# Comparing Symbolic and Embedding-Based Approaches for Relational Blocking

Daniel Obraczka[1,2][0000−0002−0366−9872] and Erhard Rahm[1,2][0000−0002−2665−1114]

[1] Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI) Dresden/Leipzig, Germany `https://scads.ai`
[2] Department of Computer Science, Leipzig University, Leipzig, Germany
`{obraczka,rahm}@informatik.uni-leipzig.de`

**Abstract.** Blocking aims to avoid unnecessary comparisons in a data matching pipeline. The heterogeneous nature of Knowledge Graphs (KG) is challenging for blocking approaches that traditionally were implemented for tabular data. While there is vast research on blocking approaches in general, the domain of KGs lacks systematic investigation, especially when comparing embedding-based and symbolic approaches. In this study, we generalize relational blocking, which incorporates neighborhood information of entities, to enable a variety of approaches across the neuro-symbolic spectrum. The results of our study are three-fold: (1) The relational enhancements to state-of-the-art approaches significantly improve their results. (2) (Neuro-)Symbolic approaches can outperform sophisticated deep-learning-based methods in terms of speed and quality. (3) Hybrid methods that combine symbolic and embedding-based techniques are promising avenues that have not been explored thoroughly yet. Our experiments were run on 16 real-world datasets of varying sizes with mono- and multi-lingual settings. We ensure statistical significance with a Bayesian analysis. We release our framework as open-source library.

**Keywords:** Entity Resolution · Knowledge Graphs · Blocking · Data Integration · Knowledge Graph Embedding · Entity Alignment

## 1 Introduction

Knowledge Graphs (KGs) have become a vital backbone for the information needs of the modern world. Complex tasks such as question answering [36] and recommendations [33] rely on high-quality data integration from multiple sources. The matching of KGs has seen wide research attention with approaches that use a variety of different methods [19,38,34]. A major performance bottleneck in the Entity Resolution (ER) step of the data integration process is comparing entity pairs from all sources, which in cases of binary matching has an a-priori quadratic complexity. Blocking aims to decrease this complexity and reduce the number of unnecessary comparisons by assigning likely matches into separate blocks. When dealing with databases, blocking approaches can often rely on the (relative) homogeneity of the underlying schemata, especially since entity resolution on tables most often only considers a single entity type (e.g. persons). KGs, on the other hand, contain a plethora of different entity types with vast heterogeneity across the schemata. Therefore, schema-agnostic approaches such as Token Blocking [25] were needed to match heterogeneous data sources [28]. While approaches initially relied on symbolic overlap across data sources to create blocks (e.g. by using common tokens), newer approaches utilize pre-trained word embeddings and even deep learning-based methods [35]. Originally from the database area, most blocking approaches are not built to utilize the rich relational information present in KGs. While Knowledge Graph Embeddings (KGEs) have been used as a method to encode this data in a lower-dimensional embedding space [34] and find matches via nearest neighbors, they have not been used for blocking. To tackle the blocking task in the KG domain, an approach needs to be schema-agnostic and able to utilize the relational information present in the KG. While there is already work [24] comparing embedding-based with symbolic methods on tabular data and approaches like MinoanER [7] that utilize relational information, but consist solely of symbolic techniques, there is no systematic study comparing such relational blocking techniques across the spectrum of neuro-symbolic approaches.

The main contributions of our study are, therefore, the following:

- We generalize the composite blocking scheme of MinoanER [7] to utilize not only symbolic but all techniques across the neuro-symbolic spectrum ranging from the sophisticated DeepBlocker [35] variants to approaches that rely on symbolic overlap like Token Blocking.
- Furthermore, we present a novel hybrid blocking approach that combines Token Blocking with token/attribute embedding clustering, which shows a promising future research direction.
- Our results show that the relational enhancements of our generalized framework significantly outperform their non-relational counterparts. Furthermore, the hybrid and purely symbolic methods outperform the sophisticated DeepBlocker variants as well as a state-of-the-art KGE method.
- Our experiments were performed on 16 real-world datasets. We ensured the significance of our results with a Bayesian signed rank test, and we released our framework as an open-source Python library to enable reproducibility and simplify future research [3].

We begin by defining some preliminaries, followed by a discussion of related work in Section 3. In Section 4, we present our framework and the integrated methods. After we present our experimental results in Section 5, we end with a conclusion and future work.

## 2 Preliminaries

Knowledge Graphs (KG) consist of triples in the form of (*entity*, *property*, *value*), where *property* can be either an attribute property or a relationship

[3] `https://github.com/dobraczka/klinker`

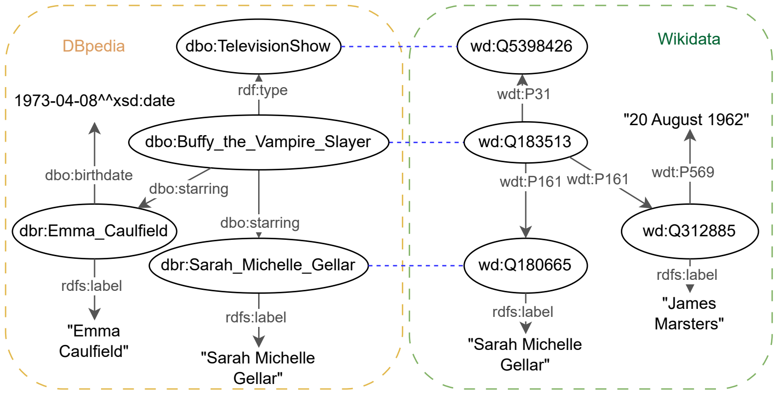

Fig. 1: Subgraphs of DBpedia and Wikidata. Blue dashed lines show entities that should be matched. Some URIs are shortened for brevity.

and *value* a literal or another entity, respectively. Therefore, a KG is a tuple $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{A}, \mathcal{L}, \mathcal{T})$, where $\mathcal{E}$ is the set of entities, $\mathcal{A}$ the set of attribute properties, $\mathcal{R}$ the set of relationship properties, $\mathcal{L}$ the set of literals, and $\mathcal{T}$ is the set of triples. We distinguish attribute triples $\mathcal{T}_A$ and relationship triples $\mathcal{T}_R$, where $\mathcal{T}_A : \mathcal{E} \times \mathcal{A} \times \mathcal{L}$ are triples connecting entities and literals, e.g. `(dbr:Emma_Caulfield, dbo:birthDate, "1973-04-08")` and $\mathcal{T}_R : \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ connect entities, e.g. `(dbr:Buffy_the_Vampire_Slayer, dbo:starring, dbr:Sarah_Michelle_Geller)` as seen in Figure 1. For a relation triple $(e_1, r, e_2)$, we will refer to $e_1$ as the *head* and $e_2$ as the tail entity. The task of Entity Resolution (ER) aims to find $\mathcal{M} = \{(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2 | e_1 \equiv e_2\}$, where $\equiv$ refers to the equivalence relation. ER has an a-priori quadratic complexity because every entity pair $(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2$ would have to be compared. Blocking aims to reduce this complexity by eliminating unnecessary comparisons by only clustering likely matches in a set of blocks $\mathcal{B}$. The number of comparisons in a set of blocks $\mathcal{B}$ is given by $||\mathcal{B}|| = \sum_{b_i \in \mathcal{B}} ||b_i||$, with $||b_i||$ denoting the number of comparisons in a single block $b_i$ [28]. A blocking approach aims to optimize two measures: $Recall(\mathcal{B}, \mathcal{M}) = truePositive(\mathcal{B})/|\mathcal{M}|$ shows the ratio of true positives compared to all matches. The Reduction Ratio $RR(\mathcal{B}, \mathcal{E}_1, \mathcal{E}_2) = ||\mathcal{B}||/(|\mathcal{E}_1||\mathcal{E}_2|)$ signifies the relation between comparisons with blocking versus without. Since either one of these measures can be optimized by making sacrifices in the other, the harmonic mean $h3r(rec, rr) = 2(rr * rec)/(rr + rec)$ can be used as an aggregate measure.

## 3 Related Work

Entity Resolution is a crucial step in the construction of KGs [13] and has seen significant research attention. Approaches range from probabilistic [32], clustering-based [31,30] over methods relying on traditional machine learning [21] all the way to transformer-based [12] and neuro-symbolic methods [23]. In the last years, a large family of techniques has relied on Knowledge Graph Embeddings (KGEs). These approaches encode entities from the given KGs into dense vectors and minimize the distance in the embedding space between similar entities. While many approaches are supervised, there are also models like LightEA [16], which can be used in the unsupervised setting. This specific approach relies on a three-view label propagation scheme, making it time-efficient while still achieving state-of-the-art quality. An overview of such approaches is given in these surveys [34,38]. The usefulness of such KGEs for blocking is largely unexplored.

Numerous approaches have been developed to perform blocking in a schema-agnostic manner on tabular data. Token Blocking [25] places entities into the same block that share a common token. Attribute Clustering Blocking [27] first clusters attributes across data sources by their values and then only places entities into the same block if the common tokens they share also belong to the same attribute cluster. Another notable work is the neural approach DeepBlocker [35], which we will describe in detail in Section 4.

A variety of approaches deal specifically with blocking on KGs. Prefix-Infix-Suffix Blocking [26] exploits naming patterns in the entities URI's. MultiBlock [14] first builds indices over multiple similarity measures, preserving the distances between entities. In the next phase, these indices are aggregated into a single multi-dimensional index. MinoanER [7] uses a composite blocking scheme, where entities are assigned to the same blocks if they (1) have the same *name* that is not used by any other entity, (2) common tokens or (3) their top neighbors share a common token. The *name* properties are defined by top-$k$ attributes whose literal values have high importance. Similarly, the top neighbors are connected to entities via relations that appear often but have many distinct values.

For a general overview of blocking methods, we recommend this survey [28]. Several benchmark studies also exist in the area of blocking. Efthymiou et.al.[8] show that Token and Attribute Clustering Blocking work well if datasets are derived from common sources, while Prefix-Infix-Suffix was more promising for more diverse data sources. In their study, they mention that utilizing neighborhood information could be an important signal for blocking algorithms. Zeakis et al. [37] investigate the usefulness of pre-trained embeddings for Entity Resolution in the domain of tabular data. While their study is not focused on blocking, it is one aspect of their analysis. They use averaged token embeddings and language models like BERT [6] or Sentence-BERT [29]. Their results suggest that the Sentence-BERT models are best suited for blocking. Papadakis et al. [24] investigate a variety of blocking techniques on tabular data in the schema-based and schema-agnostic settings. Similar to Zeakis et al. [37], they investigate embedding-based methods.

While there is work [24] on comparing deep learning-based approaches and symbolic blocking on tabular data, there is, to our knowledge, no systematic investigation of blocking approaches that utilize neighborhood information comparing embedding-based and symbolic approaches in the domain of KGs.

## 4 Method

### 4.1 General Framework

To compare embedding-based and non-embedding-based approaches in a common framework, our investigation relies on a relaxed generalization of the blocking scheme described in [7]. Our open-source Python implementation is named `klinker`. We use a composite blocking scheme

$$\Theta(\mathcal{KG}_1, \mathcal{KG}_2) = \Theta_A(\mathcal{C}_{a_1}, \mathcal{C}_{a_2}) \vee \Theta_R(\mathcal{C}_{r_1}, \mathcal{C}_{r_2}) \tag{1}$$

consisting of the *attribute* blocking function $\Theta_A$ and the *relational* blocking function $\Theta_R$. This blocking scheme is a relaxed variant of the one used in MinoanER [7] because it drops the unique name blocking scheme, and it is generalized because it allows any blocking function as $\Theta_A$ and $\Theta_R$. The concatenated attribute values of the most important attributes are signified as $\mathcal{C}_a$, and the concatenated attribute values of the most important neighbors are denoted as $\mathcal{C}_r$. In order to determine the importance of a relation or an attribute, we use the respective support and discriminability described by Efthymiou et al. [7].
The **support of a relation** $r \in \mathcal{R}$ is defined as $support(r) = \frac{|instances(r)|}{|\mathcal{E}|^2}$, with $instances(r) = \{(h,t)|(h,r,t) \in \mathcal{T}_R\}$ being the set of head and tail entity tuples of all relation triples that contain $r$.
The **support of an attribute** $a \in \mathcal{A}$ is defined as $support(a) = \frac{|heads(a)|}{|\mathcal{E}|}$, with $heads(a) = \{h|(h,a,t) \in \mathcal{T}_A\}$ being the set of head entities of the attribute triples that contain $a$.
The **discrimininablity** of a relation $r \in \mathcal{R}$ is defined as $discriminability(r) = \frac{|tails(r)|}{|instances(r)|}$, with $tails(r) = \{t|(h,r,t) \in \mathcal{T}_R\}$. The formula is analogous for an attribute $a \in \mathcal{A}$.
The **importance** of a relation $r \in \mathcal{R}$ is then defined as the harmonic mean of support and discriminability: $importance(r) = 2 \cdot \frac{support(r) \cdot discriminability(r)}{support(r) + discriminability(r)}$. The formula for the importance of an attribute $a \in \mathcal{A}$ is again analogous. We can now formally define the aforementioned concatenated attribute values of the entity and its most important neighbors.

$$\mathcal{C}_a = \{(e, l_c) | l_c = \bigoplus_{(e,a,l_i) \in \mathcal{T}_A \wedge a \in top_{n_a}(e)} l_i\} \tag{2}$$

Here, $\oplus$ represents the concatenation operation, and $top_{n_a}(e)$ is the $n_a$ attributes connected to entity $e$ with the highest importance score.

$$\mathcal{C}_r = \{(e_i, l_{c_r}) | l_{c_r} = \bigoplus_{r \in top_{n_r}(e_i) \wedge (e_i,r,e_j) \in \mathcal{T}_R \wedge (e_j, l_{c_j}) \in \mathcal{C}_a} l_{c_j}\} \tag{3}$$

with $top_{n_r}(e)$ being the $n_r$ relations connected to entity $e$ with the highest importance score.

## 4.2 Embedding-based Blockers

Blocking schemes that rely on embeddings are made up of two basic building blocks: *Frame Encoders*, that encode entities into an embedding matrix, and an *Embedding Block Builder* that creates blocks from the given embeddings.

**Frame Encoders** Each of the implemented Frame Encoders is an encoder function $\psi : \mathcal{C} \rightarrow \mathbb{R}^d$, which encodes each tuple $(e, l_c)$ into an embedding vector $\mathbf{v_e}$ of dimensionality $d$.

*Token Embedding Aggregation* Since entity attribute values often consist of a varying number of tokens, we need to aggregate token embeddings from pre-trained word embeddings into a uniform shape. We investigate two variants for aggregation. The first is the SIF (smooth inverse frequency) approach [2], which has been used successfully in previous blocking studies [35]. The second relies on sentence embeddings using siamese BERT-networks [29].

*DeepBlocker [35]* uses deep learning methods for blocking. In `klinker`, we adapted three methods from this framework: Autoencoder, Cross-Tuple Training (CTT), and hybrid. All of them rely on token embedding aggregation as the first step. The Autoencoder variant uses encoder and decoder layers in the following steps. The aggregated entity embedding $\mathbf{v_e}$ is used as input for the encoder, which outputs the hidden vector $\mathbf{u_e} \in \mathbb{R}^{d_u}$. Subsequently, the decoder uses $\mathbf{u_e}$ to produce the output $\mathbf{o_e} \in \mathbb{R}^{d_e}$. During training the model learns a concise representation for $\mathbf{u_e}$ by minimizing $||\mathbf{v_e} - \mathbf{o_e}||$. The Cross-Tuple Training variant learns to generate representations on entity pairs $e_1, e_2$ and associated match/non-match labels. Since no training data is provided, the pairs are generated synthetically. Given a tuple $(e, l_c) \in \mathcal{C}_a$ and it's tokens $w_e = tokens(l_c)$, a subset $w'_e$ of tokens in $w_e$ is selected in order the create a positive pair $(w_e, w'_e)$. A negative pair can be created by sampling another entity $e_n \neq e$. Analogous to the Autoencoder method, the aggregated entity embedding is used. Here, the embedding pairs $(\mathbf{v_{e_1}}, \mathbf{v_{e_2}})$ are sent through a Siamese summarizer and classifier to learn a good representation utilizing the synthetically created match/non-match pairs and labels. The Siamese summarizer uses the same model parameters to generate the representations for $e_1$ and $e_2$. The third DeepBlocker variant is called *hybrid* because it combines the Autoencoder and CTT approach by first training the Autoencoder and then using the representations generated by the Autoencoder as input for the Siamese summarizer and classifier of the CTT model.

**Embedding Block Builders** Creating entity embeddings does not give us blocks yet. We present two variants for creating blocks from embeddings. We start with $k$ nearest neighbor search and then illustrate another method using clustering. Given an entity $e_a \in \mathcal{E}_a$ of the Knowledge Graph $\mathcal{KG}_a$ with the embedding $\mathbf{v_{e_a}}$ and the entity embeddings $\mathbb{E}_b = \left[\mathbf{v_{e_{b_1}}}, ..., \mathbf{v_{e_{b_m}}}\right]$ of the entities

$\{e_{b_1}, ..., e_{b_m}\} \in \mathcal{E}_b$ belonging to $\mathcal{KG}_b$, we create the block $b_a = \{e_a, kNN(\mathbf{v_{e_{a_i}}}, \mathbb{E}_b)\}$. Where $kNN(\mathbf{v_{e_{a_i}}}, \mathbb{E}_b)$ returns the $k$ nearest neighbors of $\mathbf{v_{e_{a_i}}}$ among $\mathbb{E}_b$ w.r.t a specific distance measure (e.g. euclidean distance). We create a set of $n$ blocks $\mathcal{B} = \{b_1, ..., b_n\}$ for all $\{e_{a_1}, ..., e_{a_n}\} \in \mathcal{E}_a$. The number of comparisons in $\mathcal{B}$ is therefore $k|\mathcal{E}_a|$. To improve the speed, we use Faiss as the nearest neighbor library [15].

Alternatively, we can use a clustering function that assigns a cluster label $\omega_i \in \Omega$ to each embedding $\mathbf{v_e} \in \{\mathbb{E}_a, \mathbb{E}_b\}$. Then we can create a set of blocks $\mathcal{B} = \{b_1, ..., b_n\}$ for all $\{\omega_1, ..., \omega_n\} \in \Omega$, with all entities in $b_i$ having the same cluster label $\omega_i$ for their embedding.

## 4.3 Generalized Relational Blocking

The most straightforward approach for relational blocking uses token blocking [25] for $\Theta_A$ and $\Theta_R$. For each tuple $(e, l_c) \in \mathcal{C}$ with $\mathcal{C} \in \{\mathcal{C}_a, \mathcal{C}_r\}$, the concatenated attribute values $l_c$ are tokenized and entities that share a token $t$ are put into the same block:

$$\forall e_1, e_2 \in b_i : b_i \in \mathcal{B}_{tok} \wedge tokens(l_{c_1}) \cap tokens(l_{c_2}) \neq \emptyset \tag{4}$$

Here $tokens$ is the tokenization function, and $\mathcal{B}_{tok}$ is the set of generated blocks. Since we do not use the unique name blocking of MinoanER [7], this blocking scheme has at least the same recall but at most the same reduction ratio as theirs. We will refer to this blocking scheme as **RelTB**.

*Hybrid Block Building via Embedding Clustering* To investigate different granularities for the embedding block builder's clustering variant, we use token or attribute embeddings as inputs for the clustering method. For example, for the token embeddings clustering approach, we take the concatenated literal values $l_{c_i}$ of an entity $e_i$ and use $token(l_{c_i})$ as input for an embedding function. This provides us with multiple embeddings $\{\mathbf{v_{e_{i_1}}}, ..., \mathbf{v_{e_{i_n}}}\}$ per entity $e_i$ if the number of tokens is $n$. This can analogously be done by using the embeddings of literal values. In this case, the number of embeddings for each entity $e_i$ is equal to the number of attribute triples $(e_i, a, l_j) \in \mathcal{T}_A$. Clustering methods like HDBSCAN [17] can cluster specific data points as noise. In the domain of blocking, we see this as a hint that there is insufficient semantic overlap to create a block. Instead, we perform token blocking for all token/attribute embeddings assigned to the noise cluster. We use the other cluster labels as described in Section 4.2. We will refer to the token/attribute embedding clustering-based variant as $\mathbf{RelTB}_{TC}$ or $\mathbf{RelTB}_{AC}$ respectively.

*Enhanced DeepBlocker* Approaches that rely on DeepBlocker for $\Theta_A$ and $\Theta_R$ will be denoted as **RelDeepBlocker**. Furthermore, we also incorporate hybrid approaches, which use token blocking for $\Theta_A$ but use DeepBlocker as Frame Encoder and $k$ nearest neighbor search as embedding block builder for $\Theta_R$. We will denote these methods as $\mathbf{RelTB}_{DeepBlocker}$. In a preliminary study, we investigated whether clustering could also be used with these embeddings, but HDBSCAN assigned most of the entities to the noise cluster.

Table 1: Dataset statistics

| Task Name | Dataset | $\|\mathcal{E}\|$ | $\|\mathcal{T}_R\|$ | $\|\mathcal{T}_A\|$ | $\|\mathcal{R}\|$ | $\|\mathcal{A}\|$ | $\|\mathcal{L}\|$ | $\|\mathcal{M}\|$ |
|---|---|---|---|---|---|---|---|---|
| S-DW1 | DBpedia | 15000 | 38265 | 52134 | 248 | 341 | 28236 | 15000 |
| | Wikidata | 15000 | 42746 | 138246 | 169 | 649 | 118515 | 15000 |
| S-DW2 | DBpedia | 15000 | 73983 | 51378 | 167 | 174 | 25690 | 15000 |
| | Wikidata | 15000 | 83365 | 175686 | 121 | 457 | 146977 | 15000 |
| S-DY1 | DBpedia | 15000 | 30291 | 52093 | 165 | 256 | 25297 | 15000 |
| | YAGO | 15000 | 26638 | 117114 | 28 | 34 | 105710 | 15000 |
| S-DY2 | DBpedia | 15000 | 68063 | 49602 | 72 | 89 | 22561 | 15000 |
| | YAGO | 15000 | 60970 | 116151 | 21 | 19 | 104546 | 15000 |
| L-DW1 | DBpedia | 100000 | 293990 | 334911 | 413 | 492 | 133931 | 100000 |
| | Wikidata | 100000 | 251708 | 687860 | 261 | 874 | 542921 | 100000 |
| L-DW2 | DBpedia | 100000 | 616457 | 360696 | 318 | 327 | 137483 | 100000 |
| | Wikidata | 100000 | 588203 | 878219 | 239 | 760 | 682367 | 100000 |
| L-DY1 | DBpedia | 100000 | 294188 | 360415 | 287 | 378 | 101386 | 100000 |
| | YAGO | 100000 | 400518 | 649787 | 32 | 37 | 497633 | 100000 |
| L-DY2 | DBpedia | 100000 | 576547 | 374785 | 230 | 276 | 97433 | 100000 |
| | YAGO | 100000 | 865265 | 755161 | 31 | 35 | 578596 | 100000 |
| S-ED1 | DBpedia-EN | 15000 | 47676 | 62403 | 215 | 285 | 28973 | 15000 |
| | DBpedia-DE | 15000 | 50419 | 133776 | 131 | 193 | 35630 | 15000 |
| S-ED2 | DBpedia-EN | 15000 | 84867 | 59511 | 169 | 170 | 23831 | 15000 |
| | DBpedia-DE | 15000 | 92632 | 161315 | 96 | 115 | 33185 | 15000 |
| S-EF1 | DBpedia-EN | 15000 | 47334 | 57164 | 267 | 307 | 30281 | 15000 |
| | DBpedia-FR | 15000 | 40864 | 54401 | 210 | 403 | 28760 | 15000 |
| S-EF2 | DBpedia-EN | 15000 | 96318 | 52396 | 193 | 188 | 22761 | 15000 |
| | DBpedia-FR | 15000 | 80112 | 56114 | 166 | 220 | 21645 | 15000 |
| L-ED1 | DBpedia-EN | 100000 | 335359 | 423666 | 381 | 450 | 147142 | 100000 |
| | DBpedia-DE | 100000 | 336240 | 586207 | 196 | 251 | 199527 | 100000 |
| L-ED2 | DBpedia-EN | 100000 | 622588 | 430752 | 323 | 325 | 139867 | 100000 |
| | DBpedia-DE | 100000 | 629395 | 656458 | 170 | 188 | 200356 | 100000 |
| L-EF1 | DBpedia-EN | 100000 | 309607 | 384248 | 400 | 465 | 145103 | 100000 |
| | DBpedia-FR | 100000 | 258285 | 340725 | 300 | 518 | 157791 | 100000 |
| L-EF2 | DBpedia-EN | 100000 | 649902 | 396150 | 379 | 363 | 145382 | 100000 |
| | DBpedia-FR | 100000 | 561391 | 342768 | 287 | 467 | 157564 | 100000 |

Table 2: Reduction Ratio (RR) and Recall (Rec) results in percent. The RR values of the respective variants are shown in a single column, since they are identical by design. Per Task the RR and Rec values for the three highest h3r values are colored with a darker color implying a better value. All results shown use SIF embeddings.

| | DeepBlocker | | | | RelDeepBlocker | | | | RelTB$_{DeepBlocker}$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | AE | CTT | hyb | | AE | CTT | hyb | | AE | CTT | hyb |
| | RR | Rec | Rec | Rec | RR | Rec | Rec | Rec | RR | Rec | Rec | Rec |
| S-DW1 | 99.904 | 8.6 | 16.9 | 9.8 | 99.902 | 28.0 | 34.9 | 14.3 | 99.914 | 45.4 | 48.5 | 46.0 |
| S-DW2 | 99.920 | 14.8 | 14.6 | 9.2 | 99.840 | 27.6 | 33.0 | 12.6 | 99.918 | 47.5 | 47.8 | 39.9 |
| S-DY1 | 99.885 | 26.3 | 19.6 | 12.4 | 99.769 | 52.9 | 33.8 | 17.8 | 99.922 | 70.3 | 66.8 | 60.3 |
| S-DY2 | 99.873 | 21.8 | 13.8 | 6.7 | 99.872 | 49.5 | 37.8 | 18.8 | 99.905 | 46.4 | 55.7 | 52.5 |
| S-ED1 | 99.914 | 18.4 | 17.1 | 13.1 | 99.914 | 28.3 | 25.6 | 12.3 | 99.877 | 77.7 | 76.6 | 72.5 |
| S-ED2 | 99.924 | 13.1 | 14.6 | 13.0 | 99.923 | 14.3 | 22.9 | 11.4 | 99.869 | 73.6 | 73.4 | 71.3 |
| S-EF1 | 99.782 | 27.3 | 26.6 | 21.8 | 99.777 | 36.2 | 34.4 | 24.8 | 99.846 | 65.5 | 69.7 | 67.7 |
| S-EF2 | 99.759 | 24.8 | 31.3 | 29.7 | 99.753 | 23.5 | 43.4 | 27.1 | 99.825 | 83.6 | 82.2 | 79.6 |

## 5 Experiments

### 5.1 Datasets

Several benchmark datasets were considered for this study but ultimately were not deemed the right fit. The movie datasets used by Obraczka et al. [23] have a shallow graph structure but are relatively small and do not necessarily require blocking. The datasets used in the KG track of the Ontology Alignment Evaluation Initiative [1] contain millions of entities, making them challenging in terms of scalability and therefore a perfect candidate for our experiments. However, we found that Token Blocking (without involving relations) yields 100% recall on the available partial gold standard, making it unsuitable for this investigation. The datasets used by Efthymiou et al. [7] are also sufficiently large. Still, we could not load them properly due to inconsistent encodings across the gold standard and respective data sources, and we found some entities in the gold standard that do not show up in either data source. Ultimately, we chose the 16 matching tasks published by Sun et al. [34], which consist of matching samples of the KGs DBpedia, Wikidata, and YAGO. Half of the matching tasks consist of a multi-lingual setting, which aim to match entities from different DBpedia variants (English-German and English-French). The similar but larger DBP1M dataset [10] is unsuitable for our study because it does not contain attribute triples. Table 1 shows the statistics of the matching tasks. With up to 2.3 million triples, these datasets are challenging w.r.t scalability, but they still have problems. They rely on an unrealistic 1-to-1 matching scenario, where each dataset contains a respective counterpart for the other dataset. Furthermore, all the datasets are derived from Wikipedia in one way or the other, providing limited heterogeneity. More research is needed to create realistic benchmark datasets, but this is outside the scope of this study.

Table 3: Reduction Ratio (RR) and Recall (Rec) results in percent. The RR values of the RelTB$_{DeepBlocker}$ variants are shown in a single column, since they are identical by design. Per Task the RR and Rec values for the three highest h3r values are colored with a darker color implying a better value. Runs that did not complete after 10 hours are marked with '-'.

| | | LightEA | | RelTB$_{DeepBlocker}$ | | | | RelTB$_{AC}$ | | RelTB$_{TC}$ | | RelTB | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | AE | CTT | hyb | | | | | | |
| | EM | RR | Rec | RR | Rec | Rec | Rec | RR | Rec | RR | Rec | RR | Rec |
| S-DW1 | SIF | 99.896 | 29.9 | 99.914 | 45.4 | 48.5 | 46.0 | 97.631 | 91.1 | 95.796 | 96.5 | 98.886 | 90.1 |
| | ST | 99.896 | 33.8 | 99.914 | 49.3 | 42.6 | 47.4 | 99.414 | 89.8 | 99.014 | 90.5 | | |
| S-DW2 | SIF | 99.917 | 23.9 | 99.918 | 47.5 | 47.8 | 39.9 | 96.397 | 99.9 | 94.866 | 100.0 | 97.649 | 97.5 |
| | ST | 99.917 | 29.2 | 99.918 | 41.6 | 39.8 | 43.5 | 98.619 | 97.6 | 97.193 | 98.1 | | |
| S-DY1 | SIF | 99.877 | 55.2 | 99.922 | 70.3 | 66.8 | 60.3 | 99.759 | 94.9 | 96.118 | 98.4 | 99.625 | 95.4 |
| | ST | 99.877 | 36.8 | 99.922 | 67.2 | 59.7 | 56.8 | 99.835 | 95.1 | 99.617 | 96.1 | | |
| S-DY2 | SIF | 99.870 | 35.4 | 99.905 | 46.4 | 55.7 | 52.5 | 99.561 | 99.0 | 93.730 | 100.0 | 98.209 | 99.1 |
| | ST | 99.870 | 28.0 | 99.905 | 46.6 | 46.9 | 46.4 | 99.193 | 99.1 | 95.008 | 99.1 | | |
| L-DW1 | SIF | 99.957 | 26.1 | 99.934 | 43.8 | 43.7 | 43.1 | 95.973 | 92.8 | 95.316 | 94.1 | 98.815 | 84.2 |
| | ST | 99.957 | 29.6 | 99.934 | 46.1 | 44.9 | 46.2 | 99.312 | 83.7 | 99.031 | 84.6 | | |
| L-DW2 | SIF | 99.968 | 20.4 | 99.929 | 44.3 | 45.2 | - | 98.245 | 97.3 | 91.984 | 99.8 | 95.486 | 97.6 |
| | ST | 99.968 | 26.2 | 99.929 | 44.3 | 46.6 | 46.0 | 97.502 | 97.5 | 96.101 | 97.6 | | |
| L-DY1 | SIF | 99.957 | 42.7 | 99.950 | 68.4 | 68.8 | - | 96.889 | 96.8 | 95.496 | 98.5 | 99.258 | 96.5 |
| | ST | 99.957 | 26.4 | 99.950 | 68.7 | 69.3 | 70.4 | 99.600 | 96.4 | 99.316 | 96.9 | | |
| L-DY2 | SIF | 99.965 | 38.1 | 99.938 | 74.9 | - | - | 96.878 | 99.3 | 93.387 | 99.9 | 96.201 | 99.5 |
| | ST | 99.965 | 24.0 | 99.938 | 74.3 | 73.8 | 74.1 | 98.008 | 99.4 | 96.851 | 99.3 | | |
| S-ED1 | SIF | 99.910 | 30.4 | 99.877 | 77.7 | 76.6 | 72.5 | 99.421 | 92.3 | 97.035 | 96.2 | 99.155 | 92.1 |
| | ST | 99.910 | 41.5 | 99.877 | 84.0 | 72.5 | 70.2 | 99.246 | 92.5 | 99.233 | 92.6 | | |
| S-ED2 | SIF | 99.922 | 26.4 | 99.869 | 73.6 | 73.4 | 71.3 | 98.822 | 95.9 | 96.682 | 99.0 | 98.453 | 95.9 |
| | ST | 99.922 | 34.7 | 99.869 | 80.1 | 71.4 | 69.6 | 98.383 | 96.3 | 98.421 | 96.4 | | |
| S-EF1 | SIF | 99.759 | 37.4 | 99.846 | 65.5 | 69.7 | 67.7 | 99.196 | 84.6 | 93.686 | 88.5 | 98.947 | 84.2 |
| | ST | 99.759 | 63.8 | 99.846 | 76.3 | 71.0 | 66.0 | 98.758 | 85.0 | 98.664 | 84.5 | | |
| S-EF2 | SIF | 99.745 | 40.5 | 99.825 | 83.6 | 82.2 | 79.6 | 96.935 | 98.2 | 86.246 | 98.8 | 96.934 | 96.9 |
| | ST | 99.745 | 72.4 | 99.825 | 87.1 | 80.2 | 76.0 | 95.012 | 98.1 | 95.199 | 97.0 | | |
| L-ED1 | SIF | 99.960 | 20.6 | 99.925 | 66.5 | 66.4 | 67.1 | 99.531 | 89.3 | 95.902 | 96.2 | 99.323 | 89.3 |
| | ST | 99.960 | 38.8 | 99.925 | 66.3 | 68.4 | 68.6 | 99.274 | 89.7 | 99.331 | 89.8 | | |
| L-ED2 | SIF | 99.965 | 22.8 | 99.907 | 63.6 | 63.6 | 63.6 | 98.289 | 92.0 | 98.071 | 92.6 | 98.475 | 92.0 |
| | ST | 99.965 | 33.4 | 99.907 | 63.4 | 65.1 | 64.9 | 98.277 | 92.4 | 98.353 | 92.6 | | |
| L-EF1 | SIF | 99.924 | 22.2 | 99.939 | 59.5 | 59.8 | 59.8 | 99.027 | 80.4 | 98.611 | 81.1 | 99.082 | 80.5 |
| | ST | 99.924 | 45.5 | 99.939 | 59.1 | 63.0 | 62.5 | 99.247 | 80.6 | 99.136 | 80.7 | | |
| L-EF2 | SIF | 99.926 | 25.8 | 99.928 | 68.4 | 68.5 | 68.4 | 96.861 | 95.6 | 93.910 | 95.8 | 95.515 | 95.5 |
| | ST | 99.926 | 53.1 | 99.928 | 67.9 | 70.8 | 70.6 | 96.876 | 96.0 | 94.740 | 95.9 | | |

### 5.2 Setup

We tuned the hyperparameters[4] on S-DW1 where necessary and used these values for all other datasets. Using all attributes was the best setting for the top $top_{n_a}$ attributes. Still, for the $top_{n_r}$ relations, we used the 90th percentile of the number of distinct relations as the cutoff point to avoid highly connected nodes gathering a vast neighborhood. For all approaches, we cleaned the literals from XSD datatype information. For Token Blocking, we removed stopwords[5] and tokens shorter than character length 3. Approaches that utilize nearest neighbor search use $k = 500$ on the small datasets and $k = 1000$ on the large datasets. For RelDeepBlocker, the $\Theta_A$ and $\Theta_R$ components use half the $k$ to ensure comparability. Our experiments include a comparison to LightEA, a state-of-the-art *unsupervised* KGE method, as an embedding method that incorporates the rich relational information present in the KG in a more elaborate manner. This approach was chosen for its speed and high-quality results [16]. As previously mentioned, we investigate two different embedding approaches: SIF aggregated token embeddings, which rely on fasttext word embeddings, and Sentence-BERT embeddings (ST), where we use GTR-T5 [20] for the mono-lingual datasets and LaBSE [9] for the multi-lingual tasks. On the large datasets, we reduce the dimensionality to fit the embeddings in one GPU. For the Sentence-BERT embeddings, we perform PCA on 30% of the data and use the first principal component as the final layer[6] of the Sentence-BERT model to reduce to dimensionality from 768 to 196 dimensions. The fasttext embeddings can be reduced using the dimensionality reduction provided by the fasttext library. Here, we reduce the dimensionality from 300 to 100 dimensions. For RelTB$_{AC}$ and RelTB$_{TC}$, we further reduce the dimensionality since HDBSCAN works best on lower dimensional embeddings. Here, we reduce the Sentence-BERT embeddings to 32, and the fasttext embeddings via UMAP [18] to 25 dimensions. In the future, we want to investigate whether hubness-reduction methods are viable alternatives here [22]. We use the RAPIDS[7] library for faster GPU implementations of HDBSCAN and UMAP. The experiments were run on a machine with an AMD EPYC 7551P CPU. For the small datasets, we used a GeForce RTX 2080 Ti GPU; for the large datasets, we used one Tesla V100. More detailed information on reproducibility can be found in our GitHub repository.

### 5.3 Results

We rely on the Bayesian analysis for comparing machine learning models proposed by Benavoli et al. [3] to make sound statistical performance comparisons. A Bayesian signed rank test [4] determines significant differences between the

[4] Detailled information can be found here: `https://github.com/dobraczka/klinker/tree/main/run_scripts/hyperparam_sweeps`

[5] Using NLTK's [5] English stopword list

[6] see `https://www.sbert.net/examples/training/distillation/README.html#dimensionality-reduction`

[7] `https://rapids.ai/`

two approaches. Compared to frequentist hypothesis testing, this allows rejecting *or* accepting a null hypothesis. Furthermore, a *region of practical equivalence* (ROPE) can be defined for approaches with equally good performance. We rely on the `Autorank` [11] package to automatically set the ROPE in relation to effect size. The Bayesian analysis gives us a probability that one approach is better, worse, or equal to another. In our study, we decide if one of these probabilities is $\geq 95\%$, or else we give no verdict (i.e., see it as inconclusive).

We first investigate the performance of the different approaches that utilize DeepBlocker. Table 2 shows Recall and Reduction Ratio values on the small datasets using the SIF embeddings. Since the number of neighbors controls the reduction ratio, we show these values in a single column for each family of approaches. For space reasons, the other datasets and embedding variants are omitted here. Per matching task, the three best approaches w.r.t h3r values are highlighted in decreasing strength of color. It is evident that the RelTB$_{DeepBlocker}$ variants outperform their counterparts, and using our statistical comparison regime, we can say that this difference is significant. We can also see that RelDeepBlocker is better than DeepBlocker, except for the hyb variant, which generally performs the worst. For AE and CTT, the RelDeepBlocker variant is also significantly better than its non-relational counterpart.

In Table 3, we show the results on all datasets with both embedding variants. Per matching task, the three best approaches w.r.t h3r values are again highlighted in decreasing strength of color. We can see that RelTB and the attribute/token embedding clustering variants RelTB$_{AC}$ and RelTB$_{TC}$ perform the best. In fact, on all 16 matching tasks, the best value is achieved either by RelTB$_{AC}$ or RelTB$_{TC}$. Looking at the RR and Recall values, we can see that RelTB$_{AC}$ generally has a higher RR than RelTB$_{TC}$, with the latter dominating in Recall, even achieving 100% on two datasets (S-DW2 and S-DY2). The KGE-based approach LightEA performs the worst, not managing to outperform RelTB$_{DeepBlocker}$ on a single dataset. In Figure 2, we provide the result of the Bayesian statistical analysis w.r.t h3r values. We also distinguished the different embedding approaches to provide a more detailed analysis. All other approaches outperform LightEA. It is also notable that RelToken and the RelTB variants without DeepBlocker are significantly better than the other approaches. Among these five dominant approaches, we cannot reach a conclusive answer as to which of these performs the best. In Figure 3, we take a more detailed analysis w.r.t Recall and Reduction ratio on these five approaches. While RelTB$_{TC}$ with SIF embeddings is significantly better regarding Recall than all other approaches, it is also significantly worse regarding Reduction Ratio. RelTB$_{AC}$ with SentenceTransformer embeddings strikes the best balance between Recall and Reduction Ratio of these approaches. It performs similarly to, e.g., RelToken regarding Recall but is significantly better in terms of Reduction Ratio. While it is inconclusive whether RelTB$_{AC}$ with SentenceTransformer embeddings is significantly better than RelToken w.r.t h3r values, we see this nuanced analysis on Recall and Reduction Ratio as a sign that hybrid approaches deserve more research attention.

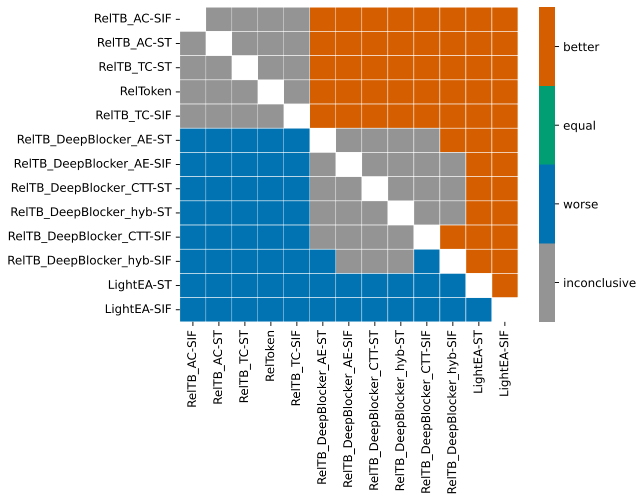

Fig. 2: Decision matrix comparing blocking approaches using Bayesian signed rank tests w.r.t h3r values. Each cell shows the decision when comparing the row approach to the column approach. A decision is reached if the posterior probability is $\geq 95\%$.

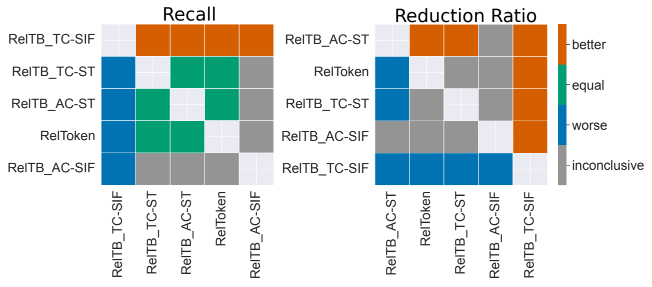

Fig. 3: Decision matrices comparing a subset of blocking approaches using Bayesian signed rank tests w.r.t Recall and Reduction Ratio. Each cell shows the decision when comparing the row approach to the column approach. A decision is reached if the posterior probability is $\geq 95\%$.

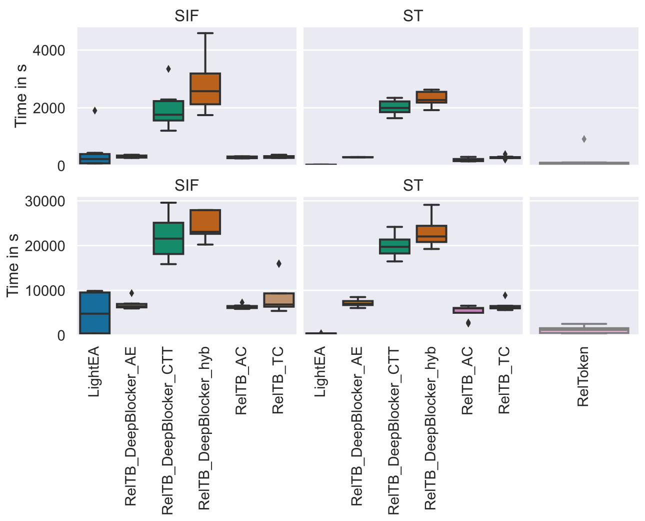

Fig. 4: Time in seconds as aggregated box plots, with the first row showing the 15k datasets and the second row showing the 100k datasets. Note the differing ranges of the y-axis per row.

Since the main goal of employing blocking is to speed up the matching process, runtime is an important factor. In Figure 4, we show the runtimes of the different approaches. The purely symbolic approach RelToken is the fastest of the high-quality methods. RelTB$_{TC}$ and RelTB$_{AC}$ are roughly on par with the AutoEncoder variant of RelTB$_{DeepBlocker}$. The more sophisticated DeepBlocker variants are one order of magnitude slower. While LightEA is the fastest (especially when using Sentence-BERT embeddings), it does not produce high-quality results.

## 5.4 Impact of relational enhancement

Since one of our contributions is enhancing some state-of-the-art blocking approaches for the KG domain, we will now examine the differences between relational and non-relational blocking. In Figure 5, we compare relational blockers and their non-relational counterparts on S-DY2. Since LightEA is relational by design, we compare it with the SIF aggregated fasttext embeddings. For the RelTB$_{DeepBlocker}$ variants, we compare them with their DeepBlocker counterparts. In terms of Recall, we can, in many cases, more than double the values while retaining a high Reduction Ratio. The highest cost of relational enhancement is in terms of speed. For example, Token Blocking without relational enhancement takes 8 seconds, while RelToken takes 27. The speed penalty is higher for the attribute/token embedding clustering variants, where the additional steps of embedding and clustering take most of the time and up to 313 seconds. The dimensionality reduction needed for HDBSCAN is included in this time as well.

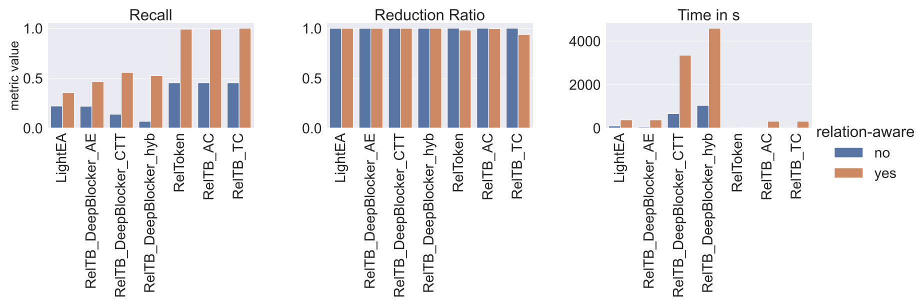

Fig. 5: Comparison between relational blockers with their non-relational counterparts on S-DY2, with SIF aggregated fasttext embeddings (where applicable).

## 6 Conclusion and Future Work

In this work, we presented a unified framework that integrates embedding-based and symbolic approaches, enhances state-of-the-art methods to utilize relational information, and explores various hybrid methods that combine embedding-based and symbolic techniques. Our study showed that simple symbolic approaches can outperform sophisticated, state-of-the-art deep learning methods like DeepBlocker, even when adapted to incorporate neighborhood information. We showed that Relational Token Blocking and the hybrid variants that utilize clustering on token/attribute embeddings are significantly better than the other approaches we investigated. Token Blocking relies on symbolic overlap at a lower level of granularity than entity embeddings. Our study has shown that similarities at this level can be exploited by embedding-based methods as well. Subsequent studies should investigate how these findings can be incorporated into new hybrid methods, which are currently an underexplored field in (relational) blocking research. Future work must address the current limitations of existing benchmark datasets to enable more robust investigations. Even on some of the datasets used in this study, no approach managed blocking quality that would be deemed acceptable in a practical setting (e.g., on L-EF1 we achieved at best 81% Recall), emphasizing the importance of additional research.

**Acknowledgments.** The authors acknowledge the financial support by the BMBF and by the SMWK in the programme Center of Excellence for AI-research "Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig", project identification number: ScaDS.AI. The authors gratefully acknowledge the computing time made available to them on the high-performance computer at the NHR Center of TU Dresden. This center is jointly supported by the Federal Ministry of Education and Research and the state governments participating in the NHR (www.nhr-verein.de/unsere-partner). Computations for this work were done (in part) using resources of the Leipzig University Computing Center.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Algergawy, A., Faria, D., Ferrara, A., Fundulaki, I., Harrow, I., Hertling, S., Jiménez-Ruiz, E., Karam, N., Khiat, A., Lambrix, P., Li, H., Montanelli, S., Paulheim, H., Pesquita, C., Saveta, T., Shvaiko, P., Splendiani, A., Thiéblin, É., Trojahn, C., Vatascinová, J., Zamazal, O., Zhou, L.: Results of the ontology alignment evaluation initiative 2019. In: Shvaiko, P., Euzenat, J., Jiménez-Ruiz, E., Hassanzadeh, O., Trojahn, C. (eds.) Proceedings of the 14th International Workshop on Ontology Matching co-located with the 18th International Semantic Web Conference (ISWC 2019), Auckland, New Zealand, October 26, 2019. CEUR Workshop Proceedings, vol. 2536, pp. 46–85. CEUR-WS.org (2019), https://ceur-ws.org/Vol-2536/oaei19_paper0.pdf
2. Arora, S., Liang, Y., Ma, T.: A Simple but Tough-to-Beat Baseline for Sentence Embeddings. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), https://openreview.net/forum?id=SyK00v5xx
3. Benavoli, A., Corani, G., Demsar, J., Zaffalon, M.: Time for a Change: a Tutorial for Comparing Multiple Classifiers Through Bayesian Analysis. J. Mach. Learn. Res. **18**, 77:1–77:36 (2017), http://jmlr.org/papers/v18/16-305.html
4. Benavoli, A., Corani, G., Mangili, F., Zaffalon, M., Ruggeri, F.: A Bayesian Wilcoxon signed-rank test based on the Dirichlet process. In: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014. JMLR Workshop and Conference Proceedings, vol. 32, pp. 1026–1034. JMLR.org (2014), http://proceedings.mlr.press/v32/benavoli14.html
5. Bird, S., Klein, E., Loper, E.: Natural Language Processing with Python. O'Reilly (2009), http://www.oreilly.de/catalog/9780596516499/index.html
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/V1/N19-1423, https://doi.org/10.18653/v1/n19-1423
7. Efthymiou, V., Papadakis, G., Stefanidis, K., Christophides, V.: Minoaner: Schema-agnostic, non-iterative, massively parallel resolution of web entities. In: Herschel, M., Galhardas, H., Reinwald, B., Fundulaki, I., Binnig, C., Kaoudi, Z. (eds.) Advances in Database Technology - 22nd International Conference on Extending Database Technology, EDBT 2019, Lisbon, Portugal, March 26-29, 2019. pp. 373–384. OpenProceedings.org (2019). https://doi.org/10.5441/002/EDBT.2019.33, https://doi.org/10.5441/002/edbt.2019.33
8. Efthymiou, V., Stefanidis, K., Christophides, V.: Benchmarking Blocking Algorithms for Web Entities. IEEE Transactions on Big Data **6**(2), 382–395 (Jun 2020). https://doi.org/10.1109/TBDATA.2016.2576463, https://ieeexplore.ieee.org/document/7485873/
9. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT Sentence Embedding. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 878–891. Association for Computational Linguistics, Dublin, Ireland (May 2022). https://doi.org/10.18653/v1/2022.acl-long.62, https://aclanthology.org/2022.acl-long.62

10. Ge, C., Liu, X., Chen, L., Zheng, B., Gao, Y.: LargeEA: Aligning Entities for Large-scale Knowledge Graphs. Proc. VLDB Endow. **15**(2), 237–245 (2021). https://doi.org/10.14778/3489496.3489504, http://www.vldb.org/pvldb/vol15/p237-gao.pdf
11. Herbold, S.: Autorank: A Python package for automated ranking of classifiers. J. Open Source Softw. **5**(48), 2173 (2020). https://doi.org/10.21105/JOSS.02173, https://doi.org/10.21105/joss.02173
12. Hertling, S., Portisch, J., Paulheim, H.: KERMIT - A Transformer-Based Approach for Knowledge Graph Matching. CoRR **abs/2204.13931** (2022). https://doi.org/10.48550/ARXIV.2204.13931, https://doi.org/10.48550/arXiv.2204.13931, arXiv: 2204.13931
13. Hofer, M., Obraczka, D., Saeedi, A., Köpcke, H., Rahm, E.: Construction of knowledge graphs: Current state and challenges. Inf. **15**(8), 509 (2024). https://doi.org/10.3390/INFO15080509, https://doi.org/10.3390/info15080509
14. Isele, R., Jentzsch, A., Bizer, C.: Efficient multidimensional blocking for link discovery without losing recall. In: Marian, A., Vassalos, V. (eds.) Proceedings of the 14th International Workshop on the Web and Databases 2011, WebDB 2011, Athens, Greece, June 12, 2011 (2011), http://webdb2011.rutgers.edu/papers/Paper%2039/silk.pdf
15. Johnson, J., Douze, M., Jégou, H.: Billion-Scale Similarity Search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2021). https://doi.org/10.1109/TBDATA.2019.2921572
16. Mao, X., Wang, W., Wu, Y., Lan, M.: LightEA: A Scalable, Robust, and Interpretable Entity Alignment Framework via Three-view Label Propagation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 825–838. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (2022). https://doi.org/10.18653/v1/2022.emnlp-main.52, https://aclanthology.org/2022.emnlp-main.52
17. McInnes, L., Healy, J.: Accelerated hierarchical density based clustering. In: Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. pp. 33–42. IEEE (2017)
18. McInnes, L., Healy, J.: UMAP: uniform manifold approximation and projection for dimension reduction. CoRR **abs/1802.03426** (2018), http://arxiv.org/abs/1802.03426
19. Nentwig, M., Hartung, M., Ngomo, A.C.N., Rahm, E.: A survey of current Link Discovery frameworks. Semantic Web **8**(3), 419–436 (2017). https://doi.org/10.3233/SW-150210
20. Ni, J., Qu, C., Lu, J., Dai, Z., Hernandez Abrego, G., Ma, J., Zhao, V., Luan, Y., Hall, K., Chang, M.W., Yang, Y.: Large Dual Encoders Are Generalizable Retrievers. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 9844–9855. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Dec 2022). https://doi.org/10.18653/v1/2022.emnlp-main.669, https://aclanthology.org/2022.emnlp-main.669
21. Obraczka, D., Ngomo, A.C.N.: Dragon: Decision Tree Learning for Link Discovery. In: Bakaev, M., Frasincar, F., Ko, I.Y. (eds.) Web Engineering - 19th International Conference, ICWE 2019, Daejeon, South Korea, June 11-14, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11496, pp. 441–456. Springer (2019). https://doi.org/10.1007/978-3-030-19274-7_31, https://doi.org/10.1007/978-3-030-19274-7_31

22. Obraczka, D., Rahm, E.: Fast Hubness-Reduced Nearest Neighbor Search for Entity Alignment in Knowledge Graphs. SN Comput. Sci. **3**(6), 501 (2022). https://doi.org/10.1007/S42979-022-01417-1, https://doi.org/10.1007/s42979-022-01417-1
23. Obraczka, D., Schuchart, J., Rahm, E.: Embedding-assisted entity resolution for knowledge graphs. In: Chaves-Fraga, D., Dimou, A., Heyvaert, P., Priyatna, F., Sequeda, J.F. (eds.) Proceedings of the 2nd International Workshop on Knowledge Graph Construction co-located with 18th Extended Semantic Web Conference (ESWC 2021), Online, June 6, 2021. CEUR Workshop Proceedings, vol. 2873. CEUR-WS.org (2021), https://ceur-ws.org/Vol-2873/paper8.pdf
24. Papadakis, G., Fisichella, M., Schoger, F., Mandilaras, G., Augsten, N., Nejdl, W.: Benchmarking Filtering Techniques for Entity Resolution. 2023 IEEE 39th International Conference on Data Engineering (ICDE) pp. 653–666 (Apr 2023). https://doi.org/10.1109/ICDE55515.2023.00389, https://ieeexplore.ieee.org/document/10184692/, conference Name: 2023 IEEE 39th International Conference on Data Engineering (ICDE) ISBN: 9798350322279 Place: Anaheim, CA, USA Publisher: IEEE
25. Papadakis, G., Ioannou, E., Niederée, C., Fankhauser, P.: Efficient entity resolution for large heterogeneous information spaces. In: King, I., Nejdl, W., Li, H. (eds.) Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011. pp. 535–544. ACM (2011). https://doi.org/10.1145/1935826.1935903
26. Papadakis, G., Ioannou, E., Niederée, C., Palpanas, T., Nejdl, W.: Beyond 100 million entities: large-scale blocking-based resolution for heterogeneous data. In: Adar, E., Teevan, J., Agichtein, E., Maarek, Y. (eds.) Proceedings of the Fifth International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, February 8-12, 2012. pp. 53–62. ACM (2012). https://doi.org/10.1145/2124295.2124305
27. Papadakis, G., Ioannou, E., Palpanas, T., Niederee, C., Nejdl, W.: A Blocking Framework for Entity Resolution in Highly Heterogeneous Information Spaces. IEEE Transactions on Knowledge and Data Engineering **25**(12), 2665–2682 (Dec 2013). https://doi.org/10.1109/TKDE.2012.150, http://ieeexplore.ieee.org/document/6255742/
28. Papadakis, G., Skoutas, D., Thanos, E., Palpanas, T.: Blocking and filtering techniques for entity resolution: A survey. ACM Comput. Surv. **53**(2) (mar 2020). https://doi.org/10.1145/3377455, https://doi.org/10.1145/3377455
29. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. pp. 3980–3990. Association for Computational Linguistics (2019). https://doi.org/10.18653/V1/D19-1410, https://doi.org/10.18653/v1/D19-1410
30. Saeedi, A., David, L., Rahm, E.: Matching Entities from Multiple Sources with Hierarchical Agglomerative Clustering. In: Aveiro, D., Dietz, J.L.G., Filipe, J. (eds.) Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2021, Volume 2: KEOD, Online Streaming, October 25-27, 2021. pp. 40–50. SCITEPRESS (2021). https://doi.org/10.5220/0010649600003064

31. Saeedi, A., Peukert, E., Rahm, E.: Using Link Features for Entity Clustering in Knowledge Graphs. In: Gangemi, A., Navigli, R., Vidal, M.E., Hitzler, P., Troncy, R., Hollink, L., Tordai, A., Alam, M. (eds.) The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings. Lecture Notes in Computer Science, vol. 10843, pp. 576–592. Springer (2018). https://doi.org/10.1007/978-3-319-93417-4_37, https://doi.org/10.1007/978-3-319-93417-4_37
32. Suchanek, F.M., Abiteboul, S., Senellart, P.: PARIS: probabilistic alignment of relations, instances, and schema. Proceedings of the VLDB Endowment **5**(3), 157–168 (Nov 2011). https://doi.org/10.14778/2078331.2078332, https://dl.acm.org/doi/10.14778/2078331.2078332
33. Sun, R., Cao, X., Zhao, Y., Wan, J., Zhou, K., Zhang, F., Wang, Z., Zheng, K.: Multi-modal Knowledge Graphs for Recommender Systems. In: d'Aquin, M., Dietze, S., Hauff, C., Curry, E., Cudré-Mauroux, P. (eds.) CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020. pp. 1405–1414. ACM (2020). https://doi.org/10.1145/3340531.3411947
34. Sun, Z., Zhang, Q., Hu, W., Wang, C., Chen, M., Akrami, F., Li, C.: A benchmarking study of embedding-based entity alignment for knowledge graphs. Proceedings of the VLDB Endowment **13**(12), 2326–2340 (Aug 2020). https://doi.org/10.14778/3407790.3407828, https://dl.acm.org/doi/10.14778/3407790.3407828
35. Thirumuruganathan, S., Li, H., Tang, N., Ouzzani, M., Govind, Y., Paulsen, D., Fung, G., Doan, A.: Deep learning for blocking in entity matching: a design space exploration. Proceedings of the VLDB Endowment **14**(11), 2459–2472 (Jul 2021). https://doi.org/10.14778/3476249.3476294, https://dl.acm.org/doi/10.14778/3476249.3476294
36. Usbeck, R., Röder, M., Hoffmann, M., Conrads, F., Huthmann, J., Ngomo, A.C.N., Demmler, C., Unger, C.: Benchmarking question answering systems. Semantic Web **10**(2), 293–304 (2019). https://doi.org/10.3233/SW-180312
37. Zeakis, A., Papadakis, G., Skoutas, D., Koubarakis, M.: Pre-trained embeddings for entity resolution: An experimental analysis. Proc. VLDB Endow. **16**(9), 2225–2238 (2023). https://doi.org/10.14778/3598581.3598594, https://www.vldb.org/pvldb/vol16/p2225-skoutas.pdf
38. Zhang, R., Trisedya, B.D., Li, M., Jiang, Y., Qi, J.: A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning. The VLDB Journal — The International Journal on Very Large Data Bases **31**(5), 1143–1168 (May 2022). https://doi.org/10.1007/s00778-022-00747-z, https://doi.org/10.1007/s00778-022-00747-z