Pre-trained Language Models for Entity Blocking: A Generalization Study

Runhui Wang* and **Yongfeng Zhang**
110 Frelinghuysen Rd, Piscataway, NJ 08854, USA
runhui.wang@rutgers.edu and yongfeng.zhang@rutgers.edu

Abstract

Entity Resolution (ER) is an essential task in data integration and its goal is to find records that represent the same entity in a dataset. Deep learning models, especially large pre-trained language models, have achieved state-of-the-art results on this task. A typical ER pipeline consists of Entity Blocking and Entity Matching: Entity Blocking finds candidate record pairs that potentially match and Entity Matching determines if the pairs match. The goal of the entity blocking step is to include as many matching pairs as possible while including as few non-matching pairs as possible. On the other hand, the blocking task can also be considered as an Information Retrieval (IR) task. However, state-of-the-art neural IR models that are based on large language models have not been evaluated on the ER task. What's more, the generalization ability of state-of-the-art methods for entity blocking is not well-studied but an import aspect in real-world applications. In this work, we evaluate state-of-the-art models for Entity Blocking along with neural IR models on a wide range of real-world datasets, and also study their in-distribution and out-of-distribution generalization abilities.

1 Introduction

Entity Resolution (ER) aims to find data instances that belong to the same entity, and is an essential problem in data integration (Getoor and Machanavajjhala, 2012; Konda et al., 2016; Rajaraman and Ullman, 2011). Over the decades of study, the focus has shifted from rule-based methods (Singh et al., 2017) and machine learning-based methods (Bilenko and Mooney, 2003; Konda et al., 2016), to deep learning-based methods, especially language model-based methods (Kasai et al., 2019; Li et al., 2021; Miao et al., 2021; Mudgal et al., 2018; Wang et al., 2022; Tu et al., 2022).

In various real-world applications, data instances representing entities, such as product profiles or news articles, are commonly represented as text items composed of words. A pair of text items is deemed a **match** if they correspond to the same real-world entity. One may compare every pair of items to solve the problem, but the computational cost of comparing each pair in a dataset to identify matches is prohibitively high, particularly for large datasets where the number of all pairs grows quadratically. To address this challenge, the entity resolution (ER) pipeline typically contains two primary stages: blocking and matching. The blocking stage aims to identify potential matches while excluding non-matching pairs, and the matching stage verifies whether candidate pairs, obtained through the blocking stage, indeed match. In the IR literature, this is a classic two-stage filter-and-refine pipeline, where the first stage (Entity Blocking) is a candidate generation phase.

In recent literature of ER, significant advances in accuracy have been achieved through the fine-tuning of Large Pre-trained Language Models (PLMs) (Thirumuruganathan et al., 2021; Wang et al., 2022; Peeters and Bizer, 2022; Li et al., 2021; Miao et al., 2021). Compared to previous methods, these PLM-based methods excel at capturing the semantics of text items and correlations of the words in the items. it has been demonstrated that PLM-based methods achieve state-of-the-art performance on both the blocking phase and the matching phase of the ER pipeline in (Thirumuruganathan et al., 2021; Miao et al., 2021; Li et al., 2021; Wang et al., 2022).

Within the literature, the blocking stage is typically composed of two steps: embedding and similarity search (Thirumuruganathan et al., 2021; Wang et al., 2022). In the embedding step, pre-trained language models are fine-tuned using either unsupervised or supervised learning approaches to encode entities into embeddings that encapsulate their semantic meaning. Subsequently, in the similarity search step, the search for similar item pairs

*Work done prior to joining Amazon.

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8717–8727

is conducted directly on the embeddings.

On the other hand, the entity blocking task can be viewed as an Information Retrieval (IR) task, which has seen significant advancements in recent IR literature with the emergence of large language models (PLMs). These PLMs play a crucial role in both dense and sparse neural ranking models, such as Contriever (Izacard et al., 2021), ColBERT (Khattab and Zaharia, 2020), and Splade (Formal et al., 2022). Dense models, such as Contriever (Izacard et al., 2021) and ColBERT (Khattab and Zaharia, 2020), leverage pre-trained language models and contrastive loss to learn dense representations for text items. Although these dense neural IR models are primarily fine-tuned for question answering tasks, they can be naturally applied in the entity blocking task. Sparse retrieval models, exemplified by Splade (Formal et al., 2022), predict the importance of tokens in text items and combine this importance with token embeddings generated by large language models for ranking purposes. These sparse neural IR models show potential for the entity blocking task. Since the performance of both dense and sparse LLM-based IR models has yet to be thoroughly investigated within the entity resolution literature, we also include them in our reproducibility study to better understand the advancement of both the IR and the ER literature.

In real-world applications of large-scale entity resolution (ER), labeled data is often only available for a small subset of the entire dataset or not available at all. What's more, it is not always feasible to use a large corpus of diverse real data during the training process, and real customer data is sometimes inaccessible during the training process due to various constraints such as privacy and security. Hence, it is crucial to assess the generalization capabilities of PLM-based methods, for the blocking task—an aspect largely unexplored in recent literature. Besides, commonly used datasets for evaluating entity resolution are not specifically designed for evaluating generalization.

In this paper, we examine the reproducibility and generalization ability of deep learning methods based on large language models (PLMs) for entity blocking. We conduct an evaluation and comparison of state-of-the-art methods from both the entity resolution and information retrieval literature as well as non-PLMs based methods, prepare new datasets and experimental settings that are specifically designed for evaluating generalization abilities of entity blocking methods.

2 Problem Definition

The Entity Resolution (ER) process frequently operates on two distinct tables A and B, each containing item profiles. The primary objective is to identify all pairs (x, y) where $x \in A \wedge y \in B$ and both x and y correspond to the same real-world entity - these pairs are referred to as matches. When focusing on identifying duplicates within a single dataset, we consider $B = A$ for the remaining definitions in this section.

Figure 1 provides an example with two tables, each containing product profiles with identical attributes ("Product Name", "Manufacturer", "Price"). Solid arrows in the figure represent matches, i.e., the second profile in Table A matches the first profile in Table B, and the third profile in Table A matches the third profile in Table B. The dashed arrow denotes an unmatched pair: the second item in Table A does not match the second item in Table B.

To use pre-trained language models for processing item profiles, the raw texts are first serialized the same way as in existing methods (Li et al., 2021; Miao et al., 2021; Wang et al., 2022): for each data entry $e = (attr_i, val_i)_{1 \le i \le k}$, we let serialize($e$) ::= [COL] $attr_1$ [VAL] val_1 ... [COL] $attr_k$ [VAL] val_k, where [COL] and [VAL] are special tokens that indicate the beginning of attribute names and values respectively. For example, the first item in the left table in Figure 1 can be serialized as follows: *[COL] Product Name [VAL] instant immersion spanish deluxe 2.0 [COL] Manufacturer [VAL] topics entertainment [COL] Price [VAL] 39.99*

Definition 2.1 (Embedding)**.** An embedding model M_{emb} with d dimensions takes a serialized item profile x as input, and outputs a real vector $M_{\mathsf{emb}}(x) \in \mathbb{R}^d$. Given a distance function dist, such as euclidean distance, for a pair of profiles (x, x'), the value of $\mathsf{dist}(x, x')$ is small if and only if (x, x') constitutes a match.

In the interest of simplicity, we normalize all output vectors, implying that the L-2 norm $\|M_{\mathsf{emb}}(x)\|_2 = 1$ for each data item $x \in D$.

Definition 2.2 (Entity Blocking)**.** Entity Blocking, as a critical step in the entity resolution process, involves generating a candidate set of pairs $C = \{(x, y) | x \in A, y \in B\}$ from tables A and B of item profiles.

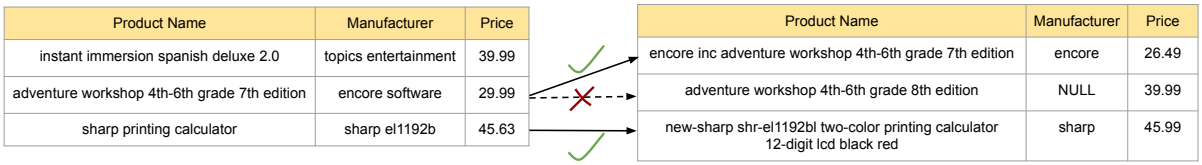


Figure 1: Entity Resolution: determine the matching entries from two datasets.

If we denote the ground-truth matches as G, the aim of the blocking step is to maximize recall $|C \cap G|/|G|$, while minimizing the size of the candidate set $|C|$. A smaller $|C|$ for the same recall implies fewer non-matching pairs in C and higher precision, leading to reduced computational cost for the subsequent matching phase. For a given blocking model, a larger candidate set generally enhances recall but diminishes precision. Therefore, striking a balance between recall and precision is a necessity in real-world applications.

Post fine-tuning of M_{emb}, we apply the embedding model M_{emb} to each data item to generate high-dimensional vectors. In this work, we use the high dimensional similarity search library FAISS (Johnson et al., 2019) to find the k most similar embeddings for every embedding, forming the candidate set. We note that k is a tunable parameter that controls the candidate set size and recall.

3 Reproducibility Dimensions of Entity Blocking

For each data entry, the model M_{emb} will convert each serialized text into in an embedding. In this study, we focus on utilizing a pre-trained Transformer-based language model, such as BERT (Devlin et al., 2018) and its variations like RoBERTa (Robertson et al., 2009). Transformer-based language models excel in generating highly contextualized embeddings that offer enhanced understanding of textual information (Devlin et al., 2018; Robertson et al., 2009; Wolf et al., 2020).

3.1 PLM-based Entity Blocking methods

Utilizing the embeddings directly generated by off-the-shelf pre-trained language models (e.g., RoBERTa (Liu et al., 2019)) has been shown to yield less competitive results (Li et al., 2021). Consequently, recent studies in the entity resolution (ER) literature have focused on fine-tuning pre-trained language models with self-supervised learning objectives specifically for the entity blocking step. These self-supervised methods involve several data augmentation (DA) operators designed for the ER problem (Li et al., 2021; Thirumuruganathan et al., 2021; Wang et al., 2022) to generate similar item pairs for the training. For example, a DA may randomly removes a small portion of words from the profile and the modified profile and the original profile still represent the same entity.

DeepBlocker (Thirumuruganathan et al., 2021) proposes a set of deep learning solutions for entity blocking and one of them employs Sentence-BERT and minimizes the triplet loss. Its DA operator is randomly deleteing up to 40% of the words and it randomly sample negative examples to construct training triplets. Since the authors do not open-source their code for this specific solution, we implement it and tune its performance to the best of our knowledge. To train DeepBlocker, we replace sentence-BERT with RoBERTa (Liu et al., 2019) because it is empirically better in the entity resolution task (Li et al., 2021; Wang et al., 2022). We generate 64 randomly augmented versions for each item as positive examples and randomly sample 64 items in the training set as negative examples.

Sudowoodo (Wang et al., 2022) employs self-supervised contrastive learning, combining state-of-the-art contrastive learning methods SimCLR (Chen et al., 2020) and Barlow Twins (Zbontar et al., 2021). Its includes more diverse DA operators from (Li et al., 2021) and also token embedding level DA operation cutoff (Shen et al., 2020). For training Sudowoodo, we also use RoBERTa as the pre-trained LM and apply Cutoff as the data augmentation operator. We follow most of Sudowoodo's experimental settings for evaluating the ER task, but use Cosine Annealing Warm Restarts for the learning rate scheduler.

3.2 Neural IR Models

In recent literature of Information Retrieval, PLM-based neural retrieval models achieve state-of-the-art performance.

Contriever (Izacard et al., 2021) is a recently proposed neural retrieval model that uses contrastive learning to learn representations for documents and queries. The model is trained on a large corpus of documents and queries, and it learns to map them into a shared embedding space where

documents and queries with similar semantics are close to each other. Its contrastive learning involves hard negative sampling where both in-batch and cross-batch negatives are used for optimizing the InfoNCE loss. For the entity blocking task, a straightforward way is to train the model on the entity blocking data and generate embeddings for getting candidate pairs by similarity search. We use the authors' code[1] to fine-tune the checkpoint "facebook/contriever" and train until the loss converges.

SPLADE (Formal et al., 2022) (Scalable Productive Learning with Automated DEsign) is a recent sparse neural retrieval model that uses a novel architecture that combines dense and sparse representations of documents and queries. The model is trained to predict the probability of a document being relevant to a given query. SPLADE has been shown to outperform other state-of-the-art retrieval models on several benchmark datasets. It can seamlessly migrate to the entity blocking task by preparing training triplets similar to DeepBlocker. To train Splade for entity blocking, we modified the authors' code[2] to load their pre-trained model for fine-tuning, which achieves higher accuracy in our evaluation. The checkpoint we fine-tuned is "navar/splade-cocondenser-ensembledistil".

ColBERT (Khattab and Zaharia, 2020) is a neural retrieval model that uses a transformer-based architecture and a pre-trained longformer model to generate representations for documents and queries. The model relies on fine-grained contextual late interactionis to rank documents based on their relevance to a given query. Like Splade, ColBERT can be easily applied in the entity blocking task by training with triplets. For ColBERT, we train the model with our prepared training data following their instructions[3] and disabled the compression for better accuracy.

3.3 Non-PLM based Methods

In the literature of both entity resolution and information retrieval, many work has focused on non-PLM based methods. In this work, we also include two such methods that are empirically proven to be effective: Sparkly (Paulsen et al., 2023) and BM25 (Robertson et al., 2009). Specifically, Sparkly is a TF-IDF based method and achieves state-of-the-art performance on various datasets.

[1]https://github.com/facebookresearch/contriever
[2]https://github.com/naver/splade
[3]https://github.com/stanford-futuredata/ColBERT

Table 1: Statistics of Common ER Datasets.

Dataset	TableA	TableB	# Matched Pairs
Abt-Buy (AB)	1,081	1,092	1,028
Amazon-Google (AG)	1,363	3,226	1,167
DBLP-ACM (DA)	2,616	2,294	2,220
DBLP-Scholar (DS)	2,616	64,263	5,347
Walmart-Amazon (WA)	2,554	22,074	962

Table 2: Statistics of Processed WDC Product Dataset

Dataset	# clusters	# profiles	# Groundtruth Pairs
Computers_small	1,000	4,050	18,818
Sports_small	1,000	3,505	13,030
Computers_large	121K	165K	215K
Sports_large	257K	311K	222K
Mixed	233K	500K	3.1M

We use Sparkly Auto in our evaluations [4]. For BM25 evaluation, we use a widely used package [5].

3.4 In-Distribution and Out-of-Distribution Generalization

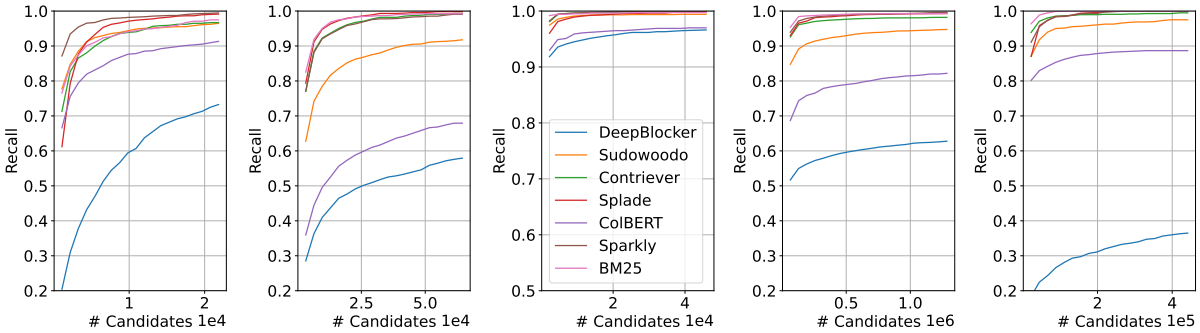
In this work, in-distribution generalization refers to the case where the model only see a small portion of data from a large scale dataset and the model is tested on the whole dataset. We expect PLM-based methods to generalize well in this task because PLMs have been exposed to a significantly large corpse during its pre-training. Similarly, in this work, out of distribution generalization represents the scenario where the model is trained on data from one domain but tested on other domains.

We evaluate the performance of state-of-the-art PLM-based methods for entity blocking on five small scale ER datasets as listed in table 1, and also the WDC dataset, a large-scale product matching dataset (Peeters et al., 2023).

The small ER datasets include various domains such as product profiles and scholar articles, and each dataset contains two tables from two different sources. The goal of entity resolution is the find matched items across two tables. The smallest table in these datasets has 1k profiles while the largest one has 64k profiles. They all have pairwise labels that indicate whether the pairs match or not. We use this set of datasets to evaluate out-of-distribution generalization abilities of different methods, by training each model using the training set of Abt-Buy and evaluating on five datasets.

The WDC dataset comprises 26 million product profiles from 79 thousand websites, wherein profiles representing the same product are grouped together with the same cluster ID. Out of 25 top-level catagories, we randomly select two

[4]https://github.com/anhaidgroup/sparkly
[5]https://pypi.org/project/rank-bm25/



(a) AB (b) AG (c) DA (d) DS (e) WA

Figure 2: Comparison of Models trained on AB

product categories, “Computers_and_Accessories" and “Sports_and_Outdoors", to evaluate the in-distribution generalization of blocking models. We select the following attributes: “title", “brand", “price", “description", and “specTableContent" for evaluation. The Computers dataset we use has 165K profiles, 121K clusters, and 215K matching pairs. The Sports dataset has 311K profiles, 257K clusters, and 222K matching pairs. We use “Computers_large" and “Sports_large" to denote the whole sets of profiles in these categories respectively. From each of the two categories, we randomly sample 1000 clusters that have a size of between 2 and 10 and use every pair of profiles in each cluster as a matching example. We use “Computers_small" and “Sports_small" to denote these two small sets. To further evaluate out-of-distribution generalization abilities, we construct a mixture of four categories from WDC, namely “Shoes," “Automotive," “Electronics," and “Jewelry," which we use "Mixed" to denote. This dataset contains 500K profiles, 233K clusters, and 3.1M matching pairs. Due to hardware limits, we do not explore larger datasets in this work. The statistics of our processed datasets is listed in table 2.

we limit the maximum cluster size when evaluating the performance (namely the recall of groundtruth pairs) of blocking models. This is because some clusters in the WDC dataset contain over a thousand profiles, which can generate millions of matching pairs, thus dominating the testing set. To address this, we randomly select 20 profiles to represent a cluster if its size is larger than 20, enabling us to eliminate the dominating effect of large clusters and focus on the most common use cases for entity resolution.

4 Evaluations

We conduct the evaluations on a machine with a 12-core AMD Ryzen CPU (3.7GHz), 64GB main memory, and a NVidia RTX 4090 GPU (24 GB GPU memory). We limit the length of input tokens of text items to 128 for all methods. For each evaluated method that involves training, we use the same training data to ensure fair comparisons. We followed the best practice of each selected method according to their paper and official GitHub repository, to conduct experiments in the entity blocking task.

The main performance metric is recall in the problem of entity blocking (Thirumuruganathan et al., 2021; Wang et al., 2022). A higher recall can always be achieved by including more candidate pairs, but will a larger number of candidate pairs will incur higher cost for the entity matching step in entity resolution. To compare the effectiveness of difference methods, we draw the curve that represent the relation between the size of the candidate set (or the number of candidates) and the recall in our evaluation. Each figure presented in this section and the following section displays the recall of the corresponding candidate set size on the y-axis, and the number of candidate pairs obtained by kNN search is shown on the x-axis. In our evaluation, k ranges from 1 to 20. This means we find k candidates for each profile in the table, and the total number of candidates equals k multiplies the number of rows in the table. In comparing two methods for entity resolution, our primary focus is on the recall metric. This is because the ultimate goal of entity resolution is to accurately identify all data instances that correspond to the same entity. Under the assumption of achieving equal recall performance, the method that yields the smallest candidate set size is considered as the best. This is because a smaller candidate set size reduces the computational overhead of downstream processing and facilitates more efficient data handling.

We note that BM25 is not trainable and we use BM25 directly on the datasets in our evaluations,

so the curves of BM25 are not related to "out-of-distribution" or "in-distribution". We include the performance of BM25 for easy comparisons with other methods.

4.1 Out-of-Distribution Generalization

Figure 2 shows the out-of-distribution performance of models trained on the Abt-Buy dataset. For reference, we also include their performance on Abt-Buy in Figure 2a. The ranking of model performance on Abt-buy is Sparkly, Splade, Contriever, Sudowoodo, BM25, ColBERT, and DeepBlocker. This indicates that neural IR methods excel at the entity blocking task.

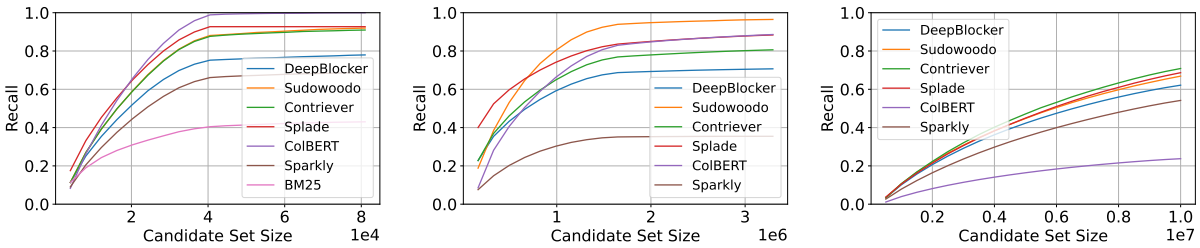
Next, we compare the out-of-distribution generalization ability of all methods in Figure 2b, 2c, 2d, and 2e. DA is fairly easy dataset to solve and all methods achieve good performance on it. Notably, Splade and Contriever consistently out-performs other methods and generalize exceptionally well. Sudowoodo is also competitive in terms of out-of-distribution generalization and achieves reasonable performance on all datasets. ColBERT underperforms sudowoodo and experience obvious performance drop on AG and DS. Lastly, DeepBlocker is less competitive on all datasets but still generalizes well on AG and DS. We only notice a significant performance drop on WA for DeepBlocker. The reason why DeepBlocker underperforms Contriever and Sudowoodo is due to its training does not gaurantee a robust representation learning - its DA operator is simple and it simply randomly select negative examples. On the other hand, both Contriever and Sudowoodo incorporates hard negative sampling techniques, which is a harder training process for the models. Contriever and Sudowoodo are closely related because both use contrastive learning to learn robust representations of the profiles. Their main differences lie in that they use different methods for building positive and negative pairs. Specifically, Contriever uses Inverse Cloze Task and Independent Cropping for building positive pairs, while Sudowoodo uses diverse data augmentation operators to inject "noises" into the original profiles. Furthermore, Contriever uses in-batch and cross-batch negatives for its contrastive learning, while Sudowoodo only uses in-batch negatives but includes clustering-based negative sampling to obtain hard negative examples. As shown in Figure 2a, Contreiver and Sudowoodo have very similar in-distribution performance. However, in Figure 2b, 2c, 2d, and 2e, Sudowoodo underperforms Contriever, which indicates the strategy of building training pairs in Contriever is better than Sudowoodo for out-of-distribution generalization.

4.2 In-Distribution Generalization

In this section, we present the results of our in-distribution generalization evaluations on small-scale training sets (Computers_small and Sports_small), large-scale in-distribution testing sets (Computers_large and Sports_large), and a mix of in-distribution and out-of-distribution testing set (Mixed) consisting of profiles from four different product categories. We trained two models. The first model is trained on Computer_small and its performance is shown in Figure 3. The second model is trained on Sports_small and its performance is shown in Figure 4. It is worth noting that the small training sets are subsets of the large testing sets. Therefore, the models have some prior exposure to a small portion of profiles in the large sets. That being said, none of the methods have been exposed to any profiles in the Mixed set during training because the Mixed set does not overlap with the two training sets.

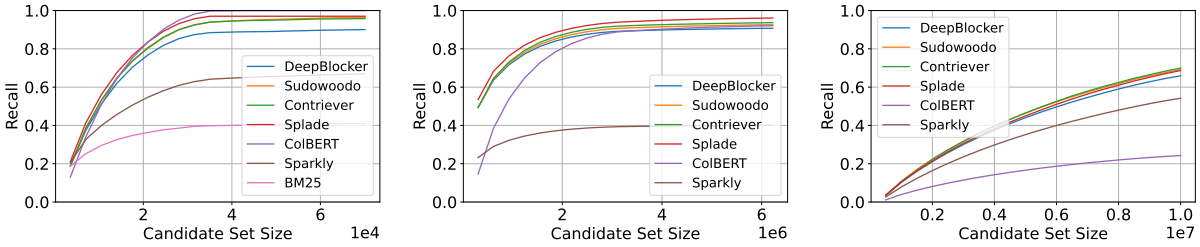
Figure 3a and Figure 4a display the performance of these methods on Computer_small and Sports_small respectively. Overall, on these two datasets, ColBERT outperforms other datasets and Splade is the runner-up. Sudowoodo and Contriever have almost identical performance while DeepBlocker is less competitive. These results are similar to the previous section and demonstrates that neural IR models can achieve excellent performance in the entity blocking task. However, traditional methods, Sparkly and BM25, obviously under perform PLM-based methods. Especially, BM25 does not scale to larger datasets due to the quadratic computation complexity.

Figure 3b and Figure 4b display the performance of these methods on Computer_large and Sports_large respectively and indicate the in-distribution generalization abilities of these methods. Although ColBERT is the top-performer on smaller datasets, there is a noticable performance drop on these large datasets, which indicates an inferior in-distribution generalization ability. Sudowoodo outperforms all other methods on Computer_large and is very close to the top-performer on Sports_large, which indicates an excellent in-distribution generalization ability. Contriever has a



(a) Computer-small (b) Computer-large (c) Mixed

Figure 3: Comparison of Models trained on Computers_small



(a) Sports-small (b) Sports-large (c) Mixed

Figure 4: Comparison of Models Trained on Sports_small

noticable drop on Computer_large while still performs well on Sports_large. Splade has a consistent performance on both datasets and demonstrates excellent generalization abilities. DeepBlocker underperforms other methods but its performance is almost consistent in the in-distribution generalization setting.

Finally, Figure 3c and Figure 4c show the performance of these methods under the mixed out-of-distribution generalization setting, where the trained models are evaluated on a mixture of out-of-distribution datasets. On these datasets, all of the methods suffer a noticable performance drop at k=20. We note that this is mainly caused by a higher density of matched profiles in the dataset. To explain, the number of profiles in Mixed is at most more than 3 times larger than the two large datasets, but the number of groundtruth pairs in Mixed is around 15 times more. This caused the entity blocking task to be much harder and requires larger candidate set sizes to achieve higher recalls. Contriever outperforms other methods under this setting, while Sudowoodo and Splade are very close to it. DeepBlocker slightly underperforms the aforementioned three methods. However, ColBERT sufffers a significant performance drop and it falls far behind other methods under this setting. This is likely due to insufficient training examples involved during this training. In order to achieve optimal performance for Late-interaction methods like ColBERT, it often requires quite a bit of data for training. Therefore, ColBERT is not an idead choice when only a limited number of training data is available.

4.3 Running Time Evaluation

Next, in Table 3, Table 4, and Table 5, we report detailed running time separated by training, indexing, and retrieval, for various entity blocking methods on different datasets.

Among the considered methods, DeepBlocker, Sudowoodo, and Contriever require the least amount of time for training their models, while Splade requires at least 3 times longer time. Notably, ColBERT needs the longest training time, exceeding 10 hours for each training set.

Regarding indexing and retrieval time, DeepBlocker, Sudowoodo, and Contriever use very similar embedding processes and they all employ FAISS-GPU for kNN search. This results in similar performance for these three methods, although the indexing time of contriever is longer than other methods. FAISS-GPU creates index and performs similarity search on the GPU and is an order of magnitude faster than FAISS-CPU in our evaluation and the searching time for each dataset is less than half a minute in all of our evaluations. Next, Splade requires the least indexing time, while its retrieval is significantly slower than that of the FAISS-GPU-based methods, especially on large scale datasets. Its scalability over the dataset size is limited. We note that Splade's slow retrieval is partially due to the sequential nature of its current retrieval code - it can only process one query in GPU at a time. Finally, ColBERT is the most computationally expensive method, requiring the

Table 3: Training Time of each evaluated method.

Datasets	DeepBlocker	Sudowoodo	Contriever	Splade	ColBERT
Abt-Buy	6m30s	3m94s	32m54s	1h33m	10h14m
Computers_small	24m51s	8m42s	33m02s	1h34m	10h19m
Sports_small	36m15	10m18s	32m50	1h33m	10h28m

Table 4: Indexing Time for all samples in each dataset.

Datasets	DeepBlocker	Sudowoodo	Contriever	Splade	ColBERT
Abt-Buy	2.8s	2.5s	2.4s	8.3s	35.5s
Amazon-Google	2.7s	2.4s	2.1s	7.2s	28.8s
DBLP-ACM	5.1s	5.0s	4.9s	8.7s	38.8s
DBLP-Scholar	42.1s	41.3s	40.6s 37.7s	10.1s	37.4s
Walmart-Amazon	19.5s	22.8s	26.2s	9.8s	37.2s
Computers_small	7.5s	7.4s	16.35s	3.1s	39.6s
Sports_small	8.9s	9.0s	14.11s	3.0s	15.2s
Computers_large	2m05s	2m04s	11m13s	1m27s	7m25s
Sports_large	3m53s	3m53s	21m19s	2m46s	10m22s
Mixed (computer)	6m13s	6m13s	35m8s	4m24s	14m37s
Mixed (Sports)	6m25s	6m13s	33m53s	4m24s	13m43s

Table 5: Retrieval Time for all samples in each dataset. DNF=Did Not Finish within 1 week

Datasets	DeepBlocker	Sudowoodo	Contriever	Splade	ColBERT	BM25	Sparkly
Abt-Buy	0.43s	0.40s	0.39s	16s	5.8s	2.1s	26.2s
Amazon-Google	0.41s	0.44s	0.40s	35s	12.25s	6.9s	31.9s
DBLP-ACM	0.40s	0.39s	0.41s	25s	10.51s	19.3	27.6s
DBLP-Scholar	0.78s	0.81s	0.80s	10m40s	4m19s	8m11s	57.1s
Walmart-Amazon	0.50s	0.52s	0.49s	3m14s	1m24s	2m41s	38.2s
Computers_small	0.44s	0.55s	0.43s	35s	9m47s	1m38s	36.7
Sports_small	0.41s	0.47s	0.43s	30s	4m31s	1m37	31.6s
Computers_large	2.93s	2.98s	2.83s	29m47s	45m52s	DNF	2m39s
Sports_large	6.92s	7.83s	6.89s	1h15m34s	1h39m46	DNF	1m37s
Mixed	15.10s	15.40s	15.34s	2h20m02s	5h32m0s	DNF	13m24s

longest training on all datasets and the longest retrieval time on many datasets, although its indexing time is acceptable.

In this work, we use IndexFlatL2 in FAISS-GPU, which is a basic method. The FAISS-GPU library consists of various kNN search techniques[6] including quantization, LSH-based methods, and graph-based search. These techniques can be potentially useful for PLM-based entity blocking on larger scale datasets.

5 Related Work

The field of Entity Resolution (ER) - the process of determining data items that represent the same real-world entity - has been a significant area of research over the past few decades (Getoor and Machanavajjhala, 2012; Konda et al., 2016). This process typically consists of two primary phases: blocking and matching. While a considerable body of work has proposed deep learning techniques for the matching phase (Kasai et al., 2019; Peeters et al., 2020; Li et al., 2021; Miao et al., 2021; Akbarian Rastaghi et al., 2022; Yao et al., 2022), the literature on techniques for the blocking phase remains comparatively sparse. Most recently, (Papadakis et al., 2023) and (Zeakis et al., 2023) perform experimental studies on traditional blocking workflows and embedding-based nearest-neighbor search methods. They analyze the relative performance of the main representatives per category over numerous established datasets. However, these works do not study the generalization capabilities of state of the art methods in the literature, and they overlook latest advancements in the Information Retrieval (IR) literature which is important because the entity blocking problem can also be solved using IR methods. Our work focuses on the generalization capabilities of latest advancements in both the entity resolution and information retrieval literature.

Blocking is pivotal as it minimizes the number of pairs to be compared in the matching phase. This is particularly vital considering the number of potential pairs can potentially reach the square of the dataset size, making a naive approach computation-

[6]https://github.com/facebookresearch/faiss/wiki/Faiss-indexes

ally prohibitive, especially when employing intricate models such as deep neural networks (Wang et al., 2022; Li et al., 2021). These networks may need to process millions or even billions of items. The goal of the blocking phase is to identify as many genuine matches as feasible while maintaining a minimal candidate set.

The research community has proposed a variety of blocking techniques, including rule-based blocking (Das et al., 2017; Gokhale et al., 2014; Paulsen et al., 2023), schema-agnostic blocking (Simonini et al., 2019), meta-blocking (Simonini et al., 2016), deep learning techniques (Thirumuruganathan et al., 2021; Zhang et al., 2020), and Locality-Sensitive Hashing (LSH)-based blocking techniques (Borthwick et al., 2020). These methods can manage billions of items for entity matching. Most recently, sparkly (Paulsen et al., 2023) is a TF-IDF based method and achieves state-of-the-art performance on various datasets. However, its success relies on sophisticated tuning of the parameters for each specific dataset, which cannot generalize in the out-of-distribution setting.

Recently, pre-trained language models, such as BERT-based models, have been leveraged to encapsulate the semantics of text items (Li et al., 2021; Wang et al., 2022; Peeters and Bizer, 2022). These models are refined using contrastive learning methods and/or labeled data to generate item embeddings. Similar item pairs can then be discerned by executing a similarity search on the embeddings. For instance, Peeters et al. (Peeters and Bizer, 2022) proposed R-SupCon, a supervised contrastive learning model for product matching, utilizing the learned embeddings for blocking.

Entity blocking can also be framed as an Information Retrieval (IR) task. Recent literature in IR (Tonellotto, 2022) has seen methods such as DPR (Karpukhin et al., 2020), GTR (Ni et al., 2021), and Contriever (Izacard et al., 2021), which utilize Pre-trained Language Model-based (PLM) methods to learn dense document representations. Candidate pairs can be identified by conducting a similarity search on these dense representations using FAISS (Johnson et al., 2019). Conversely, Splade++ (Formal et al., 2022) learns sparse representations and constructs an inverted index for lexical matching. This fundamentally circumvents the necessity for pair-wise comparison and scales appropriately to large datasets. The out-of-distribution generalization capabilities has been studied in the IR literature (Thakur et al., 2021), and the evaluations incidate neural retrievers or rankers can poorly generalize to different domains/datasets .

6 Conclusion

In this paper, we study the reproducibility of large language model based methods for entity blocking. Our study shows that state-of-the-art PLM-based methods from both the entity resolution and information retrieval literature performs generally well in the in-distribution generalization evaluations for the entity blocking task. We also provide detailed break up of the running time for comprehensive understanding of each part of these PLM-based methods. The majority of these methods also demonstrate excellent out-of-distribution generalization abilities. We highlight the challenge of achieving good performance when the density of matched profiles is higher on larger mixed datasets.

Limitations

Unlike traditional entity blocking methods, PLM-based methods requires training on real data samples, either labeled data or unlabeled data. Although PLM-based methods have empirically shown success on a wide range of real-world datasets, they typically require more computational resources than traditional entity blocking methods. For example, the training and embedding of these methods requires the usage of GPUs to be effective, while traditional methods do not need GPUs and are more accessible to broader users. Despite that, the practice of training PLM-based methods for the entity blocking task is a promising direction.

Ethical Considerations

Statement of Intended Use The foundation of our research is built upon open-source datasets sourced from e-commerce platforms and computer science bibliography platforms. These datasets are characterized by a diverse array of descriptions related to everyday consumer products and research papers. It is crucial to note that when our work is applied to scenarios like customer profile consolidation, where the data involves attributes of human demographics, strict measures for data privacy must be enforced. This includes the essential step of de-identification or anonymization of such data to protect individual privacy.

References

Mehdi Akbarian Rastaghi, Ehsan Kamalloo, and Davood Rafiei. 2022. Probing the robustness of pre-trained language models for entity matching. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3786–3790.

Mikhail Bilenko and Raymond J Mooney. 2003. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48.

Andrew Borthwick, Stephen Ash, Bin Pang, Shehzad Qureshi, and Timothy Jones. 2020. Scalable blocking for very large databases. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 303–319. Springer.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Sanjib Das, Paul Suganthan G. C., AnHai Doan, Jeffrey F. Naughton, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, Vijay Raghavendra, and Youngchoon Park. 2017. Falcon: Scaling up hands-off crowdsourced entity matching to build cloud services. In *SIGMOD*, pages 1431–1446.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. 2022. From distillation to hard negative sampling: Making sparse neural ir models more effective. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 2353–2359, New York, NY, USA. Association for Computing Machinery.

Lise Getoor and Ashwin Machanavajjhala. 2012. Entity resolution: theory, practice & open challenges. volume 5, pages 2018–2019. VLDB Endowment.

Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F. Naughton, Narasimhan Rampalli, Jude W. Shavlik, and Xiaojin Zhu. 2014. Corleone: hands-off crowdsourcing for entity matching. In *SIGMOD*, pages 601–612.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource deep entity resolution with transfer and active learning. In *ACL*, pages 5851–5861.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Pradap Konda, Sanjib Das, Paul Suganthan G. C., AnHai Doan, and et al. 2016. Magellan: Toward building entity matching management systems. volume 9, pages 1197–1208.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2021. Deep entity matching with pre-trained language models. volume 14, pages 50–60.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhengjie Miao, Yuliang Li, and Xiaolan Wang. 2021. Rotom: A meta-learned data augmentation framework for entity matching, data cleaning, text classification, and beyond. In *SIGMOD*, pages 1303–1316.

Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *SIGMOD*, pages 19–34.

Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. 2021. Large dual encoders are generalizable retrievers.

George Papadakis, Marco Fisichella, Franziska Schoger, George Mandilaras, Nikolaus Augsten, and Wolfgang Nejdl. 2023. Benchmarking filtering techniques for entity resolution. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 653–666. IEEE.

Derek Paulsen, Yash Govind, and AnHai Doan. 2023. Sparkly: A simple yet surprisingly strong tf/idf blocker for entity matching. *Proceedings of the VLDB Endowment*, 16(6):1507–1519.

Ralph Peeters and Christian Bizer. 2022. Supervised contrastive learning for product matching. *arXiv preprint arXiv:2202.02098*.

Ralph Peeters, Christian Bizer, and Goran Glavas. 2020. Intermediate training of BERT for product matching. In *DI2KG@VLDB*.

Ralph Peeters, Anna Primpeli, and Christian Bizer. 2023. Wdc product data corpus and gold standard for large-scale product matching-version 2.0.

Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Dinghan Shen, Mingzhi Zheng, Yelong Shen, Yanru Qu, and Weizhu Chen. 2020. A simple but tough-to-beat data augmentation approach for natural language understanding and generation. *arXiv preprint arXiv:2009.13818*.

Giovanni Simonini, Sonia Bergamaschi, and H. V. Jagadish. 2016. BLAST: a loosely schema-aware meta-blocking approach for entity resolution. volume 9, pages 1173–1184.

Giovanni Simonini, George Papadakis, Themis Palpanas, and Sonia Bergamaschi. 2019. Schema-agnostic progressive entity resolution. volume 31, pages 1208–1221.

Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Synthesizing entity matching rules by examples. *Proceedings of the VLDB Endowment*, 11(2):189–202.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *arXiv preprint arXiv:2104.08663*.

Saravanan Thirumuruganathan, Han Li, Nan Tang, Mourad Ouzzani, Yash Govind, Derek Paulsen Glenn Fung, and AnHai Doan. 2021. Blocking in entity matching: A design space exploration. volume 14, pages 2459–2472.

Nicola Tonellotto. 2022. Lecture notes on neural information retrieval.

Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Chengliang Chai, Guoliang Li, Ruixue Fan, and Xiaoyong Du. 2022. Domain adaptation for deep entity resolution. In *Proceedings of the 2022 International Conference on Management of Data*, pages 443–457.

Runhui Wang, Yuliang Li, and Jin Wang. 2022. Sudowoodo: Contrastive self-supervised learning for multi-purpose data integration and preparation.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Dezhong Yao, Yuhong Gu, Gao Cong, Hai Jin, and Xinqiao Lv. 2022. Entity resolution with hierarchical graph attention networks. In *Proceedings of the 2022 International Conference on Management of Data*, pages 429–442.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. 2021. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR.

Alexandros Zeakis, George Papadakis, Dimitrios Skoutas, and Manolis Koubarakis. 2023. Pre-trained embeddings for entity resolution: An experimental analysis. *Proceedings of the VLDB Endowment*, 16(9):2225–2238.

Wei Zhang, Hao Wei, Bunyamin Sisman, Xin Luna Dong, Christos Faloutsos, and David Page. 2020. Autoblock: A hands-off blocking framework for entity matching. In *WSDM*, pages 744–752.