BEAM: A First Benchmark for Microdata Entity Alignment with Knowledge Graphs

Hanane Kteich
hanane.kteich@lisn.fr
LISN, CNRS (UMR9015)
University Paris-Saclay, France

Gianluca Quercini
gianluca.quercini@lisn.fr
LISN, CNRS (UMR9015)
University Paris-Saclay, France

Joe Raad
joe.raad@lisn.fr
LISN, CNRS (UMR9015)
University Paris-Saclay, France

Fatiha Sais
fatiha.sais@lisn.fr
LISN, CNRS (UMR9015)
University Paris-Saclay, France

Abstract

Nearly half of all web pages contain semi-structured data (RDFa, microdata, JSON-LD), yet this information remains poorly aligned with public knowledge graphs (KGs). Existing entity alignment (EA) benchmarks, typically derived from structured KGs such as DBpedia, YAGO, and Wikidata, represent idealized settings with high schema overlap and dense link structures, conditions that rarely occur in realistic cross-KG scenarios. To address this gap, we introduce *BEAM*, a microdata benchmark for entity alignment methods between Web Data Commons microdata and Wikidata.

Unlike prior benchmarks that rely on potentially erroneous `owl:sameAs` links, *BEAM* establishes ground-truth alignments through *key-based matching* (e.g., IATA codes for airports, ISBNs for books), providing reliable identity resolution. We retain much of the noise, heterogeneity, and structural sparsity of web data, rather than artificially cleaning or rebalancing the graphs. As a proof of concept, the current release covers two classes (`Airport`, `Book`) and is accompanied by a reusable pipeline to extend BEAM to further classes when suitable keys exist.

Experimental results show a substantial drop in the performance of state-of-the-art EA models on *BEAM* compared to curated benchmarks, revealing their limited robustness in unstructured web KGs and highlighting the importance of realistic evaluation settings. *BEAM* is publicly available and adheres to the *FAIR principles* (Findable, Accessible, Interoperable, Reusable), providing a reproducible foundation for advancing research in entity alignment between semi-structured web data and knowledge graphs.

CCS Concepts

• Information systems → Information integration; **• General and reference → Evaluation**.

Keywords

Semantic Web, Knowledge graphs, Microdata, Entity alignment

SAC '26, Thessaloniki, Greece

ACM ISBN 979-8-4007-2294-3/2026/03
https://doi.org/10.1145/3748522.3779966

ACM Reference Format:
Hanane Kteich, Gianluca Quercini, Joe Raad, and Fatiha Sais. 2026. BEAM: A First Benchmark for Microdata Entity Alignment with Knowledge Graphs. In *The 41st ACM/SIGAPP Symposium on Applied Computing (SAC '26), March 23–27, 2026, Thessaloniki, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3748522.3779966

1 Introduction

In this paper, we use the term *benchmark* to denote a reusable, publicly available dataset together with its construction protocol and evaluation scripts, designed to compare methods on a clearly defined task. In our case, the task is to align web microdata with a reference knowledge graph.

Knowledge graphs (KGs) underpin a wide range of applications in several domains, including information retrieval, question answering, and data integration. A central challenge in this context is *entity alignment* (EA), i.e., aligning entities in different KGs that refer to the same real-world object. Embedding-based EA methods have recently shown impressive results on benchmarks such as DBP15K and OpenEA [12, 15], where different subsets (e.g., ZH–EN, JA–EN, FR–EN) are sampled from DBpedia, YAGO, and Wikidata. These benchmarks contain tens of thousands of entities, exhibit relatively high schema overlap, and rely on pre-existing `owl:sameAs` links as ground truth.

However, these commonly used datasets are manually structured and adapted to ideal EA scenarios that are far from the noisy, heterogeneous graphs found "in the wild". For instance, DBP15K subsets contain only about 15k aligned pairs; FB15K and FB15K-237 [2, 19] have about 15k entities and 592k triples, which is much smaller and cleaner than modern KGs with millions of entities and facts. OpenEA improves realism via Iterative Degree-based Sampling (IDS) [15], but still starts from well-curated KGs and heavily relies on `owl:sameAs` links. Recent studies on domain-specific KGs (e.g., DOREMUS and AGROLD) report strong performance drops for models trained on DBP15K when evaluated on more heterogeneous data [27, 1, 21]. This suggests that existing benchmarks may overestimate the robustness of current EA methods.

At the same time, more and more websites embed structured microdata using the `schema.org`[1] vocabulary and formats such as microdata, RDFa, and JSON-LD, which we collectively refer

[1] `Schema.org` is a collaborative project led by major search engines providing a common vocabulary for annotating web pages.

to as "microdata". The Web Data Commons (WDC) project [3] extracts such microdata at scale from the Common Crawl. For example, the 2018 extraction reports that 944 million pages out of 2.5 billion (37.1%) contain structured data [24], and more recent reports indicate that around half of all pages now include semantic microdata [20]. Despite this prevalence, microdata remains weakly integrated with public KGs: search engines still treat web content and KGs as largely separate resources.

Aligning microdata with KGs such as Wikidata would enable "search by Things rather than strings": given a KG entity, one could retrieve web pages that describe it and enrich search with attributes such as `schema:price` or `schema:location`. Yet existing EA benchmarks focus almost exclusively on pairs of structured KGs and ignore semi-structured microdata. To the best of our knowledge, there is no publicly available benchmark that systematically evaluates alignment between WDC-style microdata and Wikidata.

In this paper, we present BEAM[2], the first benchmark for entity alignment between Web Data Commons microdata and Wikidata. As a proof of concept, we currently focus on two classes, `schema:Airport` and `schema:Book`, for which globally used identifiers (IATA codes and ISBNs) make it possible to build high-precision ground truth via *key-based matching*. Many other classes (e.g., persons, events, organizations) lack a single universally adopted identifier; for those, our pipeline can still be applied in combination with key-discovery tools, but the construction of reliable ground truth is more challenging and remains an important direction for future work.

Our contributions are as follows:

- We construct class-specific web datasets (Airports, Books) by combining WDC microdata and Wikidata through key-based entity matching (e.g., IATA code for airports). Unlike previous benchmarks, we preserve the noise, heterogeneity, and size disparities inherent to real web graphs, and we explicitly remove the key triples used for alignment to avoid trivial solutions.
- We evaluate five representative embedding-based EA models (MTransE, AliNet, GCN-Align, AlignE, BootEA) using the same hyperparameters as in the OpenEA study [15]. On DBP15K EN–FR-15K (V1), these models reach Hits@5 scores between 0.47 and 0.72 [15], whereas on BEAM their Hits@5 drops below 0.03, revealing a drastic performance gap.
- We publicly release our benchmark, accompanied by a step-by-step guide and scripts for processing a class (illustrated with the `Airport` class) under a permissive license. The shared resource complies with the FAIR principles [25] and is designed to be extensible: practitioners can plug in additional classes, keys, and baselines, including recent transformer-based and entity-matching approaches.
- We further provide a generic and parameterizable construction pipeline: given a target class (e.g., `schema:SportsEvent`) and a candidate key, the same sequence of steps (extraction, cleaning, expansion, Wikidata querying, gold-standard generation) can be reproduced. We are currently developing an automation tool that will generate a new benchmark instance from the parameters "class" and "key", ensuring methodological consistency and FAIR-compliant releases for future versions of BEAM.

The remainder of the paper is organized as follows. Section 2 discusses related work in EA, entity matching, and microdata. Section 3 describes the benchmark construction. Section 4 reports experimental results, and Section 5 compares BEAM with existing benchmarks. Sections 6 and 7 discuss FAIR compliance and ethical considerations. Section 8 concludes and outlines future work.

2 Related Work

Entity alignment benchmarks. The most widely used datasets for embedding-based EA are *DBP15K*, *DWY15K*, and *DY15K*, constructed from DBpedia, Wikidata, and YAGO, respectively. OpenEA [15] derives 15K- and 100K-sized subsets with varying sparsity (V1/V2) using the Iterative Degree-based Sampling (IDS) algorithm and uses existing `owl:sameAs` links as ground truth. These benchmarks are relatively clean, schema-aligned, and balanced, which makes them convenient but somewhat idealized; many models achieve Hits@5 above 0.7 on EN–FR-15K (V1) [15]. Subsequent datasets such as SRPRS and related sparse variants attempted to increase sparsity or realism, but still start from curated KGs and `sameAs`-based links.

Recent work has examined more heterogeneous settings. DAEA [27] and related studies show that models tuned on DBP15K degrade substantially on domain-specific KGs such as DOREMUS and AGROLD [1, 21], confirming that synthetic benchmarks do not fully reflect real-world heterogeneity. However, these datasets still involve pairs of structured KGs; semi-structured web microdata has been largely ignored in EA evaluation.

Families of EA models. Embedding-based EA methods can be broadly grouped into three families. (i) *Translation-based* approaches such as MTransE [4], JAPE [12], BootEA [13], and AlignE learn a shared embedding space from relation and attribute triples. (ii) *GNN-based* models such as GCN-Align [23], RDGCN [26], AttrGNN [9], and AliNet [14] leverage graph convolutions and attention to aggregate neighborhood information. (iii) *Language-model-enhanced* approaches (e.g., BERT-INT [18], DERA [22], TEA-style models [28], LightEA [11]) incorporate pretrained language models to encode labels and descriptions. All three families are typically evaluated on DBP15K/OpenEA, sometimes augmented with description corpora from DBpedia and Wikidata.

Self-supervised and semi-supervised EA, such as SelfKG [8], JTEA [10], CPL-OT [5], and LLM-assisted methods like ChatEA [6] and HLMEA [7], aim to reduce reliance on large labeled alignment sets or to exploit large language models for interactive refinement. These approaches remain largely unexplored on microdata-based benchmarks.

Entity matching and record linkage. EA is closely related to entity matching and record linkage in databases, where the goal is to detect duplicate or corresponding records across heterogeneous sources. Traditional approaches exploit similarity in names, attributes, and sometimes relational neighborhoods, and are evaluated on datasets such as DBLP–ACM, DBLP–Scholar, or WDC product matching corpora. Our setting shares with this line of work the focus on

[2] Code and data available at: https://github.com/hananekth/BEAM-A-First-Benchmark-for-Knowledge-Graph-Entity-Alignment-with-Microdata

noisy, heterogeneous descriptions, but differs in that microdata entities participate in explicit RDF graphs and must be aligned to a large open KG (Wikidata) rather than to another table or catalog. In future extensions of BEAM we plan to include baselines from the entity matching literature as well as recent transformer-based alignment models, allowing a unified comparison across EA and entity matching techniques.

Microdata and Web Data Commons. Web Data Commons provides large-scale extractions of RDFa, microdata, and JSON-LD from the Common Crawl [3, 24]. Existing WDC corpora have been widely used for tasks such as product matching, schema analysis, and extraction quality assessment, but not yet for systematic EA with Wikidata. Our benchmark fills this gap by building class-specific microdata–Wikidata alignment sets, preserving much of the noise and incompleteness of web annotations while providing reproducible ground truth via key-based matching.

3 Benchmark Construction

Our benchmark aims to facilitate the task of aligning entities between the Web Data Commons (WDC) microdata corpus and Wikidata. As a proof of concept, we focus on the `schema:Airport` and `schema:Book` classes, and we provide tools and documentation that allow reproducing the same pipeline for other classes (e.g., `SportsEvent`, `StadiumOrArena`). The overall construction process, involves five stages: (i) microdata extraction, (ii) cleaning and filtering, (iii) graph expansion, (iv) Wikidata extraction, and (v) ground-truth generation.

3.1 Microdata extraction

We begin by importing class-specific subsets from the WDC website.[3] The Schema.org extractions are distributed in chunks, each containing quads in N-Quads format. For instance, the `Airport` subset comprises 53,684,719 RDF quads extracted from 173,702 URLs across 1,003 hosts (the 2018 WDC corpus as a whole contains 556 million pages and over 20 billion RDF quads [3]). We transform these quads into subject–predicate–object triples to simplify subsequent processing and integration with Wikidata.

3.2 Cleaning and filtering

Although WDC extractions are organized by class, each subset contains not only entities of the target type but also related types. For example, the `Airport` subset includes not only `schema:Airport` (3.56M instances) but also `schema:GeoCoordinates`, `schema:Flight`, `schema:Airline`, and `schema:Offer`. This reflects the natural co-occurrence of concepts on web pages. To ensure consistency, we retain only triples whose subjects are explicitly typed as `schema:Airport` (resp. `schema:Book`).

The retained triples include both generic properties (e.g., `name`, `description`, `url`, `image`) and class-specific properties (e.g., `location`, `country`, `iataCode` for airports; `author`, `isbn` for books). For benchmarking purposes we filter out triples that provide little information for alignment, such as outbound `url` links, images, or logos, while preserving the entity identifier so that tracing is still possible. This step reduces noise and dimensionality while maintaining reproducibility.

We normalize literals by keeping only English values for textual properties (`name`, `description`) and deduplicate identical triples. Unlike existing EA benchmarks that rely heavily on `owl:sameAs`, WDC extractions contain very few explicit alignments to Wikidata; therefore we construct a ground truth via key-based matching. Keys are unique or near-unique identifiers that allow linking across datasets. For `schema:Airport` the natural choice is the IATA code, widely used both in WDC and in Wikidata. For `schema:Book` the ISBN plays an equivalent role, but only for books for which this information is present in both sources.

After merging all WDC chunks for a given class, we unify triples that belong to subjects sharing the same key. This aggregation produces richer and more complete entity descriptions, closer to their representations in Wikidata. We discard entities described by fewer than three triples, since these typically contain only a type assertion, a key property, and a single other attribute, which provides insufficient context for alignment.

For training we split cleaned triples into two files: one containing attribute triples (where the object is a literal) and the other containing relational triples (where the object is an IRI). This separation mirrors the processing applied to Wikidata and facilitates usage by embedding models.

3.3 Graph expansion

Microdata often encodes information indirectly through linked nodes. For example, an airport entity may be linked to a `GeoCoordinates` node containing latitude and longitude, or to a `Location` node with nested country information. To capture such information we expand entity graphs to include nested nodes up to a bounded distance. For the `Airport` and `Book` classes, we observed that WDC folders encapsulate at most four hop distances per entity, and we therefore keep all nodes and triples within this radius. This preserves relevant context while keeping the graph size manageable. The choice of depth = 4 is empirical and is driven by the shallow structure of WDC microdata for these classes: in practice, all observed graphs are fully covered within four hops. For more densely connected classes such as `schema:SportsEvent` or `schema:LocalBusiness`, the depth can be increased (e.g., to 5 or 6) without loss of performance. In the automation tool we are developing, this parameter will be configurable so that future users can adjust it after inspecting the structure of their target class.

3.4 Wikidata extraction

The Wikidata side of the benchmark is obtained via SPARQL queries to the public endpoint. Concretely, we enumerate entities of the target class (such as `Airport`, which corresponds to `wd:Q1248784` in Wikidata) using the `instance-of` relation (`wdt:P31`). For each entity, denoted by s, we retrieve outgoing triples `<s> ?p ?o`, keeping only English literals. The result is split into two views: *attributes* (where `?o` is a literal) and *relations* (where `?o` is an IRI). Because the alignment relies on keys, we retain only class instances that carry at least one key predicate (e.g., IATA code for airports, ISBNs for books) and discard subjects without keys.

[3] https://webdatacommons.org/

To reduce noise and focus on semantically meaningful elements of the graph, as opposed to administrative predicates, we build a frequency dictionary over all property IRIs and filter out rare or administrative predicates. Specifically, we (i) count occurrences per property IRI, (ii) save an output file with a dictionary of property English `rdfs:label` for human inspection, and (iii) exclude meta-properties (e.g., versioning and statement counters) and low-frequency predicates. For the airport class, for example, we keep high-frequency predicates as well as low-frequency but semantically important properties such as short or official names and location.

Finally, because Wikidata represents both properties and many objects as first-class entities, we enrich the graph one distance further: for every property IRI and relational object IRI that appears at distance 1 from a subject, we fetch its English `rdfs:label` and `schema:description` and add them as literals. This produces triples that are directly comparable to WDC literals without requiring schema-level lookups at training time.

3.5 Ground truth generation

Although some WDC microdata include `schema.org/sameAs` assertions, these rarely point to Wikidata link instances of the target class. In many cases, the few `schema:sameAs` links present in WDC refer to related but different objects (e.g., offers, tickets, products, or external landing pages) rather than to the canonical entity we wish to align. As a result, they are not suitable as a reliable source of ground truth. We therefore use *key-based matching*. Keys are globally recognized identifiers. Because WDC subjects are automatically generated per page and may duplicate the same entity under different URIs, we first identify candidate keys from Wikidata for the chosen class and then check whether these keys appear in WDC files. This allows merging duplicate WDC entities and aligning them to their Wikidata counterparts.

For the class `Airport` the IATA code (`schema:iataCode` in WDC, `wdt:P238` in Wikidata) is a reliable key. For the class `Book` the ISBN plays an equivalent role; however, not all books have ISBNs defined in Wikidata or in WDC, and we exclude such cases from our evaluation. For classes where keys are less obvious, automatic key-discovery tools such as SAKEY [16] and VICKEY [17] can assist, but the existence of a single global key cannot be assumed in general. Importantly, we can tell that keys serve to construct a high-precision gold standard in a setting where `schema:sameAs` links are rare or unreliable.

Example. A WDC snippet may describe Charles de Gaulle airport as:

```
airport1 rdf:type  schema:Airport ;
schema:name "Charles de Gaulle Airport" ;
schema:iataCode "CDG" .
```

while Wikidata contains:

```
wd:Q8685 wdt:P31 wd:Q1248784 ;
wdt:P238 "CDG" ;
rdfs:label "Charles de Gaulle Airport"@en .
```

Since both entities share the IATA code `"CDG"`, they are aligned in our ground truth set.

The linking pipeline proceeds in three steps: (i) extract key–subject mappings from WDC, (ii) extract key–subject mappings from Wikidata, and (iii) intersect the key spaces to generate aligned pairs. To ensure robustness, keys are normalized (lowercased, stripped of quotes) and duplicate WDC subjects are merged. This procedure yields precise, reproducible alignments that reflect class-specific, large-scale web data. To avoid trivial leakage, we remove the key triples used for alignment (e.g., IATA codes, ISBNs) from both WDC and Wikidata graphs before training. Models are therefore evaluated on structural and descriptive signals rather than on the identifiers themselves.

Table 1: Statistics of the benchmark. "Attr." and "Rel." denote attribute and relation triples; "Alignment links" denotes ground-truth alignments.

Class	*Airport*	*Book*
Triples attr. WDC	6,728	206
Triples rel. WDC	28,973	70
Triples attr. Wikidata	61,090	573
Triples rel. Wikidata	163,517	651
Alignment links	2,526	82

Table 1 summarizes the statistics of our benchmark for the `Airport` and `Book` classes. The number of aligned instances is small compared to the total number of WDC microdata items in these classes: many web annotations either lack keys altogether or use identifiers that do not appear in Wikidata. This aligns with real-world expectations and is an important characteristic of the benchmark.

3.6 Dataset quality assessment

Key-based matching can only be as reliable as the key values themselves. To assess the quality of the automatically generated alignments, we therefore perform an initial manual validation. We randomly sample a set of aligned pairs from the Airport class and inspect the corresponding WDC snippets and Wikidata pages, comparing labels and descriptions. In this sample, we do not observe any incorrect alignments, which supports the assumption that IATA codes are globally unique and consistently used across the two sources. Potential errors can occur only if a key is mis-specified in WDC or Wikidata; our spot checks did not reveal any such inconsistencies for the current classes. We emphasize that key-based matching is used solely to construct a gold standard in the absence of reliable `schema:sameAs` links. As long as keys are correctly populated in both graphs, the resulting ground truth is precise.

4 Evaluation results

We evaluate five embedding-based EA models implemented in the OpenEA library:[4] MTransE [4], AliNet [14], AlignE [13], GCN-Align [23], and BootEA [13]. We adopt the same hyperparameters as in the OpenEA study [15]: for the 15K-sized dataset a batch size of 5,000 relation triples, a maximum of 2,000 epochs, and early stopping when Hits@1 decreases. Hits@k is the proportion of test entities for which the correct alignment appears in the top-k ranked candidates returned by a model. Ideally Hits@1 corresponds to strict accuracy, and Hits@5/50 capture the quality of the candidate ranking. For each version of our benchmark we perform five random folds and report the average Hits@5 and Hits@50.

On the curated EN–FR-15K (V1) subset of OpenEA, Sun et al. [15] report Hits@5 scores of 0.467 for MTransE, 0.589 for GCN-Align, and 0.718 for BootEA. In contrast, on BEAM-Airport the same models achieve Hits@5 between 0.012 and 0.022 and Hits@50 between 0.10 and 0.25 (Table 2). This gap of more than an order of magnitude is consistent across all evaluated models and also appears on the Book class (not shown due to space constraints).

[4] https://github.com/nju-websoft/OpenEA

Table 2: Average Hits@5 and Hits@50 (%) on BEAM-Airport (five folds; 70% training, 20% test, 10% validation).

	MTransE		AliNet		AlignE		GCN-Align		BootEA	
Dataset	Hits@5	Hits@50	Hits@5	Hits@50	Hits@5	Hits@50	Hits@5	Hits@50	Hits@5	Hits@50
Airport	1.58	10.29	1.24	10.20	1.38	9.99	1.78	12.07	2.17	25.28

Table 3: Comparison with existing EA benchmarks.

	OpenEA	*BEAM*
Domain	General-purpose	Class-specific
Cleanliness	Manually cleaned	Semi-automatic cleaning
Schema	Aligned / identical	Heterogeneous, partial overlap
Alignment method	`owl:sameAs`	Key-based alignment
Data source	DBpedia, YAGO, Wikidata	WDC, Wikidata
Evaluation splits	Fixed 15K / 100K	Driven by keys and data

These performance differences are directly linked to the structural and semantic disparities between BEAM and classical benchmarks. Datasets such as DBP15K and DWY15K are constructed from pairs of already homogeneous KGs (e.g., DBpedia–Wikidata or DBpedia–YAGO). Existing `owl:sameAs` links, often manually curated, provide dense and reliable alignments. IDS-based sampling preserves degree distributions and yields two subgraphs with strongly overlapping schemas and highly similar neighborhood structures: attributes are shared or have direct translations, and local graph topologies are comparable. Models thus operate in a near-ideal environment with little noise. In BEAM, by contrast, we align two fundamentally different sources: semi-structured, incomplete, and heterogeneous WDC microdata on the one hand, and a rich, coherent, strongly typed Wikidata graph on the other. The `schema.org` vocabulary only partially overlaps with Wikidata properties; neighborhood structures diverge significantly; and WDC entities contain on average only a few descriptive triples (typically between two and ten), compared to several dozens for their Wikidata counterparts. We deliberately retain real web noise, duplicates, lexical variants, inconsistent encodings, to measure the effective robustness of EA models. This structural and schema-level heterogeneity, combined with the absence of pre-aligned schemas, explains the drastic drop in Hits@k scores: it exposes the limitations of approaches designed for idealized scenarios.

These results indicate that methods tuned and validated on schema-aligned, `sameAs`-based benchmarks fail when confronted with the noisy, incomplete, and heterogeneous nature of web microdata. Compared to their performances on classical EA datasets, Hits@5 values on BEAM are negligible, empirically confirming that microdata alignment is substantially more challenging. Qualitative inspection further reveals that many WDC entities contain only a name and a very short description, with little structural context; in addition, schemas between WDC and Wikidata only partially overlap, and alignment keys have been removed from the graphs.

It is important to emphasize that our goal is not to show that these models are "bad", but rather that existing benchmarks are insufficient to assess their robustness. The architectures of MTransE, BootEA, GCN-Align, and AliNet were all designed for settings where two KGs share a substantial portion of their schema and where ground truth is derived from `owl:sameAs` links. BEAM deliberately violates these assumptions. We see this as an opportunity: the benchmark offers a realistic testbed for novel approaches that combine schema mapping, robust literal matching, and representation learning, including recent transformer-based and entity-matching methods, which we plan to evaluate in future work.

5 Comparison with existing benchmarks

Table 3 summarizes the main differences between BEAM and the OpenEA datasets. While OpenEA samples relatively clean KGs using IDS to preserve degree distributions [15], BEAM links web microdata to Wikidata under heterogeneous schemas, sparse links, and partially preserved noise.

OpenEA datasets have undeniably accelerated research on EA, but they represent idealized laboratory conditions: schemas are aligned or even identical, most entities have rich relational neighborhoods, and ground truth is derived from `owl:sameAs` links. Under these conditions, many embedding-based approaches reach Hits@5 scores above 0.6–0.7 on EN–FR-15K (V1) [15]. In contrast, BEAM is class-specific, constructed semi-automatically from Web Data Commons and Wikidata, and relies on key-based alignment rather than `sameAs`. The resulting graphs are much sparser, attributes are often incomplete, and schemas only partially overlap.

The performance drop observed in Section 4 (from Hits@5 around 0.7 on DBP15K/OpenEA to below 0.03 on BEAM) is therefore not surprising, but it is informative. It shows that current embedding-based EA methods—including those that perform best on OpenEA—are not yet robust enough for microdata–KG alignment at web scale. Our contribution is not a new EA model; instead, we provide a challenging, FAIR, and reproducible benchmark that exposes this gap and offers a concrete target for future methods. We see BEAM as complementary to OpenEA: models can be developed and tuned on existing structured benchmarks, but must eventually be stress-tested on more realistic data such as microdata and domain-specific KGs.

6 FAIR Principles and Availability

We follow the FAIR principles to promote the usability of our benchmark. Our dataset is *Findable*: we assign a DOI[5] and host it on Zenodo together with complete metadata. *Accessible*: all files, scripts, and documentation are publicly available under the CC BY license. *Interoperable*: triples are provided in RDF and OpenEA formats, facilitating import into graph databases or EA libraries. *Reusable*: we include clear provenance information and a permissive license to facilitate extension to other classes. Although the current release covers only two classes (Airport and Book), the construction scripts and documentation are generic and are intended to support future extensions to additional domains when suitable identifiers are available.

7 Ethical Considerations and Limitations

Our benchmark contains only publicly available data from WDC and Wikidata and therefore does not intentionally include personal or sensitive information. Nevertheless, linking noisy web data to canonical KGs may occasionally produce incorrect alignments due to errors in the microdata, ambiguous identifiers, or outdated annotations. Users should exercise caution when interpreting the results and avoid deploying models trained on BEAM in high-stakes settings without additional checks.

The current benchmark covers only two classes (Airport and Book), chosen because they have globally adopted identifiers (IATA codes and ISBNs) that enable precise key-based matching. This design yields high-precision alignments but biases the benchmark toward classes with strong identifiers; many important domains (e.g., persons, events, or organizations)

[5] https://doi.org/10.1145/3748522.3779966

lack such keys or use multiple competing identifiers. Extending BEAM to these domains will require careful key selection or discovery, as well as more extensive manual validation.

As discussed in Section 3.6, we conduct an initial manual sanity check of a random sample of aligned pairs, which indicates that most errors stem from noisy or inconsistent web annotations rather than from the matching procedure itself. Nonetheless, the benchmark is not error-free, and we encourage users to treat it as a realistic but imperfect testbed.

Finally, our experiments focus on a subset of embedding-based EA models implemented in OpenEA. We do not claim that these models are representative of all possible approaches, particularly recent transformer-based and entity-matching methods. Incorporating such baselines is part of our future work and will further refine our understanding of the strengths and weaknesses of existing techniques on microdata.

8 Conclusion and Future Work

We presented BEAM, a new benchmark that aligns Web Data Commons microdata with Wikidata to evaluate entity alignment under realistic web conditions. By relying on key-based matching rather than `owl:sameAs` links and by preserving the sparsity and noise of microdata, BEAM reveals substantial limitations of current embedding-based EA models. Our experiments show that models which perform strongly on classical benchmarks experience a dramatic performance drop on BEAM, highlighting the need for more robust alignment techniques.

Beyond the concrete datasets and evaluation scripts, BEAM is intended as a starting point for a broader research agenda on microdata–KG alignment. In the short term, we plan to (i) extend the benchmark to additional classes by automating the construction process into a reusable benchmark generation tool, and (ii) conduct a more extensive experimental study including transformer-based EA models and entity-matching baselines from the record linkage literature. This will enable a systematic comparison of different model families across classes and domains.

At a methodological level, our pipeline is generic and independent of the particular class or key: the extraction, cleaning, expansion, querying, and alignment steps remain the same, and only the parameters "class" and "key" change. When no explicit key is known, key discovery becomes the first stage of the process. Tools such as SAKEY and VICKEY, which automatically mine (quasi-)unique keys from RDF graphs, can be integrated to propose candidate identifiers. This allows BEAM-style benchmarks to be generated even for classes that lack obvious global identifiers, although more extensive validation will then be required.

In the longer term, we aim to explore complementary evaluation dimensions, such as robustness to annotation errors in microdata, sensitivity to lexical noise, and transferability to other knowledge bases (e.g., DBpedia or YAGO). By situating itself at the intersection of the semantic web, information integration, and KG alignment, BEAM invites the community to rethink EA not as a problem solved on idealized graphs, but as an open challenge that must embrace the diversity, messiness, and dynamism of the Web itself.

References

[1] Manel Achichi, Pasquale Lisena, Konstantin Todorov, Raphaël Troncy, and Jean Delahousse. 2018. Doremus: a graph of linked musical works. In *International Semantic Web Conference*. Springer, 3–19.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

[3] Alexander Brinkmann, Anna Primpeli, and Christian Bizer. 2023. The web data commons schema. org data set series. In *Companion Proceedings of the ACM Web Conference 2023*, 136–139.

[4] Muhao Chen, Yingtao Tian, Mohan Yang, and Carlo Zaniolo. 2016. Multilingual knowledge graph embeddings for cross-lingual knowledge alignment. *arXiv preprint arXiv:1611.03954*.

[5] Qijie Ding, Daokun Zhang, and Jie Yin. 2022. Conflict-aware pseudo labeling via optimal transport for entity alignment. In *2022 IEEE International Conference on Data Mining (ICDM)*. IEEE, 915–920.

[6] Xuhui Jiang, Yinghan Shen, Zhichao Shi, Chengjin Xu, Wei Li, Zixuan Li, Jian Guo, Huawei Shen, and Yuanzhuo Wang. 2024. Unlocking the power of large language models for entity alignment. *arXiv preprint arXiv:2402.15048*.

[7] Xiongnan Jin, Zhilin Wang, Jinpeng Chen, Liu Yang, Byungkook Oh, Seungwon Hwang, and Jianqiang Li. 2025. Hlmea: unsupervised entity alignment based on hybrid language models. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 11. Vol. 39, 11888–11896.

[8] Xiao Liu, Haoyun Hong, Xinghao Wang, Zeyi Chen, Evgeny Kharlamov, Yuxiao Dong, and Jie Tang. 2022. Selfkg: self-supervised entity alignment in knowledge graphs. In *Proceedings of the ACM web conference 2022*, 860–870.

[9] Zhiyuan Liu, Yixin Cao, Liangming Pan, Juanzi Li, and Tat-Seng Chua. 2020. Exploring and evaluating attributes, values, and structures for entity alignment. *arXiv preprint arXiv:2010.03249*.

[10] Kai Lu, Jing Zhao, Lichao Ding, and Zenghao Hao. 2024. Jtea: implementing semi-supervised entity alignment using joint teaching strategies. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 3050–3055.

[11] Xin Mao, Wenting Wang, Yuanbin Wu, and Man Lan. 2022. Lightea: a scalable, robust, and interpretable entity alignment framework via three-view label propagation. *arXiv preprint arXiv:2210.10436*.

[12] Zequn Sun, Wei Hu, and Chengkai Li. 2017. Cross-lingual entity alignment via joint attribute-preserving embedding. In *International semantic web conference*. Springer, 628–644.

[13] Zequn Sun, Wei Hu, Qingheng Zhang, and Yuzhong Qu. 2018. Bootstrapping entity alignment with knowledge graph embedding. In *IJCAI* number 2018. Vol. 18.

[14] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, and Yuzhong Qu. 2020. Knowledge graph alignment network with gated multi-hop neighborhood aggregation. In *Proceedings of the AAAI Conference on Artificial Intelligence* number 01. Vol. 34, 222–229. doi:10.1609/aaai.v34i01.5354.

[15] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, and Chengkai Li. 2020. A benchmarking study of embedding-based entity alignment for knowledge graphs. *arXiv preprint arXiv:2003.07743*.

[16] Danai Symeonidou, Vincent Armant, Nathalie Pernelle, and Fatiha Saıs. 2014. Sakey: scalable almost key discovery in rdf data. In *International Semantic Web Conference*. Springer, 33–49.

[17] Danai Symeonidou, Luis Galárraga, Nathalie Pernelle, Fatiha Saıs, and Fabian Suchanek. 2017. Vickey: mining conditional keys on knowledge bases. In *International Semantic Web Conference*. Springer, 661–677.

[18] Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, and Cuiping Li. 2020. Bert-int: a bert-based interaction model for knowledge graph alignment. *Interactions*, 100, e1.

[19] Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd workshop on continuous vector space models and their compositionality*, 57–66.

[20] Jarno van Driel. 2024. Stuffing pages with all of schema.org = a strategy-killing tactic. Accessed August 2025. (Mar. 2024). https://inlinks.com/insight/stuffing-pages-with-all-of-schema-org-a-strategy-killing-tactic/.

[21] Aravind Venkatesan, Gildas Tagny Ngompe, Nordine El Hassouni, Imene Chentli, Valentin Guignon, Clement Jonquet, Manuel Ruiz, and Pierre Larmande. 2018. Agronomic linked data (agrold): a knowledge-based system to enable integrative biology in agronomy. *PLoS One*, 13, 11, e0198270.

[22] Zhichun Wang and Xuan Chen. 2024. Dera: dense entity retrieval for entity alignment in knowledge graphs. *arXiv preprint arXiv:2408.01154*.

[23] Zhichun Wang, Qingsong Lv, Xiaohan Lan, and Yu Zhang. 2018. Cross-lingual knowledge graph alignment via graph convolutional networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, 349–357.

[24] 2018. Web data commons – rdfa, microdata, embedded json-ld, and microformats data sets – november 2018. Accessed August 2025. https://webdatacommons.org/structureddata/2018-12/stats/stats.html.

[25] Mark D. Wilkinson et al. 2016. The FAIR guiding principles for scientific data management and stewardship. *Scientific Data*, 3, 160018. doi:10.1038/sdata.2016.18.

[26] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, and Dongyan Zhao. 2019. Relation-aware entity alignment for heterogeneous knowledge graphs. *arXiv preprint arXiv:1908.08210*.

[27] Linyan Yang, Shiqiao Zhou, Jingwei Cheng, Fu Zhang, Jizheng Wan, Shuo Wang, and Mark Lee. 2025. Daea: enhancing entity alignment in real-world knowledge graphs through multi-source domain adaptation. In *The 31st International Conference on Computational Linguistics*. Association for Computational Linguistics, ACL, 5890–5901.

[28] Yu Zhao, Yike Wu, Xiangrui Cai, Ying Zhang, Haiwei Zhang, and Xiaojie Yuan. 2023. From alignment to entailment: a unified textual entailment framework for entity alignment. *arXiv preprint arXiv:2305.11501*.