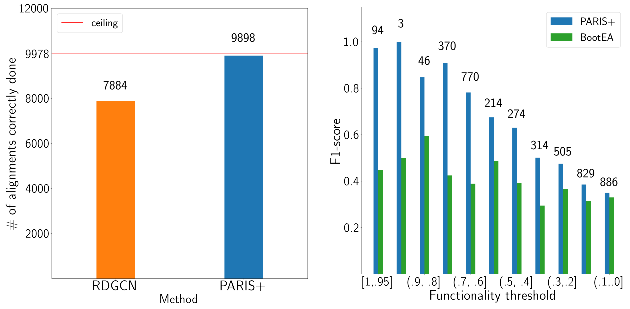
Figure 2: EA quality measured using F_1-score with varying amount of supervision on 100K RealEA datasets.

7.2 Demystifying Paris+

To unveil the characteristics that serve as the key differentiating factors between Paris+ and its closest competitors from the neural era, we perform an in-depth analysis of their outputs in the following two exemplary settings: (1) KGs possessing attributes triples, and (2) KGs lacking attribute triples.

KGs possessing attributes triples. For this analysis, we use DB-WD-15K (AttRealEA_All) and analyze the differences in the performance of methods that are designed for KGs with attributes, i.e., Paris+ and RDGCN. Note that BERT-INT was excluded for two reasons: (i) it was harder to provision a GPU with 32GB memory only for the purpose of this analysis, (ii) the difference between the performance of RDGCN and BERT-INT is not large enough to effect the conclusions drawn from this analysis. Results on other datasets show similar trends, and are therefore omitted. Since all the attribute triples are retained in the AttRealEA_All dataset variant, it facilitates assessing the impact of attributes on the performance of the considered EA methods to the fullest extent. Moreover, we focus only on those counterpart entities (denoted as Easy) that can be correctly aligned by simply performing an exact string matching on the literals corresponding to at least one attribute of the counterpart entities. The reason is twofold: such counterpart entities (i) constitute the easiest scenario for EA, thereby making any downstream analysis more interpretable, and (ii) constitute almost 80% of the entire test set, thereby possessing the ability to explain the bulk of the performance obtained by any EA method.

Fig. 3a shows that 9,978 entities—out of the 12K matchable entities in the test set—can be aligned using simple string matching, which is represented as the 'ceiling' performance obtainable by any EA method. It is important to note that Paris+ aligns almost all (9,898) the entities from this set, achieving an F_1-score of 0.99. However, RDGCN only aligns 7,884 with an F_1-score of 0.85. A noteworthy finding from this analysis is that ≈ 80% of the overall difference in the performance of Paris+ and RDGCN can be explained by the performance difference in the set of Easy alignments. This finding is somewhat counter-intuitive, as neural methods use embeddings, a more sophisticated and stronger paradigm for similarity computation between literals that can even handle cross-lingual scenarios gracefully, whereas Paris+ simply relies on exact (not even fuzzy) string matching. A careful analysis of the neural methods and their code revealed that while they possess a powerful similarity computation module they just focus on a single attribute, which possesses the least amount of missing values. This is important as only very recently, GNNs [78] started handling missing values on homogeneous graphs, however, to our knowledge, there does not exist any such work for heterogeneous graphs such as KGs. Conversely, Paris+ exploits all the attributes associated with entities, which serves as the key reason for its superior mileage.



(a) (b)

Figure 3: Comparing the (a) number of correct alignments of Paris+ with RDGCN on Easy portion of DB-WD-15K (AttRealEA_All), and (b) variation in F_1-score of Paris+ and BootEA with relationships of varying functionality for EN-JA-15K (XRealEA_Pure). For both plots, the absolute number of entity alignments are indicated above the bars.

Thus, it is safe to say that Paris+ leverages the rich information manifested in the attributes in the best possible manner, however, neural methods lack on this front. To mitigate this gap, neural methods should focus on identifying ways to leverage all the attributes instead of just one. More on this in Sec. 7.3.

KGs lacking attribute triples. For this analysis, we use EN-JA-15K (XRealEA_Pure) and analyze the differences in the performance of Paris+ and the best performing method for *shallow KG*, i.e., BootEA. We use XRealEA_Pure since it serves as the most challenging (Sec. 4.3) dataset variant for methods that leverage attribute information. In fact, since Paris+ relies on exact string matching it cannot leverage attribute information in this setting, thereby forcing it to rely solely on the signal manifested in the relationship triples and leading to a fair comparison with BootEA, which leverages only relationship triples by design. Based on the results presented in Sec. 4, it is evident that Paris+ statistically significantly outperforms all neural methods in KGs lacking attribute triples. This is a noteworthy finding as those neural methods (BootEA and TransEdge) were designed specifically for *shallow KG*. Our analysis of the technical description of Paris+ and BootEA (or for that matter TransEdge) revealed a key difference between their design principles concerning extracting information manifested in the relationship triples. As explained in Sec. 2.1, Paris+ exploits the concept of *functionality* (Eq. 1) to determine the importance of a relationship—the higher the functionality the more important the relationship—while determining alignment between counterpart entities (Eq. 2). While highly relevant, to the best of our knowledge, such a signal is not exploited by any neural method. That said, we begin this analysis with the hypothesis that functionality serves as a key differentiating factor in the performance of Paris+ and competing neural methods. Fig. 3b strongly corroborates this hypothesis. The difference in F_1-score of Paris+ and BootEA for relationships with higher functionality (≥ 0.6) is noteworthy, while for lower functionality (< 0.3) their performance is almost similar. This result substantiates the importance of *functionality* of relationships, partially explains the difference between the performance of Paris+ and BootEA. However, there are additional technical reasons that may justify the superiority of Paris+. For instance, Paris+ also finds relationship equivalences between KGs, which may have a positive influence on the entity alignment task.

• ***Empowering neural methods with functionality.*** While not being the main scope, we want to provide the reader with some first evidence that the performance of neural methods may increase by exploiting the functionality of relationships. To achieve this, we modify the loss function of BootEA in a simple and straightforward manner. The objective function of BootEA consists of a hinge loss that penalizes relation triples if their scores (as per a certain scoring function) are lower than a certain threshold (i.e. the margin of the hinge loss). Thus, it is this loss that guides the learning of entity embeddings. Our modification aims to focus the learning of entity embeddings towards relation triples whose relationships are highly functional. We modify the objective of BootEA so that the margin of the hinge loss is scaled with the relationship functionality. Additional technical details can be found in our technical report [36]. This simple modification of BootEA provides an improvement of 3 points in F1 score with respect to the original method on EN-JA-15K (XRealEA_Pure). We hypothesize that neural methods designed to better integrate functionality may translate into larger gains.

7.3 Recommendations for the Future

As stated previously, we have yet to witness the best of neural entity alignment methods. While we are convinced of the abilities of the community to identify the scenarios that will facilitate neural methods to showcase their true potential, we suggest the following interesting directions that might be worth exploring.

- Identification of ways to improve blocking techniques for generating datasets with both high precision and recall to serve as input for the RL methods. We experienced that blocking is crucial to the performance of RL methods, and thus, any improvements in this step may facilitate RL methods to obtain competitive or even superior performance than Paris+. Note that even EA methods, both neural and non-neural, can easily leverage blocking, which we recommend as an interesting avenue for exploration.
- Identification of ways to incorporate the key findings from the analysis presented in § 7.2, viz., (i) leveraging all attributes instead of just one, and (ii) incorporating the concept of functionality while modeling relationships. We believe that the aforementioned recommendations would involve non-trivial adaptations to the design of neural methods, possibly leading to novel design principles and advancements at a fundamental level.
- Identification of use cases or datasets wherein potentially matchable entities exhibit very low lexical similarity in their attribute values. This property will primarily hurt methods, such as Paris+, that depend to a certain extent on string similarity. Cross-lingual datasets emerge as likely candidates, and the aforementioned limitations are evident from our empirical analysis (cf. § 4.3), where BERT-INT (with descriptions) outperforms Paris+. That said, further investigation is recommended along these lines. Moving ahead, datasets derived from domain-specific KGs—usually marred with domain-specific jargons—such as those arising in legal, medical, or enterprise domains are also likely candidates.
- Identification of datasets wherein the majority of attributes are observed for most of the entities. As opposed to Paris+, which nicely handles sparsity in the attributes, neural methods, such as RDGCN, need to have a valid literal value for each selected attribute. Otherwise it resorts to imputation techniques to complete the missing attributes with substituted values.
- Devise methods that are amenable to the RealEA datasets, i.e., datasets that do not follow the unrealistic 1-to-1 assumption. While we introduce one simple solution—bidirectional entity matching (Alg. 1)—to enable neural methods to properly work in RealEA datasets, more sophisticated solutions to enrich the entity matching module of neural methods are warranted.
- Devising neural EA methods that scale gracefully to large real-world datasets.
- Lastly, the vast majority of non-neural methods were conceived for realistic settings that lack seed alignment information, and thus, they possess the ability to work in an unsupervised manner. On the contrary, neural methods lack this ability. Exploration of unsupervised neural EA methods is a nascent and important topic—the first method, EVA [41], was published in December 2020—that requires considerable attention from the community.

Before we conclude, it is important to clarify that we only consider methods that were published before March 2021. Neural entity alignment is, however, a highly active field of research, and the state of the art gets pushed almost on a quarterly basis. To this end, it is our aim to keep the associated GitHub repository (https://github.com/epfl-dlab/entity-matchers) updated with the inclusion of results from recent methods, as and when they get published. Talking about activity, we noticed that very recently (June 2021 to be specific) the first ever attempt to marry the design principles from the pre-neural and neural era was explored to propose a method called Prase [55]. On the one hand, it is unfortunate to not be able to include Prase in our study due to its recency. On the other hand, it is fortuitous to observe that the introduction of a method like Prase provides an early seal of approval to the recommendations provided by us in Sec. 7.3, underlining the fact that they are already being acted upon by a few researchers from the neural community.

We hope the insights obtained from this study will provide the directionality and clarity required for a more streamlined advancement in research on neural entity alignment.

ACKNOWLEDGMENTS

This project was partly funded by the Swiss National Science Foundation (grant 200021_185043), the European Union (TAILOR, grant 952215), and the Microsoft Swiss Joint Research Center. We also gratefully acknowledge generous gifts from Facebook and Google.

REFERENCES

[1] 2014. The RDF Turtle Format. https://www.w3.org/TR/turtle/. accessed: 11 November 2021.

[2] 2021. The Ontology Alignment Evaluation Initiative (OAEI). http://oaei.ontologymatching.org/. accessed: 11 November 2021.

[3] Nizal Alshammry and Phillip Lord. 2017. Identitas: A Better Way To Be Meaningless. In *Proceedings of the 8th International Conference on Biomedical Ontology (ICBO)*.

[4] A. Arasu, C. Ré, and Dan Suciu. 2009. Large-Scale Deduplication with Constraints Using Dedupalog. *2009 IEEE 25th International Conference on Data Engineering* (2009), 952–963.

[5] S. Auer, C. Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Z. Ives. 2007. DBpedia: A Nucleus for a Web of Open Data. In *ISWC/ASWC*.

[6] Bahar Ghadiri Bashardoost, R. Miller, Kelly Lyons, and F. Nargesian. 2020. Knowledge Translation. In *VLDB*, Vol. 13.

[7] Indrajit Bhattacharya and L. Getoor. 2007. Collective entity resolution in relational data. *ACM Trans. Knowl. Discov. Data* 1 (2007), 5.

[8] Christoph Böhm, Gerard de Melo, Felix Naumann, and G. Weikum. 2012. LINDA: distributed web-of-data-scale entity matching. In *CIKM '12*.

[9] K. Bollacker, C. Evans, Praveen Paritosh, Tim Sturge, and J. Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

[10] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *SIGMOD*. 1247–1250.

[11] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, J. Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-relational Data. In *NIPS*.

[12] Yixin Cao, Z. Liu, C. Li, Zhiyuan Liu, Juan-Zi Li, and Tat-Seng Chua. 2019. Multi-Channel Graph Neural Network for Entity Alignment. In *ACL*.

[13] Muhao Chen, Weijia Shi, Ben Zhou, and Dan Roth. 2021. Cross-lingual Entity Alignment with Incidental Supervision. In *EACL*. 645–658.

[14] Muhao Chen, Y. Tian, Kai-Wei Chang, S. Skiena, and C. Zaniolo. 2018. Co-training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-lingual Entity Alignment. In *IJCAI*.

[15] Muhao Chen, Y. Tian, Mohan Yang, and C. Zaniolo. 2017. Multilingual Knowledge Graph Embeddings for Cross-lingual Knowledge Alignment. In *IJCAI*.

[16] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2021. An Overview of End-to-End Entity Resolution for Big Data. *ACM Computing Surveys (CSUR)* 53 (2021), 1 – 42.

[17] Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and H. Jégou. 2018. Word Translation Without Parallel Data. In *ICLR*.

[18] Maurizio Ferrari Dacrema, Simone Boglio, P. Cremonesi, and D. Jannach. 2019. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ArXiv* (2019).

[19] Maurizio Ferrari Dacrema, P. Cremonesi, and D. Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *RecSys*.

[20] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.

[21] Pierre Senellart Fabian M. Suchanek, Serge Abiteboul. 2013. The PARIS Implementation. http://webdam.inria.fr/paris/releases/paris-0.3.gz. accessed: 20 January 2021.

[22] Daniel Faria, Catia Pesquita, Teemu Tervo, Francisco M. Couto, and Isabel F. Cruz. 2019. AML and AMLC Results for OAEI 2019. In *OM@ISWC*.

[23] Manaal Faruqui and Chris Dyer. 2014. Improving Vector Space Word Representations Using Multilingual Correlation. In *EACL*.

[24] D. Fensel, J. Hendler, H. Lieberman, and W. Wahlster. 2003. SPINNING THE SEMANTIC WEB INTRODUCTION.

[25] D. Gale and L. S. Shapley. 2013. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 120 (2013), 386 – 391.

[26] Alberto García-Durán, Antoine Bordes, and Nicolas Usunier. 2015. Composing Relationships with Translations. In *EMNLP*.

[27] J. Gracia, M. d'Aquin, and E. Mena. 2009. Large scale integration of senses for the semantic web. In *WWW*.

[28] Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann Publishers Inc.

[29] A. Isaac, L. V. D. Meij, S. Schlobach, and S. Wang. 2007. An Empirical Study of Instance-Based Ontology Matching. In *ISWC/ASWC*.

[30] Shaoxiong Ji, Shirui Pan, E. Cambria, P. Marttinen, and Philip S. Yu. 2020. A Survey on Knowledge Graphs: Representation, Acquisition and Applications. *ArXiv* abs/2002.00388 (2020).

[31] Ernesto Jiménez-Ruiz and B. C. Grau. 2011. LogMap: Logic-Based and Scalable Ontology Matching. In *ISWC*.

[32] Thomas Kipf and M. Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[33] Pradap Konda, Sanjib Das, C. PaulSuganthanG., AnHai Doan, A. Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeffrey F. Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. 2016. Magellan: Toward Building Entity Matching Management Systems. *Proc. VLDB Endow.* 9 (2016), 1197–1208.

[34] H. Kuhn. 2010. The Hungarian Method for the Assignment Problem. In *50 Years of Integer Programming*.

[35] S. Lacoste-Julien, K. Palla, Alex Davies, Gjergji Kasneci, T. Graepel, and Zoubin Ghahramani. 2013. SIGMa: simple greedy matching for aligning large knowledge bases. In *KDD*.

[36] Manuel Leone, Stefano Huber, Akhil Arora, Alberto García-Durán, and Robert West. 2022. A Critical Re-evaluation of Neural Methods for Entity Alignment. *CoRR* (2022). https://go.epfl.ch/EA_ReEval_full.pdf

[37] Adam Lerer, Ledell Wu, Jiajun Shen, Timothée Lacroix, Luca Wehrstedt, Abhijit Bose, and Alex Peysakhovich. 2019. Pytorch-BigGraph: A Large Scale Graph Embedding System. In *MLSys*.

[38] Juan-Zi Li, J. Tang, Y. Li, and Q. Luo. 2009. RiMOM: A Dynamic Multistrategy Ontology Alignment Framework. *IEEE Transactions on Knowledge and Data Engineering* 21 (2009), 1218–1232.

[39] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang Chiew Tan. 2020. Deep entity matching with pre-trained language models. *Proc. VLDB Endow.* 14 (2020), 50 – 60.

[40] Beatriz Lima, Daniel Faria, Francisco M. Couto, Isabel F. Cruz, and Catia Pesquita. 2020. OAEI 2020 results for AML and AMLC. In *OM@ISWC*.

[41] Fangyu Liu, Muhao Chen, Dan Roth, and Nigel Collier. 2021. Visual Pivoting for (Unsupervised) Entity Alignment.

[42] D. Lukovnikov, Asja Fischer, Jens Lehmann, and S. Auer. 2017. Neural Network-based Question Answering over Knowledge Graphs on Word and Character Level. *WWW* (2017).

[43] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

[44] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2021. Boosting the Speed of Entity Alignment 10 ×: Dual Attention Matching Network with Normalized Hard Sample Mining. In *WWW*. 821–832.

[45] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting Similarities among Languages for Machine Translation. *ArXiv* abs/1309.4168 (2013).

[46] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep Learning for Entity Matching: A Design Space Exploration. *SIGMOD* (2018).

[47] Maximilian Nickel and Douwe Kiela. 2017. Poincaré Embeddings for Learning Hierarchical Representations. In *NIPS*.

[48] M. Nickel, Kevin Murphy, Volker Tresp, and Evgeniy Gabrilovich. 2016. A Review of Relational Machine Learning for Knowledge Graphs. *Proc. IEEE* 104 (2016), 11–33.

[49] Jan Nößner, Mathias Niepert, Christian Meilicke, and H. Stuckenschmidt. 2010. Leveraging Terminological Structure for Object Reconciliation. In *ESWC*.

[50] George Papadakis, Georgia Koutrika, Themis Palpanas, and Wolfgang Nejdl. 2014. Meta-Blocking: Taking Entity Resolutionto the Next Level. *IEEE TKDE* 26, 8 (2014), 1946–1960.

[51] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. 2016. Comparative Analysis of Approximate Blocking Techniques for Entity Resolution. *PVLDB* 9, 9 (2016), 684–695.

[52] H. Paulheim. 2017. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web* 8 (2017), 489–508.

[53] Shichao Pei, L. Yu, R. Hoehndorf, and X. Zhang. 2019. Semi-Supervised Entity Alignment via Knowledge Graph Embedding with Awareness of Degree Difference. In *WWW*.

[54] Jan Portisch, Michaela Hladik, and Heiko Paulheim. 2020. ALOD2Vec matcher results for OAEI 2020. In *OM@ISWC*.

[55] Zhiyuan Qi, Ziheng Zhang, Jiaoyan Chen, Xi Chen, Yuejia Xiang, Ningyu Zhang, and Yefeng Zheng. 2021. Unsupervised Knowledge Graph Alignment by Probabilistic Reasoning and Semantic Embedding. *CoRR* abs/2105.05596 (2021). https://arxiv.org/abs/2105.05596 To appear in IJCAI 2021.

[56] Vibhor Rastogi, Nilesh N. Dalvi, and M. Garofalakis. 2011. Large-Scale Collective Entity Matching. *Proc. VLDB Endow.* 4 (2011), 208–218.

[57] Matteo Antonio Senese, Giuseppe Rizzo, Mauro Dragoni, and Maurizio Morisio. 2020. MTSI-BERT: A Session-aware Knowledge-based Conversational Agent. In *LREC*. 717–725.

[58] M. Speranskaya, Martin Schmitt, and Benjamin Roth. 2020. Ranking vs. Classifying: Measuring Knowledge Base Completion Quality. In *AKBC*.

[59] Fabian M. Suchanek, S. Abiteboul, and P. Senellart. 2011. PARIS: Probabilistic Alignment of Relations, Instances, and Schema. *ArXiv* abs/1111.7164 (2011).

[60] Fabian M. Suchanek, Gjergji Kasneci, and G. Weikum. 2007. Yago: a core of semantic knowledge. In *WWW*.

[61] Zequn Sun. 2020. The OpenEA Library. https://github.com/nju-websoft/OpenEA. accessed: 20 January 2021.

[62] Zequn Sun, Muhao Chen, Wei Hu, Chengming Wang, Jian Dai, and Wei Zhang. 2020. Knowledge Association with Hyperbolic Knowledge Graph Embeddings. In *EMNLP*. 5704–5716.

[63] Zequn Sun, Wei Hu, and C. Li. 2017. Cross-Lingual Entity Alignment via Joint Attribute-Preserving Embedding. In *ISWC*.

[64] Zequn Sun, Wei Hu, Qingheng Zhang, and Y. Qu. 2018. Bootstrapping Entity Alignment with Knowledge Graph Embedding. In *IJCAI*.

[65] Zequn Sun, JiaCheng Huang, W. Hu, Muhao Chen, Lingbing Guo, and Yuzhong Qu. 2019. TransEdge: Translating Relation-Contextualized Embeddings for Knowledge Graphs. In *ISWC*.

[66] Zequn Sun, Qingheng Zhang, Wei Hu, Cheng-Ming Wang, Muhao Chen, F. Akrami, and C. Li. 2020. A benchmarking study of embedding-based entity alignment for knowledge graphs. *Proceedings of the VLDB Endowment* 13 (2020), 2326 – 2340.

[67] X. Tang, Jian zhong Zhang, B. Chen, Y. Yang, Hong Chen, and Cuiping Li. 2020. BERT-INT: A BERT-based Interaction Model For Knowledge Graph Alignment. In *IJCAI*.

[68] Bayu Distiawan Trisedya, Jianzhong Qi, and Rui Zhang. 2019. Entity Alignment between Knowledge Graphs Using Attribute Embeddings. In *AAAI*.

[69] G. Tummarello, Richard Cyganiak, Michele Catasta, Szymon Danielczyk, Renaud Delbru, and S. Decker. 2010. Sig.ma: Live views on the Web of Data. *J. Web Semant.* 8 (2010), 355–364.

[70] O. Udrea, L. Getoor, and R. Miller. 2007. Leveraging data and structure in ontology integration. In *SIGMOD '07*.

[71] Denny Vrandecic and M. Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Commun. ACM* 57 (2014), 78–85.

[72] S. Wang, G. Englebienne, and S. Schlobach. 2008. Learning Concept Mappings from Instance Similarity. In *ISWC*.

[73] Xiang Wang, Dingxian Wang, Canran Xu, X. He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable Reasoning over Knowledge Graphs for Recommendation. In *AAAI*.

[74] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-Aware Entity Alignment for Heterogeneous Knowledge Graphs. In *IJCAI*.

[75] Yuting Wu, Xiao Liu, Yansong Feng, Z. Wang, and Dongyan Zhao. 2019. Jointly Learning Entity and Relation Representations for Entity Alignment. In *EMNLP/IJCNLP*.

[76] Chao Xing, D. Wang, Chao Liu, and Yiye Lin. 2015. Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation. In *HLT-NAACL*.

[77] Kun Xu, L. Wang, Mo Yu, Yansong Feng, Y. Song, Zhi guo Wang, and Dong Yu. 2019. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network. *ACL*.

[78] Jiaxuan You, Xiaobai Ma, Daisy Yi Ding, Mykel J. Kochenderfer, and Jure Leskovec. 2020. Handling Missing Data with Graph Representation Learning. In *NeurIPS*.

[79] Weixin Zeng, X. Zhao, J. Tang, and Xuemin Lin. 2020. Collective Entity Alignment via Adaptive Features. *2020 IEEE 36th International Conference on Data Engineering (ICDE)* (2020), 1870–1873.

[80] Qingheng Zhang, Zequn Sun, W. Hu, Muhao Chen, Lingbing Guo, and Y. Qu. 2019. Multi-view Knowledge Graph Embedding for Entity Alignment. In *IJCAI*.

[81] Ziheng Zhang, Jiaoyan Chen, X. Chen, H. Liu, Yuejia Xiang, B. Liu, and Yefeng Zheng. 2020. An Industry Evaluation of Embedding-based Entity Alignment. In *COLING*.

[82] Xiang Zhao, Weixin Zeng, J. Tang, W. Wang, and Fabian M. Suchanek. 2020. An Experimental Study of State-of-the-Art Entity Alignment Approaches. *IEEE TKDE* (2020).

[83] Da Zheng, Xiang Song, Chao Ma, Zeyuan Tan, Zihao Ye, J. Dong, Hao Xiong, Zheng Zhang, and G. Karypis. 2020. DGL-KE: Training Knowledge Graph Embeddings at Scale. *Proceedings of ACM SIGIR* (2020).

[84] Hao Zhu, Ruobing Xie, Zhiyuan Liu, and M. Sun. 2017. Iterative Entity Alignment via Joint Knowledge Embeddings. In *IJCAI*.

[85] Qi Zhu, Hao Wei, Bunyamin Sisman, Da Zheng, C. Faloutsos, X. Dong, and Jiawei Han. 2020. Collective Multi-type Entity Alignment Between Knowledge Graphs. *Proceedings of The Web Conference 2020* (2020).

[86] Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. 2021. Relation-Aware Neighborhood Matching Model for Entity Alignment. In *AAAI*. 4749–4756.

[87] Zhaocheng Zhu, Shizhen Xu, Jian Tang, and Meng Qu. 2019. GraphVite: A High-Performance CPU-GPU Hybrid System for Node Embedding. In *WWW*. 2494–2504.

[88] Y. Zhuang, G. Li, Zhuojian Zhong, and Jianhua Feng. 2016. PBA: Partition and Blocking Based Alignment for Large Knowledge Bases. In *DASFAA*.