ERABQS: entity resolution based on active machine learning and balancing query strategy

Article *in* Journal of Intelligent Information Systems · March 2024

DOI: 10.1007/s10844-024-00853-0

4 authors:

Mourad Jabrane
Ecole nationale science appliqués khouribga
6 PUBLICATIONS **12** CITATIONS
SEE PROFILE

Hiba Tabbaa
6 PUBLICATIONS **117** CITATIONS
SEE PROFILE

Yassir Rochd
Ensa khouribga
11 PUBLICATIONS **23** CITATIONS
SEE PROFILE

Hafidi Imad
Ecole nationale science appliqués khouribga
66 PUBLICATIONS **296** CITATIONS
SEE PROFILE

Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-024-00853-0

RESEARCH

ERABQS: entity resolution based on active machine learning and balancing query strategy

Jabrane Mourad[1] · Tabbaa Hiba[1] · Rochd Yassir[1] · Hafidi Imad[1]

Received: 7 November 2023 / Revised: 29 February 2024 / Accepted: 1 March 2024

Abstract
Entity Resolution (ER) is a crucial process in the field of data management and integration. The primary goal of ER is to identify different profiles (or records) that refer to the same real-world entity across databases. The challenging problem is that labeling a large sample of profiles can be very expensive and time-consuming. Active Machine Learning (ActiveML) addresses this issue by selecting the most representative or informative profiles pairs to be labeled. The informativeness is determined by the capacity to diminish the uncertainty of the model. Conversely, representativeness evaluates whether a selected instance effectively reflects the overall input patterns of unlabeled data. Traditional ActiveML techniques typically rely on one strategy, Which may severely restrict the performance of the ActiveML process and lead to slow convergence. Especially in ER problems with a lack of initial training data. In this paper, we overcame this issue by inventing an approach for balancing the two above strategies. The implemented solution named EBEES (Epsilon-based Balancing Exploration and Exploitation Strategy), Which contains two variations: Adaptive-ϵ and ϵ-decreasing. We evaluated the EBEES on twelve datasets. Comparing the EBEES strategy against the state-of-the-art methods, without an initial training data, showed an enhanced performance in terms of F1-score, model stability, and rapid convergence.

Keywords Entity resolution · Duplicate detection · Record linkage · Active machine learning

1 Introduction

Entity resolution, also known as entity matching, entity deduplication or duplicate detection, is the process of identifying or marking as duplicate profiles (records) that are identical or similar to the others within a dataset. In this process, a "profile pair" refers to a pair of profiles from one or more datasets. These profiles pairs are being compared to determine whether they represent the same real-world entity. In the field of ER, supervised machine learning offers a

✉ Jabrane Mourad
mourad.jabrane@usms.ac.ma

[1] Laboratory of Process Engineering, Computer Science and Mathematics, National School of Applied Sciences, Bd Béni Amir, BP 77, Khouribga 25000, Morocco

Published online: 26 March 2024

powerful capacity for learning Christophides et al. (2021); Papadakis et al. (2021); Alexakis et al. (2022); Helgertz et al. (2021). However, efficient training needs a huge quantity of labeled profile pair, which means that the data is already tagged with the correct answer. The challenge is increased when an exceedingly large and useful labeled data collection is needed. For example, in the industrial product area, a learning-based matching model needs up to 1.5 million labeled profiles pairs to attain an F1-score of 99% Dong and Rekatsinas (2018). The labeling process may be time-consuming and expensive Wu et al. (2020). This cost can be due to the requirement of one or more experts, the use/choice of a measure similarity function, or a prohibitive computation time. Those challenges are addressed by the activeML techniques, which have the significant ability to minimize the labeling time and costs while ensuring performance, in which these techniques use only a limited quantity of labeled profile pairs and provide it to an Oracle for labeling in order to update the model.

The main challenge and critical step in ActiveML systems is the selection strategy. ActiveML algorithms often utilize two kinds of query selection criteria; “Informativeness” and “Representativeness” criteria Settles (2012), which are the key concepts in data analysis. Representativeness, widely known by the term exploration, measures how well a profile pair reflects the overall characteristics of a dataset, indicating its alignment with the general patterns. On the other hand, informativeness, also known as exploitation, assesses the capacity of the data to unveil new knowledge or patterns that contribute to reducing uncertainty in predictions. While representativeness gauges typicality within the data, informativeness uncovers less apparent, informative aspects. Therefore, the decision of which profile pair to label might be regarded as a dilemma between exploration and exploitation of the input data space. To the best of our knowledge, all ActiveML algorithms in the ER context only use one of these two types of selection strategies, which may severely restrict their performance. In a cold start problem, when there are no labeled data available to train the initial model, approaches that favor informative profile pair rarely take advantage of the unlabeled profile pair structure, due to the extremely imbalanced data in ER when the negative class (non matching) is significantly larger than the positive class (matching Christen et al. (2015), which causes substantial bias in the data and, resulting in insufficient performance. Conversely, approaches that favor representative profile pair lead to slow convergence due to the huge number of examples required before identifying the best decision boundary.

We address those issues by developing an EBEES (Epsilon-based Balancing Exploration and Exploitation Strategy) solution that uses both strategies (informativeness and representativeness) throughout the entire ActiveML process. The EBEES contains two balancing strategies: Adaptive-ϵ and ϵ-decreasing. These strategies were tested on twelve databases from distinct ER domains.

In light of this, the remainder of this work is structured as follows: Section 2 presents the essential notations and describes the problem of ER and ActiveML. Section 3 introduces and reviews recent related studies. The proposed approach is explained in detail in Section 5. Section 6 demonstrates the experiment’s assessment and results. Section 7 discusses the EBEES results. In Section 8 we present our analysis, conclusions, and recommendations for further research.

2 Preliminaries

This section provides an overview of two key concepts: entity resolution and ActiveML.

2.1 Entity resolution

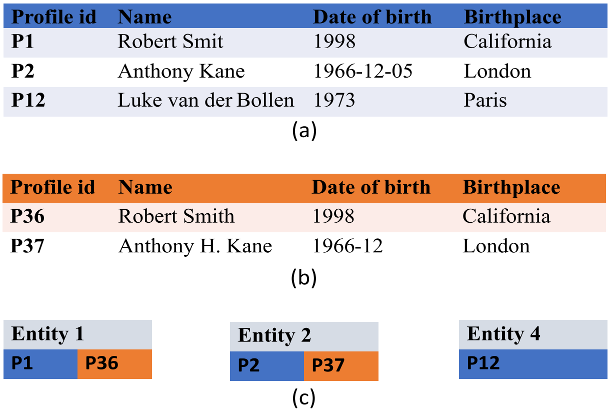
Entity resolution (ER) is an important task in a variety of contexts, such as cleaning up a database, detecting fraudulent activity or finding duplicate documents within a file system. This can be useful in healthcare, finance, and government, where accurate and consistent data is critical for decision-making. In the ER, the essential element is the profile, which is alternatively referred to as an instance, entity description, or reference. Each profile conveys information about a specific real-world entity, which may include an event, location, organization, or a person. This concept can be expressed in a more formal manner, as follows:

Definition 1 (Profile):
A profile P_i is a subset of a data source (DS), i.e., $P_i \subset DS$, offering information about the same entity.

Definition 2 (Entity):
An entity E_k is a subset (noted also as a collection) of profiles, i.e., $E_k = P_1, P_2, ...P_n$, where each P_i refers to the same entity.

Within this context, Fig. 1 serves as a key illustration, depicting an ER task performed over two databases, labeled respectively as (a) and (b). The task is characterized by three textual attributes. Furthermore, part (c) of the figure demonstrates the result of the ER process, where each entity comprises at most one profile from each dataset.

In the ER process, records are compared based on one or more attributes, such as name, address, or date of birth. The objective is to determine which records refer to the same entity, even if the attributes are not identical or have errors. The challenge arises from the potential data which may be incomplete, inconsistent, or have variations due to different datasets or data entry errors. This process depends on the matching functions like Dharavath and Singh (2015); Levenshtein (1965); Jaro (1989), which are designed to assess the level of similarity between two profiles, denoted as P_i and P_j. This similarity, often termed as matching probability or resemblance, serves as the basis for determining whether two profiles ought to be grouped together within the same entity. The output of a similarity function is usually a number between 0 and 1, where 1 means the two profiles are identical and 0 means



Profile id	Name	Date of birth	Birthplace
P1	Robert Smit	1998	California
P2	Anthony Kane	1966-12-05	London
P12	Luke van der Bollen	1973	Paris

(a)

Profile id	Name	Date of birth	Birthplace
P36	Robert Smith	1998	California
P37	Anthony H. Kane	1966-12	London

(b)

Entity 1	
P1	P36

Entity 2	
P2	P37

Entity 4
P12

(c)

Fig. 1 An illustration of the two databases : (a) and (b) represent the input database, while (c) denoting the associated entities

they are completely dissimilar. Table 1 presents profile pairs (PP) with their similarity across various attributes: name, birthplace, and date of birth. The similarity is quantified using two well-known similarities function: Levenshtein and Jaccard. Each row represents a unique pair of profiles (e.g., PP1(1-36), PP2(2-36)), with their corresponding values of similarities for each attribute. Higher values represent greater similarity, with a maximum score of 1.0 indicating complete similarity. The following table illustrates the effectiveness of similarity functions in identifying and comparing profile similarities.

2.2 Active machine learning

ActiveML is a type of machine learning (ML) in which the learning algorithm can actively choose which data to learn from rather than being passively fed with a fixed dataset. This can be useful in situations where the amount of available data is limited or whether it is expensive or time-consuming to acquire additional data, like in ER problems. In ActiveML, the learning algorithm is able to interact with an Oracle, which represents the supplier of the labeled data, to select the most useful examples to learn from. The Oracle may take the form of a human annotator or a function that is able to provide labels for the data. This interaction allows ActiveML to strategically select the most useful examples from which to learn, rather than being confined to a static dataset. This methodology significantly boosts the efficiency and effectiveness of ML classifiers, by focusing on acquiring labels for particularly useful profile pairs, ActiveML optimizes the learning process, ensuring that the classifier's performance is enhanced through a more judicious and targeted selection of data. This approach contrasts with traditional methods that rely on a fixed dataset, showcasing ActiveML's innovative strategy in prioritizing data quality over quantity to improve classification outcomes. The ActiveML process is illustrated by the diagram shown in Fig. 2, the elements of which are explained as follows:

1. Bootstrap (initial data): Gathering a small amount of labeled data to use as a starting point for the training.
2. Train the model: Apply the initial data to train a ML model.
3. Classify: All examples without labels are categorized by the trained matching model.
4. Query the model (Selection): Allow the model to interact with its environment by selecting the informative or representative unlabeled profiles pairs and request their corresponding labels. Figure 3 illustrates various strategies for selecting informative and representative profile pairs in an ER binary classification problem. Part (a) depicts the initial classification challenge. Part (b) demonstrates a selection method that targets

Table 1 Profile similarity using text similarity metrics

profile pair id	name levenshtein	name jaccard	birthplace levenshtein	birthplace jaccard	date of birth levenshtein	date of birth jaccard
PP1(1-36)	0.916	0.333	1.0	1.0	1.0	1.0
PP2(2-36)	0.0	0.0	0.199	0.0	0.199	0.0
PP3(12-36)	0.105	0.0	0.300	0.0	0.5	0.0
PP4(1-37)	0.142	0.0	0.199	0.0	0.285	0.0
PP5(2-37)	0.857	0.666	1.0	1.0	0.7	0.666
PP6(12-37)	0.157	0.0	0.0	0.0	0.285	0.0

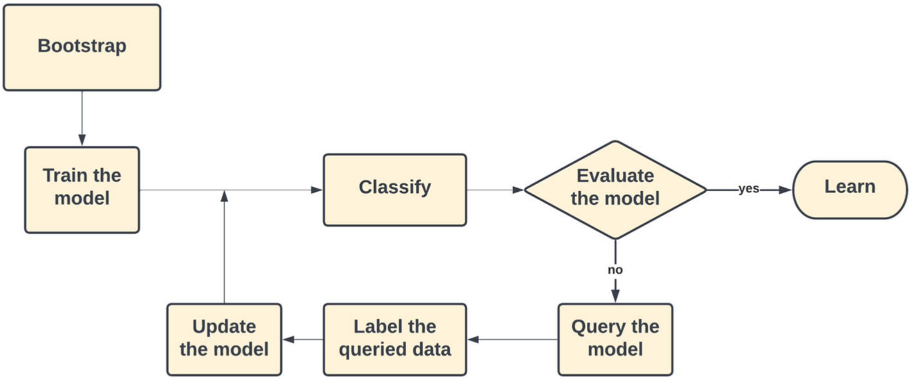


Fig. 2 The ActiveML workflow for the ER

informative pairs, likely focusing on those that provide crucial classification insights, often near decision boundaries. Part (c) contrasts this by showing a representative strategy, where the selection is distributed across the data space to reflect the overall dataset characteristics. Finally, part (d) introduces the proposed approach, which combines both strategies, aiming for a balanced selection of profile pairs that are informative and representative, potentially leading to a more robust and accurate classification model.

5. Label the queried data: Engage human annotator or another method to label the data that was requested by the model.
6. Update the model: Utilize the newly labeled data to update the model.
7. Evaluate the model: Use a separate dataset to test the performance of the updated classifier. When the requirement is met, a supervised method is used to develop a matching classifier for the final collection of labeled profile pair. Otherwise, repeat the ActiveML process by enabling the model to perform more queries and updating the model with freshly annotated data as required until the criteria are satisfied.

In an ActiveML architecture, the first labeled dataset called Bootstrap Set (BS) is crucial. The learner should have access to a tiny piece of the dataset that has been labeled. In real-world scenarios, obtaining or selecting this BS from an original dataset might be exceedingly time-consuming, costly, or useless Chen et al. (2021a). In a cold-start scenario where BS for training the initial classification model is absent, can result in the development of a model that is unstable and of low quality. What makes the problem of selecting the bootstrap set

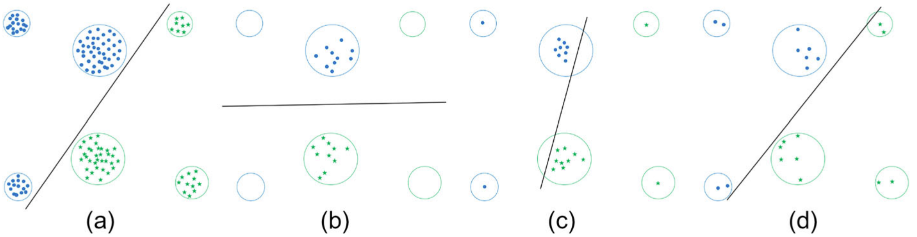


Fig. 3 A Demonstrative example illustrating the Selection of informative and representative profiles pairs: (a) An ER binary classification problem, (b) A method emphasizing informative strategy, (c) A method prioritizing representative strategy, (d) The proposed approach

more difficult is the imbalanced data issue. Statistically, the number of identical pairs rises linearly with the number of profiles, but the number of distinct pairs rises quadratically Getoor and Machanavajjhala (2012). This indicates that the dominant class corresponds to distinct profile pairs (negative occurrences), whereas the minority class refers to identical ones (positive occurrences). Also, the essential kernel of the ActiveML techniques is how the profile pairs are rated and chosen for querying their labels in each iteration. These techniques usually employ a single query strategy during the whole querying phase. There are two approaches for the querying. The first is straightforward to apply. The second method could yield spectacular performance but would require substantial computing, especially if used throughout the entire ActiveML querying phase. In practical applications, it turns out to be inefficient and lacks sufficient performance.

3 Related work

In this section, we present a survey of related work on ER within the context of machine learning. Additionally, we offer a comprehensive comparison between these studies.

In general, ER algorithms are categorized as supervised, unsupervised, or semi-supervised (ActiveML) based on the availability of labeled data.

3.1 Supervised algorithms

The approach identifies ER as a problem of binary classification, where a profile pair is the input and a matching or non-matching decision is the output. They use labeled data using matching (positive) and non-matching (negative) profile pairs to tackle this problem. In this direction, Reyes-Galaviz et al. (2017) described two models with Multilayer Perceptrons using one hidden layer and a single output node. Every internal block combines the scores of a particular similarity metric over all feature values, or the scores of a particular feature value across all similarity metrics. The significance of the related similarity measure or attribute value is determined by the learned weights between the input and the output nodes. In the other direction, Yan et al. (2020) portrayed matching as a threshold-based regression problem, whose purpose is to categorize profile matches into one of the five groups. During the regression model's learning, the thresholds for each group are discovered. This allows for a variety of application configurations to be met. Finally, Chen et al. (2020) presented a learnable and interpretable paradigm for risk analysis. Its purpose is to rate all labeled cases based on their chances of being misclassified.

In light of deep learning's significant achievements, there has been a surge of interest in representation learning for tabular data. Recent endeavors have focused on self-supervised objectives, encompassing predictive modeling of masked cells, identification or correction of corrupted cells, and employment of contrastive losses with augmentations Bahri et al. (2022); Somepalli et al. (2021). While representation learning has demonstrated remarkable success in text and knowledge bases, limited research has extensively explored contextualized representation learning applied to relational web tables. Two notable approaches involved the direct adoption of pre-trained language models for ER JointBERT Peeters and Bizer (2021) and DITTO Li et al. (2023). A dual-objective training approach (JointBERT Peeters and Bizer (2021)) incorporated binary matching and multi-class classification to compel the model to not only determine the match/non-match status but also predict the entity identifier for each entity description within a training pair. While DITTO Li et al. (2023) employed a language model for semantic parsing of natural language queries over tabular data by investigating the language model's proficiency in performing ER on tabular data-i.e., determining if two rows pertain to the same object.

In the same direction, TdavER Li and Wu (2023) introduced a novel ER model called Transformer-based Denoising Adversarial Variational ER. Designed to address limitations in existing deep learning-based ER methods, TdavER eliminated the need for manual labeling and facilitated easier model transfer to new datasets. The model has two core components: the first uses denoising autoencoders and pre-trained language models to create robust entity representations without supervision, while the second utilizes adversarial variational autoencoders to ease training constraints, enabling the model to generalize better to new datasets. For incremental learning, Vieira et al. (2019) proposed an incremental ER system for data integration, finding duplicates as new data is arrived through queries. It prioritizes efficiency and could potentially leverage machine learning techniques like clustering for matching records.

3.2 Unsupervised algorithms

In an unsupervised algorithmic context, SL-AW Jurek et al. (2017) constructed a set of Self-Learning classifiers, each of which uses a unique collection of similarity metrics. Self-learning is expanded by Hou et al. (2019) to progressive ML, in which seed cases are selected to train a model that is gradually extended to identify increasingly difficult unlabeled profile pairs. Zhang et al. (2018) provided a graph-theoretic technique based on two steps; the first, dubbed ITER, constructs a weighted bipartite network using the assumption that positive profile pairs share different feature value tokens. The second, dubbed CliqueRank, creates a network in which nodes represent individual profiles and edges link couples that share a minimum single token, weighted by ITER's similarity metric. In the same context, Niknam et al. (2021) proposed a fix by considering "transitive closure" to improve the evaluation, especially for unsupervised machine learning approaches.

3.3 Semi-supervised algorithms

Analogous to supervised techniques, every case refers to a profile pair and generates a feature vector, with each dimension indicating the score obtained by a specific or aggregated similarity metric for a specific characteristic. In general, earlier studies applied the workflow shown in Fig. 2 to limit the number of labeled cases necessary for developing a strong and successful classification model. To begin, a small collection of labeled examples is prepared, called seeds, in order to bootstrap the prediction model(s). The absence of such labeled profile pairs at the start is referred to as the cold-start issue Primpeli et al. (2020). To address this problem, either the seeds are selected randomly Qian et al. (2017) or using the distribution of similarity results Sarawagi and Bhamidipaty (2002) for labeling by the oracle. Primpeli et al. (2020) provided multiple fundamental techniques that eliminate the need for oracle involvement and do not impose a predefined limit on the number of seeds. It aggregated the feature vectors and utilized a threshold fixed by the score distribution's elbow point to discriminate between positive profile pairs and negative ones. The occurrences with the greatest degree of confidence are then chosen as seeds. The next phase involves training the prediction model(s) using the already-known labeled cases. Following that, the trained model classifies all unlabeled occurrences. Afterwards, a termination criterion is assessed. This may be as easy as achieving a fixed number of iterations or as sophisticated as achieving an estimated performance objective across an unlabeled profile pair Meduri et al. (2020). When this requirement is met, a supervised technique is used to learn a prediction model(s) using all the labeled profile pairs in the final set. Otherwise, the ActiveML process proceeds to the choice stage, where the most confusing unlabeled cases are identified. These are the cases that are more difficult to classify (i.e., they refer to the trained model's minimal level of confidence). After manually labeling the chosen occurrences, the training process is repeated.

Due to the extremely imbalanced data in ER, the most critical stage in the ActiveML workflow is querying the model. Using a random approach at this stage would result in a group of cases dominated by negative profile pairs. To address this challenge, numerous approaches have been suggested in the literature, each of which employs distinct strategies based on heuristics to select unlabeled, ambiguous examples effectively. One such approach is AdInTDS Christen et al. (2015), where the collection of unlabeled profile pairs is partitioned into clusters. This method focuses on identifying and choosing the most representative examples from each cluster, thereby streamlining the selection process. Complementing this, T3S Bianco et al. (2015) adopted a different tactic by dividing the similarity scores of unlabeled profile pairs into bins and randomly selecting one profile pair from each bin. This approach not only introduces a degree of randomness but also aims to eliminate unnecessary events, concentrating efforts on the most informative examples. Such diversity in methodologies highlights the complexity of the problem and the need for varied solutions to tackle different aspects of it. Additionally, LFP/LFN Qian et al. (2017) using rules and a similarity heuristic, the first step selects probable false positives, and the second improves recall by identifying the most false negatives. On the other hand, approaches based on classification choose ambiguous unlabeled cases based on the model's confidence across the current training data. the first strategy called Query-By-Committee (QBC). This strategy involves training a number of classifiers and selecting ambiguous cases based on the degree of disagreement between their predictions, as estimated by the entropy or variance Meduri et al. (2020); Mozafari et al. (2014). ALIAS Sarawagi and Bhamidipaty (2002) used homogeneous sets of classifiers, with each classifier taught using a different set of (randomized) variables from a distinct part of the training data. Active Atlas also makes use of a committee of decision trees Tejada et al. (2002). Random forests provide a more natural strategy, with each single tree serving as the model committee Meduri et al. (2020); Mozafari et al. (2014) expanded QBC into a learner-agnostic technique that may be used in conjunction with any ML model, whether linear, convex, tree-based, or rule-based.

The most confusing cases in the margin-based strategy are those that are closest to the classifiers' decision boundary, which is a hyperplane for linear models or 0.5 classifying probabilities for nonlinear models Meduri et al. (2020); Mozafari et al. (2014). In a similar context of addressing ER under resource constraints, ZeroER Wu et al. (2020) was introduced as an innovative framework that obviates the need for any labeled examples. This framework is predicated on a novel self-training algorithm that is designed to iteratively refine its own predictions, thereby learning to accurately label unlabeled profile pairs. This self-supervised approach offers a unique solution for ER tasks, particularly in settings where acquiring labeled data is challenging or impractical.

With the increasing number of neural network-based matching methods that have been proposed and pushed the state-of-the-art performance, especially for textual ER tasks. DIAL Jain et al. (2021) is a deep ActiveML approach that demonstrates scalability by concurrently optimizing the recall for blocking and the accuracy for matching blocked pairs through jointly learning embeddings. The methodology of DIAL is based on the Index-By-Committee framework, where each committee member utilizes robust pre-trained transformer language models to acquire representations. In the same context of deep ActiveML, a recent study, ALER-Risk Nafa et al. (2022) has shifted attention towards a novel approach that leverages risk sampling for ActiveML. This method involves the selection of representative data points with an elevated misclassification risk for labeling. ALER-Risk accomplished this by formulating an optimization model that utilizes an upper bound of the core-set loss, incorporating non-uniform Lipschitz continuity.

3.3.1 Comprehensive comparison

In general, various strategies have been employed to optimize the learning process and improve the model performance in the field of ER with ActiveML. This section provides a comprehensive comparison of these strategies based on key criteria such as seed selection methods, the incorporation of neural network-based approaches, and the basis of strategy.

- Cold-start issue: The cold-start issue in the ER is addressed through various seed selection strategies. Traditional methods, such as those cited in Primpeli et al. (2020); Qian et al. (2017), employ random selection or distribution-based approaches. Primpeli's technique Primpeli et al. (2020) is notable for not requiring oracle interaction and using score distribution's elbow point for discrimination, while Qian et al. (2017) focuses on the distribution of similarity results. A significant limitation of these methods is their potential lack of representativeness. Random selection can miss important data characteristics, while distribution-based methods may not adequately capture the diversity of real-world data, leading to biased initial models. All other strategies Christen et al. (2015); Sarawagi and Bhamidipaty (2002); Meduri et al. (2020); Bianco et al. (2015); Mozafari et al. (2014); Tejada et al. (2002); Jain et al. (2021); Nafa et al. (2022) are established on the existence of a BS for training the initial model, which does not apply in the real-world scenarios.
- Basis of Strategy: Informative or Representative: The aforementioned approaches in ER with ActiveML predominantly employ either informative or representative strategies. Informative strategies, such as those used in Sarawagi and Bhamidipaty (2002); Meduri et al. (2020); Bianco et al. (2015); Mozafari et al. (2014); Tejada et al. (2002); Jain et al. (2021), focus on selecting examples that provide the most information for model improvement. Representative strategies used in Christen et al. (2015); Nafa et al. (2022), aim to select a subset of the data that represents the overall dataset distribution. The exclusive use of either strategy can be limiting. Informative strategies may lead to a focus on outliers or non-generalizable cases, while representative strategies might overlook critical edge cases or nuanced data features. A more balanced approach that combines the elements of both strategies could potentially yield more robust and generalizable models.

4 Motivation and contributions

The existing body of research indicates challenges in the cold-start problems and basic strategy balancing in the field. The underlying motivations for our approach are:

- **Enhancing ActiveML performance:** Traditional active learning methods often limit themselves by querying either informative or representative unlabeled instances. Implementing balancing strategies can significantly improve the performance of ActiveML processes.
- **Addressing the cold-start issue:** Many models initially suffer from inadequate data, leading to poor predictive quality. Integrating both informative and representative strategies can help overcome this issue.

Guided by these motivations, our proposed contributions are as follows:

- **Balanced ActiveML strategy:** We introduce a novel ActiveML approach that incorporates balanced strategies and outperforms existing simplistic methods.
- **Cold-start ActiveML method:** Our technique addresses the cold-start problem in ActiveML without additional labeling costs, ensuring high performance at all stages.
- **Evaluation on diverse datasets:** We conduct a comprehensive assessment on twelve datasets of three types of ER problems: structured, textual, and dirty data, demonstrating the effectiveness of our proposed solution.

5 The proposed strategy

The objective is to integrate both strategies utilizing the **EBEES** solution, which incorporates both the informative and representative strategies within the same query strategy and at different steps in the ActiveML process. To achieve this, we balance the exploration and exploitation strategies through the utility metric.

5.1 Theoretical foundations

The EBEES strategy lends itself to mathematical formalization to enhance its theoretical rigor. For instance, the utility function $U(PP)$ can be formulated as:

$$U(PP) = \epsilon R(PP) + (1 - \epsilon) I(PP). \tag{1}$$

The utility score is used to calculate the utility of a profile pair PP by combining its representativeness score $R(PP)$ and informativeness score $I(PP)$. The balance between these two components is controlled by the parameter ϵ. Outlined here is the way in which this formula operates in selecting representative and informative profile pairs during the activeML process:

Typically, $R(PP)$ indicates how well the PP reflects the overall characteristics of the dataset, which is estimated by:

$$R(PP) = \frac{1}{|Ul|} \sum_{PP' \in Ul} \operatorname{sim}\left(PP, PP'\right). \tag{2}$$

The $R(PP)$ calculates the representativeness of profile pair PP by its average similarity to all the other profile pairs through computing the average similarity of a given PP to every other profile pair PP' in the unlabeled dataset Ul. The function $\operatorname{sim}\left(PP, PP'\right)$ is a similarity measure like cosine similarity, euclidean distance, Pearson's correlation coefficient, Spearman's rank correlation, or any other appropriate measure of similarity. The higher the value, the more closely the given profile pair matches the rest of the data. Consequently, the formula $R(PP)$ effectively selects representative profile pairs by quantifying how similar each pair is to the rest of the dataset. Those pairs with higher average similarity scores are deemed more representative.

The $I(PP)$ calculates the informativeness of a profile pair PP by measuring its uncertainty. This informativeness is determined by multiple functions:

- Least Confident: This strategy is often applied in classification tasks, where the model predicts the likelihood of each class for a given profile pair. The formula for the least confident strategy is focused on the probability of the most probable class predicted by the model. The idea is to measure the model's uncertainty based on how confident it is in its most confident prediction Settles (2012). For a given profile pair PP, if $P(y_1 \mid PP)$ is the probability of the most probable class y_1 predicted by the model, then the least confident score can be expressed as:

$$I_{Least\_Confident}(PP) = 1 - P(y_1 \mid PP). \tag{3}$$

Where $P(y_1 \mid PP)$ represents the highest probability assigned to any class for the profile pair PP. The least confident score ranges from 0 to 1. A score closer to 0 indicates higher confidence in the model's prediction, while a score closer to 1 indicates lower confidence.

- Margin: The margin score focuses on the difference between the probabilities assigned to the two most probable classes Settles (2012). The formula for the margin sampling in this context is:

$$I_{Margin}(PP) = P(y_1 \mid PP) - P(y_2 \mid PP), \tag{4}$$

Where $P(y_1 \mid PP)$ is the probability of the most likely class according to the model and $P(y_2 \mid PP)$ is the probability of the second most likely class. The idea is to find profile pairs where this margin is the smallest, meaning the model is most uncertain about these profile pairs. These are the data points where the model's prediction for the most likely class and the second most likely class are very close. By selecting profile pairs with the smallest margins, you aim to pick the examples where the model is currently the most uncertain, thus potentially gaining the most information from labeling these profile pairs.

- Entropy: A widely-used and quite general strategy in informative sampling (originally introduced by Shannon (1948)), the entropy of a profile pair PP can be calculated based on the probability distribution over the classes. The formula for calculating the entropy is as follows:

$$I_{Entropy}(PP) = -\left[P\left(y_1 \mid PP\right)\log_2 P\left(y_1 \mid PP\right) + P\left(y_2 \mid PP\right)\log_2 P\left(y_2 \mid PP\right)\right], \tag{5}$$

Where $P\left(y_i \mid PP\right)$ is the model's estimated probability for class y_i; $i \in \{1, 2\}$.
The idea behind this formula is to sum over all classes, considering the probability of each class and its logarithm. The negative sign ensures that the entropy is either a positive or a zero value. When the probability $P(y_i \mid PP)$ is high for one class and low for the other, the entropy will be low, indicating low informativity. Conversely, if the probabilities are evenly distributed across the classes (i.e., the model is unsure which class is the most likely), the entropy will be high, indicating high informativity.

The parameter ϵ, introduced in formula (1), is a weighting factor that determines the relative importance of representativeness and informativeness in the utility score. It ranges between 0 and 1. When ϵ is closer to 1, the utility score $U(PP)$ places more emphasis on representativeness. Contrariwise, when ϵ is closer to 0, the utility score prioritizes informativeness. Following this, during the ActiveML process, the profile pairs with higher utility scores are selected. By adjusting ϵ, you can control whether the process favors more representative or more informative profile pairs. This flexibility allows the algorithm to adapt to different stages of learning or specific objectives. At the beginning of the learning process, you might choose a higher value of ϵ to focus on building a general understanding of the dataset (emphasizing representativeness). As the learning progresses, you might decrease ϵ to focus more on extracting new, valuable information (emphasizing informativeness). Then the value of ϵ can be dynamically adjusted as needed throughout the learning process to continuously balance between exploring the general characteristics of the data and exploiting specific, informative aspects.

5.2 Algorithms

In order to achieve an optimal balance in the querying strategies, we introduce two distinct variations of the EBEES algorithm that rely on the parameter ϵ

1. Adaptive ϵ using ϵ with a dynamic value, it is based on the F1 performance progress instead of the ActiveML process. A low value of the F1-score produces a high ϵ value, which means using exploration more than exploitation. A higher value of the F1-score results in a lower ϵ value, thus leaning more towards exploitation over exploration.

Algorithm 1 : Adaptive Epsilon Query Strategy.

```
Input:
    $Ul$: Unlabeled data.
    $\epsilon$: The initial value of $\epsilon$.
    $\epsilon\_min$: The final value of $\epsilon$.
    $d\_r$ : Decay rate.
    $f1\_sw$ : Sliding window of the F1-score.
    $f1\_s$ : List of F1-scores.
    $tl$ : True labels.
    $model$ : Machine learning model.
Output: $spp$ : Selected profile pair.
1: function ADAPTIVE_EPSILON_QS($\epsilon, \epsilon\_min, dr, f1\_s\_w, f1\_s, model, tl$)
2:     $predicted\_labels = model.predict(Ul)$
3:     $f1\_s.add(get\_f1\_score(tl, predicted\_labels))$
4:     if $len(f1\_sw <= f1\_s)$ then
5:         $f1\_mean = mean(f1\_s[-f1\_sw : -1])$
6:         if $f1\_s[-1] < f1\_mean$ then
7:             $epsilon = max(\epsilon\_min, \epsilon * dr)$
8:         end if
9:     end if
10:    $utilities = \epsilon R(PP) + (1 - \epsilon)I(PP)$
11:    $idx = argmax(utilities)$
12:    $spp = Ul[idx]$
13:    return $spp$
14: end function
```

Algorithm 1 requires several parameters. The initial value of ϵ establishes the balance between exploration and exploitation. The parameter ϵ_min is the minimum value that ϵ can reach. Additionally, the decay rate d_r dictates the speed at which ϵ decreases. The algorithm also uses a sliding window for the F1-score, denoted as $f1\_sw$. This window considers a set number of recent F1-score values (i.e predetermined quantity of the latest F1-score values) to compute the average F1-score. The list is represented as $f1\_s$, stores the scores calculated during the process. Another key input is the true labels, tl, which are the actual labels for the unlabeled data and are essential for calculating the F1-score. Finally, the ML model, referred to as $model$, is used for making predictions. The $Adaptive\_epsilon\_QS$ takes the above inputs and implements the adaptive epsilon query strategy (Line 1). First, the model predicts the class of each profile pair in unlabeled data Ul (Line 2). Afterwards, the F1-score is calculated based on the true labels and predicted labels, and added to the list of F1-scores (Line 3). If the length of the sliding window does not exceed the length of the F1-scores list, we calculate the mean of recent the F1-scores contained within this window (Lines 4-5). If the most recent F1-score is less than the calculated mean, ϵ is updated. The calculation involves either multiplying by the decay rate d_r or set to ϵ_min, depending on which of the two is higher. This step dynamically adjusts ϵ based on the performance of the model. When the recent F1-score is lower, signifying poorer performance, it results in an increased ϵ value, thereby encouraging greater exploration (Lines 6-8). The utility scores are calculated for each profile pair (data point) in the unlabeled data (Line 10). This utility is a combination of exploration score ($\epsilon R(PP)$) and exploitation score($(1-\epsilon)I(PP)$). Afterwards, the index of the data point with the highest utility is found (Line 11). Then the profile pair corresponding to this index in the Ul is returned as the selected profile pair spp (Lines 12-13). To summarize, this algorithm dynamically adjusts the balance between exploring new predictions and exploiting known information, guided by the ML model's recent performance as indicated by the F1 score. This strategy proves useful in cases where adapting the model to changing data patterns over time is necessary.

2. ϵ-Decreasing: Using this strategy, it allows us to set an initial ϵ value that exponentially decreases over time or over iterations. This ensures that highly explorative behavior occurs early in the experiment and highly exploitative behavior occurs at the end. The key to this balance is the epsilon parameter (ϵ), which starts at a high value and decreases over time. When epsilon values are high, typically at the beginning of the experiment, the algorithm emphasizes exploration for the reason it has little information about the dataset. As epsilon decreases, the weight shifts towards exploitation - selecting the most informative samples - given that the algorithm has accumulated knowledge about the dataset, it is in a position to make more informed choices.

Algorithm 2 Epsilon Decreasing Query Strategy.

```
Input:
    $Ul$: Unlabeled data
    $\epsilon$ : Value of $\epsilon$
    $\epsilon\_start$ : Start value of $\epsilon$
    $\epsilon\_end$ : Final value of $\epsilon$
    $i$ : Iteration number
    $N_t$ : Total number of ActiveML iterations
Output: $spp$ : Selected profile pair
1: function EPSILON_DECREASING_QS($Ul, \epsilon, \epsilon\_start, \epsilon\_end, i, N_t$)
2:     $dr = ln(\epsilon\_end)$
3:     $\epsilon = \epsilon\_start * e^{dr*i/N_t}$
4:     $utilities = \epsilon R(PP) + (1-\epsilon)I(PP)$
5:     $idx = argmax(utilities)$
6:     $spp = Ul[idx]$
7:     return $spp$
8: end function
```

The Algorithm 2 takes unlabeled data Ul that the model is set to process, the current value of ϵ, which is a key parameter in controlling the exploration-exploitation balance, ϵ_start and ϵ_end which represent the initial and final values of ϵ, the current iteration number i in the ActiveML process, and N_t represents the total number of iterations for the entire ActiveML process. The function $Epsilon\_decreasing\_QS$ implements the epsilon decreasing strategy and receives the inputs detailed above (Line 1), the function initiates its process at Line 2 by calculating a decay rate, which is defined as the negative logarithm of ϵ_end (Line 2). This rate determines how quickly ϵ should decrease over time. Following this, the value of ϵ is then dynamically updated using an exponential decay formula (Line 3). It initiates at ϵ_start and decreases exponentially with each iteration i, in proportion to the total number of iterations N_t. The formula for exponential decay is presented as follows:

$$\epsilon = \epsilon\_start * e^{dr*i/N_t}.$$

Then the profile pair with the highest utility is returned from the unlabeled data (Lines 4-7). In this way, the algorithm balances exploration and exploitation by decreasing the value of ϵ over the ActiveML process. The algorithm starts with a high value of ϵ at the beginning, which allows for more exploration. As the number of iterations increases, the value of ϵ decreases, leading to more exploitation.

We draw attention to the fact that in every iteration of the ActiveML process, the query strategy function in Algorithms 1 and 2 is invoked to compute the new value of ϵ and select the profile pair in accordance with this newly calculated value. The initial values of the input parameters are described in Section 6.

In terms of computational complexity, the foremost factor affecting the complexity of both algorithms is the computation of the utility score, which is expressed as $O(U(PP)) = O(R(PP)) + O(I(PP))$. To elaborate, the complexity of $R(PP)$ is $O\left(\sum_{PP' \in Ul} \text{sim}(PP, PP')\right) = O(n)$, where n represents the number of unlabeled profile pairs Ul. This is due to the computation of the similarity score for each pair, which scales linearly with the dataset size. On the other hand, according to the formulas (3), (4) and (5), the complexity of $I(PP)$ is $O(2)$, this results from the fact that ER being a binary classification task. the complexity of this step is essentially $O(C)$, where C is the number of classes. In the context of our algorithms, where ER involves distinguishing between two classes, C equals 2, leading to a constant-time operation. Consequently, the overall computational complexity for both algorithms is dominated by the $O(n)$ term, stemming from the computation of $R(PP)$. Therefore, we can determine that the computational complexity of both algorithms is $O(n)$, indicating that their performance scales linearly with the number of unlabeled profile pairs.

5.3 Feature vector construction for ER

In our study, we address the ER challenge between two datasets, source and target, with the aligned schemata. To minimize computational overhead, we employ blocking based on a manually selected, domain-specific attribute that serves as a strong identifier. We utilize Relaxed Jaccard combined with inner Levenshtein distance, setting a threshold of 0.2 for block formation. Subsequent to blocking, we construct feature vectors for each entity pair by calculating similarity scores for the individual attributes. These similarity scores are computed using an array of metrics tailored to the data type: Levenshtein and Jaccard for strings; absolute difference for numeric attributes; and day, month, and year differences for date attributes. In the case of string attributes exceeding an average length of six tokens, we incorporate cosine similarity computations using the TF-IDF weighting. All calculated scores are normalized to the [0, 1] range, and any missing values are assigned a score of -1 to ensure their inclusion without compromising the integrity of the dataset.

6 Experiments and evaluation

6.1 Experimental setup

The EBEES algorithms, as delineated in Section 5, were implemented using Python 3.9. The computational experiments were conducted on a system running Ubuntu 20.04, powered by an Intel Xeon E5-2620 2.40GHz 12-core CPU, and equipped with 48GB of RAM. This setup ensures a standardized and replicable environment for future studies.

6.2 Software libraries and tools

To enhance the reproducibility of our work, we utilized well-known software libraries. Specifically, we employed the **scikit-learn**[1] library for ML algorithms, **ModAL**[2] for the ActiveML, and the **pandas**[3] library for data manipulation.

6.3 Datasets

In this segment, we meticulously outline the datasets[4] [5] employed for the comprehensive evaluation of our proposed EBEES. To ensure a robust evaluation framework, these datasets were carefully selected from diverse domains and cover three distinct ER areas:

- Structured Datasets:
  - BeerAdvo-RateBeer: This dataset amalgamates beer-related information from BeerAdvocate and RateBeer, curated as part of a class project in the CS 784 data science course at UW-Madison during the fall of 2015.
  - iTunes-Amazon: Similar in origin to BeerAdvo-RateBeer, this dataset merges music data from iTunes and Amazon, also created by students of the CS 784 course at UW-Madison.
  - Fodors-Zagats: This dataset merges restaurant information from Fodors and Zagat, which includes a list of golden matches denoting tuple pairs that refer to the same restaurant.
  - DBLP-ACM and DBLP-Scholar: Both datasets contain bibliographic data. The former is from DBLP and ACM. The latter is from DBLP and Google Scholar, each accompanied by a list of golden matches.
  - Walmart-Amazon: This dataset, which combines product information from Walmart and Amazon, was originally inclusive of the additional attribute “proddescrlong”, which was subsequently removed to maintain structural consistency.
  - DbPedia-DNB and DbPedia-Viaf: These datasets were extracted by Primpeli and utilize owl Sameaslinks to connect DbPedia profiles to those of DNB and VIAF, respectively.
- Textual Datasets:
  - Abt-Buy: The dataset in question combines product information from Abt.com and Buy.com, complete with a list of golden matches.
  - Amazon-Google: Similar to Abt-Buy, this dataset combines product data from Amazon and Google, also accompanied by a list of golden matches.
- Dirty Datasets: WDC-Phones and WDC-Headphones: These datasets were sourced from the Web Data Commons (WDC) project and feature e-commerce data about phones and headphones, respectively. It is noteworthy that these datasets are characterized by missing features and inconsistent value normalization.

[1] https://scikit-learn.org

[2] https://modal-python.readthedocs.io

[3] https://pandas.pydata.org

[4] https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md

[5] https://github.com/wbsg-uni-mannheim/UnsupervisedBootAL/tree/master/datasets

To situate our empirical evaluation within the broader research landscape, it is pertinent to note that these datasets have also been utilized in the experimental evaluation of other cutting-edge ER algorithms. The names, common attributes, and references of these datasets are cataloged in Table 2, and the details of the experimental datasets are presented in Table 3.

Table 2 Characteristics of Datasets: Source, Target, Shared Attributes, and Prior Usage in Literature

Dataset	Common attributes	References
DBLP - ACM	title, authors, venue, year	Li et al. (2023); Li and Wu (2023); Jain et al. (2021); Fu et al. (2020); Mudgal et al. (2018)
DBLP - Scholar	title, authors, venue, year	Wu et al. (2020); Li et al. (2023); Li and Wu (2023); Jain et al. (2021); Fu et al. (2020); Mudgal et al. (2018)
iTunes - Amazon	id, song name, artist name, album name, genre, price, copyRight, time, released	Li et al. (2023); Li and Wu (2023); Mudgal et al. (2018)
Walmart - Amazon	title, category, brand, modelno, price	Li et al. (2023); Jain et al. (2021); Chen et al. (2020a); Fu et al. (2020); Mudgal et al. (2018)
BeerAdvo - RateBeer	beer name, brew factory name, style, ABV	Li et al. (2023); Li and Wu (2023); Mudgal et al. (2018)
Fodors - Zagat	name, addr, city, phone, type, class	Wu et al. (2020); Li et al. (2023); Li and Wu (2023); Mudgal et al. (2018)
dbpedi - dnb	author name, birthdate, deathdate, gender	Primpeli et al. (2020)
dbpedia - viaf	author name, birthdate, deathdate, gender	Primpeli et al. (2020)
wdc - phones	manufacturer, weight, processor, f_cam_resolution, memory, width, material, depth, brand, resolution, type, height, display_size, color, dimensions, r_cam_resolution, ram, model_num	Primpeli et al. (2020); Petrovski and Bizer (2020); Petrovski et al. (2017)
wdc - headphones	weight, sensitivity, series, model, impedance, brand, micro, max_input, type, height, color, frequency, compatibility, connectivity	Primpeli et al. (2020); Petrovski and Bizer (2020); Petrovski et al. (2017)
abt - buy	name, description, price	Wu et al. (2020); Li et al. (2023); Primpeli et al. (2020); Jain et al. (2021); Mudgal et al. (2018); Chen et al. (2020a); Brunner and Stockinger (2020)
amazon - google	name, description, price, manufacturer	Wu et al. (2020); Li et al. (2023); Primpeli et al. (2020); Jain et al. (2021); Fu et al. (2020); Mudgal et al. (2018); Chen et al. (2020a)

Table 3 The established datasets for ER

	D_1 - D_2	$\|D_1\|$ - $\|D_2\|$	$\|N_A\|$	$\|Nl_{train}\|$	$\|Nl_{test}\|$	$\|Nlp_{train}\|$	$\|Nlp_{test}\|$	$\|Nln_{train}\|$	$\|Nln_{test}\|$	CR %
Structured datasets										
S1	DBLP- ACM	2,616- 2,294	4	7,417	2,473	1,332	444	6,085	2,029	17.9
S2	DBLP- Scholar	2,616-64,263	4	17,223	5,742	3,207	1,070	14,016	4,672	18.6
S3	iTunes- Amazon	6,907-55,923	8	321	109	78	27	243	82	24.4
S4	Walmart - Amazon	2,554 - 22,074	5	6,144	2,049	576	193	5,568	1,856	9.3
S5	BeerAdvo - RateBeer	4,345 - 3,000	4	268	91	40	14	228	77	15.0
S6	Fodors - Zagat	533-331	6	567	189	66	22	501	167	11.6
S7	Dbpedia - Dnb	2,230 - 9,999	4	13,864	3,465	2,310	577	11,554	2,888	16.6
S8	Dbpedia - Viaf	2,230 - 10,198	5	15316	4807	2,552	801	12,764	4,006	16.6
Dirty datasets										
D1	wdc-phones	51-448	18	1762	440	206	51	1556	389	11.6
D2	wdc-hdphone	51-444	14	1163	290	180	45	983	245	15.4
Textual datasets										
T1	Abt-Buy	1,081-1,092	3	5,743	1,916	616	206	5,127	1,710	10.7
T2	Amazon-Google - Product	1,114-1,291	4	6,755	1,687	1,041	259	5,714	1,428	15.3

$|D_i|$ is the number of records, $|N_A|$ the number of attributes, and $|Nl|$, $|Nlp|$, $|Nln|$ are the numbers of labeled, matching and non-matching profiles pairs in the training or testing set. CR is the class ratio

6.4 Evaluation measures

The following factors are frequently taken into account while evaluating the success of ER ActiveML techniques; they are ranged between 0 and 1, with greater values representing more efficacy:

Precision score Calculates the proportion of properly matched profiles:

$$Pr_s = \frac{TP}{FP + TP}.$$

Recall score Computes the proportion of detected matches:

$$Re_s = \frac{TP}{TP + FN}.$$

F1-score Denotes the harmonic mean of Re_s and Pr_s:

$$F1\_s = \frac{2.Pr_s.Re_s}{Pr_s + Re_s},$$

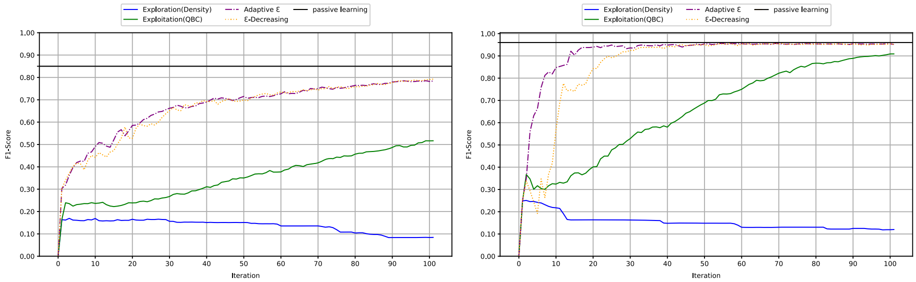
Where TP (True Positive) represents the number of positive profile pairs that actually refer to the same ones, FP (False Positive) represents the number of positive profile pairs that belong to different ones, and FN (False Negative) represents the number of negative profile pairs that relate to the same ones.

In an ER context, where the classes are imbalanced, the F1-score is employed as the one score that summarizes all scores of an ER approach's success. As shown by Hand and Christen (2017), the F1 is identical to the arithmetic average of the recall and precision.

6.5 Experimental results and discussion

To critically assess the efficacy of our EBEES algorithms in various settings, we executed a comprehensive series of experiments deploying the algorithm with multiple ϵ-based strategies. These were juxtaposed against traditional methodologies Wu et al. (2020); Li et al. (2023); Li and Wu (2023); Primpeli et al. (2020); Mudgal et al. (2018); Petrovski and Bizer (2020); Papadakis et al. (2023); Chen et al. (2021). Our experimental framework entailed 10 separate runs, devoid of any bootstrap data. The number of iterations was determined based on the dataset's size. In the context of ActiveML, we envisaged a scenario where an unlabeled pool contains all possible record pairs, whereas the labeled set is initialized as empty. Each iteration in this framework corresponds to a single manual annotation. A Random Forest classifier is trained in every iteration with the pairs of the labeled set. To ascertain the stability of our model across different datasets, we relied on the average F1-score, which was computed for each iteration using separate evaluation datasets.

In the experimental setup, special attention was given to the selection ϵ values for the various strategies, for the $Adaptive - \epsilon$ and $\epsilon - Decreasing$ strategies, the ϵ values were dynamically determined following the formulas detailed in the algorithm of Section 5. After thorough testing and fine-tuning of both strategies, we set $epsilon\_start$ at 1 as the initial value of $epsilon$, which allows the algorithm to prioritize exploration early in the training phase. This broad exploration helps avoid biases and ensures a comprehensive understanding of the data. As time progresses, reducing epsilon to 0 shifts the focus towards exploiting the



(a) WDC-Phones

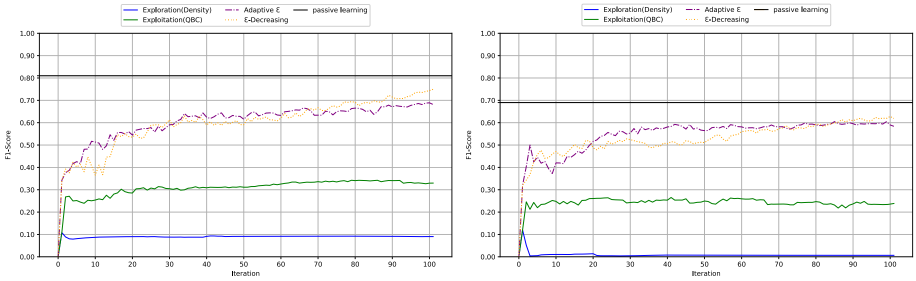
(b) WDC-Headphones

Fig. 4 F1-score per ActiveML iteration – Dirty datasets

model's learned knowledge. To realize this, we set 0 as the final value of $epsilon$ for both ϵ_end and ϵ_min. This gradual transition enhances the model's performance by utilizing insights from the exploration phase. On the other hand, we empirically determined an optimal sliding window size of 10 and a decay rate of 0.8 through extensive testing and tuning. These parameters effectively balance the data responsiveness with model stability and ensure a controlled adjustment of ϵ, achieving an optimal exploration-exploitation trade-off. Our findings demonstrate that these values significantly improve the algorithm's performance, highlighting their effectiveness in activeML scenarios.

Figures 4, 5, and 6 present the F1-score for each iteration in the EBEES strategies, comparing them with the two normal ActivaML query strategies (exploration and exploitation query) and the upper learning bound of the passive learning, in which all the available training data is used. We note that our EBEES solves the cold start issue across all datasets by creating stable models. During the initial ActiveML iterations (from 1% to 40%, depending on the dataset), EBEES consistently produces a higher-quality prediction model in comparison to the two standard ActiveML techniques for all the datasets. Additionally, the EBEES F1 stability increases after 40% of iterations, and the F1 converges to supervised ML performances. Thus, in an ActiveML environment with a restricted budget, especially when considering human annotations, our EBEES solution outperforms the others by consistently achieving satisfactory performance, even when pausing at any iteration.

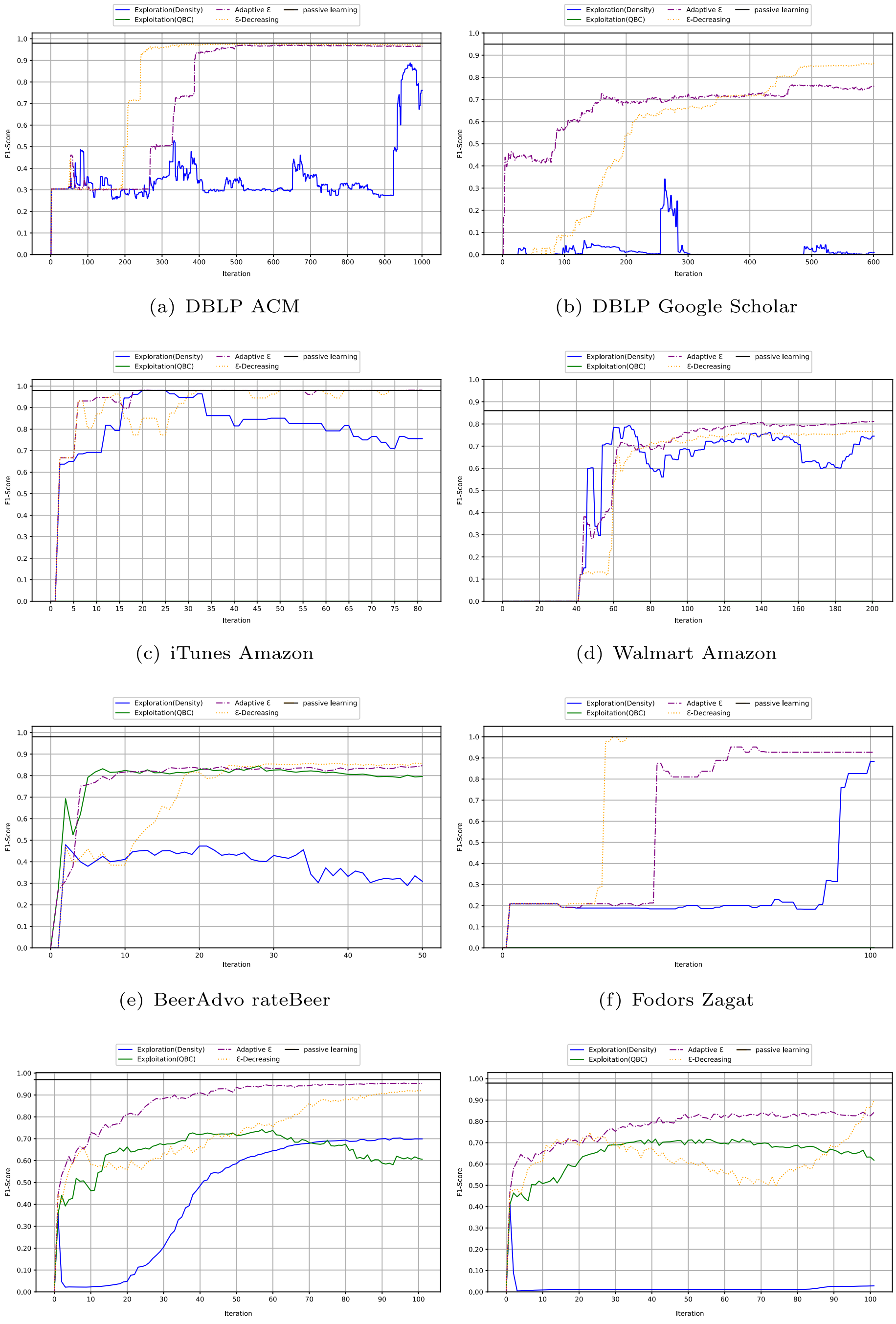
Drawing upon the empirical evidence furnished by the preceding figures, it can be conclusively stated that the EBEES algorithm outperforms traditional ActiveML methodologies across all the iterations in an ActiveML context. Notably, in scenarios typified by cold-start



(a) Abt-buy

(b) Amazon-Google-Product

Fig. 5 F1-score per ActiveML iteration – Textual datasets



(a) DBLP ACM

(b) DBLP Google Scholar

(c) iTunes Amazon

(d) Walmart Amazon

(e) BeerAdvo rateBeer

(f) Fodors Zagat

(g) Dbpedia DNB

(h) Dbpedia VIAF

Fig. 6 F1-score per ActiveML iteration – Structured datasets

conditions, the traditional strategies demonstrated slower convergence rates and manifested unstable performance metrics. Tables 4, 5, and 6 provide a detailed evaluation of the EBEES algorithm along with its variants. These tables compare their performances against traditional ActiveML query strategies and the latest supervised and semi-supervised methods. The evaluations span a range of datasets, including structured, dirty, and textual types. Throughout these comparisons, the F1-scores are used as the key metric for assessing the performance.

Table 4 highlights the effectiveness of the EBEES algorithm in the structured datasets across diverse domains. It showcases its performance in various areas, ranging from bibliographic databases such as DBLP-ACM and DBLP-Google Scholar to e-commerce platforms including iTunes-Amazon and Walmart-Amazon. The details of the performance for each dataset are the following:

- DBLP ACM and DBLP Google Scholar: In both of these bibliographic datasets, the EBEES algorithm's $\epsilon - decreasing$ strategy outperforms traditional ActiveML methods like density (Exploration) and QBC (Exploitation). Notably, for the DBLP-ACM database, the $\epsilon - decreasing$ strategy outperforms the state-of-the-art semi-supervised F1-scores, compared to the best available scores (0.960), and it even comes remarkably close to the supervised F1 (0.989) by achieving an F1-score of (0.972).
- iTunes-Amazon and Walmart-Amazon: These e-commerce datasets further validate the robustness of the EBEES. In the iTunes-Amazon dataset, the $Adaptive - \epsilon$ strategy achieves superior results to the state-of-the-art semi-supervised method, with an F1-score of 0.981 compared to 0.498.
- BeerAdvo-RateBeer and Fodors-Zagat: These datasets pertain to specialized domains of beer reviews and restaurant ratings. For BeerAdvo-RateBeer, the $\epsilon - decreasing$ strategy outperforms the state-of-the-art semi-supervised method but slightly lags behind the supervised methods. In contrast, for the Fodors-Zagat, the $\epsilon - decreasing$ strategy matches the state-of-the-art in both supervised and semi-supervised settings.
- DbPedia-DNB and DbPedia-Viaf: Within the author's domain, these datasets hold particular significance as they showcased exceptional performance achieved by both the $\epsilon - decreasing$ strategy and the $Adaptive - \epsilon$ strategy, especially for the DbPedia-DNB, they almost matched the state-of-the-art supervised methods.

The second Table 5 focuses on the dirty datasets, specifically the WDC Phones and WDC Headphones datasets. These datasets are characterized by missing or erroneous values and, as such, provide a challenging environment for the ER. The performance breakdown for each dataset is as follows:

- WDC Phones: The $Adaptive - \epsilon$ and $\epsilon - Decreasing$ strategies outperformed the state-of-the-art semi-supervised methods and came close to the supervised methods. This is significant due to the fact that dirty data typically imposes extra challenges for the ML algorithms.
- WDC Headphones: This dataset shows similar trends as WDC Phones, with the $Adaptive - \epsilon$ and $\epsilon - Decreasing$ strategies performing comparably to state-of-the-art supervised methods.

The third Table 6 provides an analysis of the EBEES algorithm's performance on the textual datasets: Abt-Buy and Amazon-Google. The performance metrics for each dataset are outlined as follows:

- Abt-Buy: The $\epsilon - decreasing$ strategy stands out in this situation as well, surpassing the state-of-the-art semi-supervised approaches and achieving an impressive F1-score of 0.834, approaching the performance of the state-of-the-art supervised methods.

Table 4 Comparative evaluation of the ActiveML query strategies across structured datasets, benchmarked against state-of-the-art supervised (S F1) and semi-supervised F1-scores (ActiveML F1)

Database	Strategy	F1	ActiveML F1	S F1
DBLP ACM	Exploration (Density)	0.745	0.960 Wu et al. (2020)	0.989 Li et al. (2023)
	Exploitation (QBC)	0.000		
	Adaptive-ϵ	0.965		
	ϵ - **Decreasing**	**0.972**		
DBLP Google Scholar	Exploration (Density)	0.010	0.860 Wu et al. (2020)	0.956 Li et al. (2023)
	Exploitation (QBC)	0.000		
	Adaptive ϵ	0.76		
	ϵ - **Decreasing**	**0.864**		
iTunes Amazon	Exploration (Density)	0.756	0.498 Papadakis et al. (2023)	0.981 Chen et al. (2021)
	Exploitation (QBC)	0.000		(0.970 Li et al. (2023))
	Adaptive ϵ	**0.981**		
	ϵ - Decreasing	0.980		
Walmart Amazon	Exploration (Density)	0.745	0.461 Papadakis et al. (2023)	0.867 Li et al. (2023)
	Exploitation (QBC)	0.606		
	Adaptive ϵ	**0.812**		
	ϵ - Decreasing	0.765		
BeerAdvo rateBeer	Exploration (Density)	0.309	0.359 Papadakis et al. (2023)	0.946 Li and Wu (2023)
	Exploitation (QBC)	0.796		(0.943 Li et al. (2023))
	Adaptive ϵ	0.846		
	ϵ - **Decreasing**	0.857		
Fodors Zagat	Exploration (Density)	0.884	1.0 Wu et al. (2020)	1.0 Li et al. (2023); Li and Wu (2023)
	Exploitation (QBC)	0.000		
	Adaptive ϵ	0.9.27		
	ϵ-**Decreasing**	**1.0**		
dbpedia dnb	Exploration (Density)	0.699	0.722 Primpeli et al. (2020)	0.976 Primpeli et al. (2020)
	Exploitation (QBC)	0.606		
	Adaptive ϵ	**0.952**		
	ϵ - Decreasing	0.918		
dbpedia viaf	Exploration (Density)	0.028	0.862 Primpeli et al. (2020)	0.983 Primpeli et al. (2020)
	Exploitation (QBC)	0.617		
	Adaptive ϵ	0.842		
	ϵ - **Decreasing**	**0.896**		

The bold entries are essential for highlighting the highest-performing results, which serves as a visual tool to direct reader attention effectively, thereby enhancing the communicative value of the tables

Table 5 Comparative evaluation of the ActiveML query strategies across dirty datasets, benchmarked against state-of-the-art supervised (S F1) and semi-supervised F1-scores (ActiveML F1)

Database	Strategy	F1	ActiveML F1	S F1
wdc phones	Exploration (Density)	0.084	0.544 Primpeli et al. (2020)	0.851 Primpeli et al. (2020)
	Exploitation (QBC)	0.516		(0.849 Petrovski and Bizer (2020))
	Adaptive ϵ	0.783		
	ϵ - **Decreasing**	**0.792**		
wdc headphones	Exploration (Density)	0.120	0.738 Primpeli et al. (2020)	0.966 Primpeli et al. (2020)
	Exploitation (QBC)	0.908		(0.940 Petrovski and Bizer (2020))
	Adaptive ϵ	**0.951**		
	ϵ - **Decreasing**	**0.951**		

The bold entries are essential for highlighting the highest-performing results, which serves as a visual tool to direct reader attention effectively, thereby enhancing the communicative value of the tables

- Amazon-Google: Similar to Abt-Buy, the $\epsilon - decreasing$ strategy significantly outperforms traditional methods and state-of-the-art semi-supervised methods. Nevertheless, there is potential for enhancement, as it has not yet reached the performance of state-of-the-art supervised methods.

In summary, across all three categories- structured, dirty, and textual datasets-the EBEES algorithm's epsilon-decreasing strategy consistently stands out with its superior performance. It not only outperforms state-of-the-art semi-supervised methods but also often approaches the performance levels of supervised ML models. These findings strongly emphasize that

Table 6 Comparative evaluation of the ActiveML query strategies across textual datasets, benchmarked against state-of-the-art supervised (S F1) and semi-supervised F1-scores (ActiveML F1)

Database	Strategy	F1	ActiveML F1	S F1
abt buy	Exploration (Density)	0.090	0.520 Wu et al. (2020)	0.893 Li et al. (2023)
	Exploitation (QBC)	0.0329		
	Adaptive ϵ	0.824		
	ϵ - **Decreasing**	**0.834**		
amazon google	Exploration (Density)	0.006	0.480 Wu et al. (2020)	0.699 Primpeli et al. (2020)
	Exploitation (QBC)	0.0238		(0.693 Mudgal et al. (2018))
	Adaptive ϵ	0.583		
	ϵ - **Decreasing**	**0.619**		

The bold entries are essential for highlighting the highest-performing results, which serves as a visual tool to direct reader attention effectively, thereby enhancing the communicative value of the tables the EBEES algorithm is a robust and versatile tool for the ER across diverse data types and varying data quality.

7 Discussion

The EBEES algorithm offers a comprehensive solution by synergistically combining representativeness and informativeness. In this manner, it allows the model to harness the full potential of the underlying data distribution structure, resulting in a robust and versatile strategy. This is particularly advantageous in scenarios where bootstrap data are absent, given that the algorithm does not rely on pre-labeled profile pairs for the initialization. The absence of the bootstrap data usually imposes inherent limitations on the ML model, as it starts the learning process devoid of any a priori knowledge. This often leads to less informed initial queries, slower convergence, and the requirement for more iterations to reach optimal performance. Despite such constraints, EBEES exhibits remarkable resilience and adaptability across different types of datasets-be they structured, dirty, or textual. It either surpasses or closely matches the performance of state-of-the-art methods, even without the benefit of bootstrap data, and converges rapidly, thereby enhancing its efficiency across a wide array of complex datasets. The EBEES methodology facilitates dynamic modifications to the value of ϵ, which significantly improves the model's robustness and generalization capabilities. Among the adaptive strategies, the $\epsilon - decreasing$ approach stands out for its universal applicability, making it an exceptionally viable choice for a broad spectrum of real-world scenarios, even when ground truth data are limited or entirely unavailable.

8 Conclusion

In essence, the EBEES algorithm leverages a dynamic Epsilon strategy to offer a balanced, efficient, and robust solution for ActiveML. It excels in the absence of bootstrap data by employing a query strategy that combines both informativeness and representativeness. Accordingly, it achieves rapid model convergence and maintains stable performance across diverse datasets. Therefore, EBEES stands as a compelling choice for real-world scenarios where robustness, efficiency, and adaptability are required, even although when bootstrap data are scarce or entirely absent. Looking forward to future research, there are two significant avenues to delve into. The first involves the development of an intelligent stopping criteria within the ActiveML framework, a challenge that naturally complements our current work. The second future direction focuses on the incorporation of transfer learning and transformer models to enhance the versatility and efficiency of the EBEES method. These promising avenues for future research have the potential to build upon the foundational contributions of this study, providing a more robust and versatile set of tools for the entity resolution.

Author Contributions The initial concept was jointly devised by J.M. and H.I. . J.M. developed the theory and executed the computational work. The analytical methods were authenticated by H.I. and T.H., while R.Y. oversaw and reviewed the research outcomes. All authors engaged in discussions about the results and collaborated on the final manuscript.

Funding The authors did not receive any financial support for this study.

Availability of supporting data Datastes are available from github.

Code availability The code is available upon reasonable request.

Declarations

Conflict of interest Authors declare that they have no conflict of interest.

Competing interests The authors declare no competing interests.

Consent for publication All authors agree with the content and give explicit consent to submit.

References

Alexakis, T., Peppes, N., Demestichas, K., et al. (2022). A machine learning-based method for content verification in the e-commerce domain. *Information, 13*(3), 116. https://doi.org/10.3390/info13030116

Bahri, D., Jiang, H., Tay, Y., et al. (2022). Scarf: Self-supervised contrastive learning using random feature corruption. arXiv:2106.15147. https://doi.org/10.48550/ARXIV.2106.15147

Bianco, G. D., Galante, R., Goncalves, M. A., et al. (2015). A practical and effective sampling selection strategy for large scale deduplication. *IEEE Transactions on Knowledge and Data Engineering, 27*(9), 2305–2319. https://doi.org/10.1109/tkde.2015.2416734

Brunner, U., & Stockinger, K. (2020). *Entity matching with transformer architectures - a step forward in data integration*. OpenProceedings. https://doi.org/10.21256/ZHAW-19637

Chen, A., Yang, P., & Cheng, P. (2021). ACTSSD: social spammer detection based on active learning and co-training. *The Journal of Supercomputing, 78*(2), 2744–2771. https://doi.org/10.1007/s11227-021-03966-3

Chen, D., Lin, Y., Li, W., et al. (2020). Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. *Proceedings of the AAAI Conference on Artificial Intelligence, 34*(04), 3438–3445. https://doi.org/10.1609/aaai.v34i04.5747

Chen, R., Shen, Y., & Zhang, D. (2021). GNEM: A generic one-to-set neural entity matching framework. In: *Proceedings of the Web Conference 2021*. ACM. https://doi.org/10.1145/3442381.3450119

Chen, Z., Chen, Q., Hou, B., et al. (2020). Towards interpretable and learnable risk analysis for entity resolution. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. ACM. https://doi.org/10.1145/3318464.3380572

Christen, P., Vatsalan, D., & Wang, Q. (2015). Efficient entity resolution with adaptive and interactive training data selection. In: *2015 IEEE International Conference on Data Mining*. IEEE. https://doi.org/10.1109/icdm.2015.63

Christophides, V., Efthymiou, V., Palpanas, T., et al. (2021). An overview of end-to-end entity resolution for big data. *ACM Computing Surveys, 53*(6), 1–42. https://doi.org/10.1145/3418896

Dharavath, R., & Singh, A.K. (2015). *Entity Resolution-Based Jaccard Similarity Coefficient for Heterogeneous Distributed Databases* (pp. 497–507) Springer India. https://doi.org/10.1007/978-81-322-2517-1_48

Dong, X. L., & Rekatsinas, T. (2018). Data integration and machine learning. In: *Proceedings of the 2018 International Conference on Management of Data*. ACM. https://doi.org/10.1145/3183713.3197387

Fu, C., Han, X., He, J., et al. (2020). Hierarchical matching network for heterogeneous entity resolution. In: C. Bessiere (ed.) *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20* (pp. 3665–3671). International Joint Conferences on Artificial Intelligence Organization. https://doi.org/10.24963/ijcai.2020/507. main track

Getoor, L., & Machanavajjhala, A. (2012). Entity resolution. *Proceedings of the VLDB Endowment, 5*(12), 2018–2019. https://doi.org/10.14778/2367502.2367564

Hand, D., & Christen, P. (2017). Using the f-measure for evaluating record linkage algorithms. *Statistics and Computing, 28*(3), 539–547. https://doi.org/10.1007/s11222-017-9746-6

Helgertz, J., Price, J., Wellington, J., et al. (2021). A new strategy for linking U.S. historical censuses: A case study for the IPUMS multigenerational longitudinal panel. *Historical Methods: A Journal of Quantitative and Interdisciplinary History, 55*(1), 12–29. https://doi.org/10.1080/01615440.2021.1985027

Hou, B., Chenm, Q., Shen, J., et al. (2019). Gradual machine learning for entity resolution. In: *The World Wide Web Conference*. ACM. https://doi.org/10.1145/3308558.3314121

Jain, A., Sarawagi, S., & Sen, P. (2021). Deep indexed active learning for matching heterogeneous entity representations. *Proceedings of the VLDB Endowment, 15*(1), 31–45. https://doi.org/10.14778/3485450.3485455

Jaro, M. A. (1989). Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association, 84*(406), 414–420. https://doi.org/10.1080/01621459.1989.10478785

Jurek, A., Hong, J., Chi, Y., et al. (2017). A novel ensemble learning approach to unsupervised record linkage. *Information Systems, 71*, 40–54. https://doi.org/10.1016/j.is.2017.06.006

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady, 10*, 707–710. https://api.semanticscholar.org/CorpusID:60827152

Li, S., & Wu, H. (2023). Transformer-based denoising adversarial variational entity resolution. *Journal of Intelligent Information Systems*. https://doi.org/10.1007/s10844-022-00773-x

Li, Y., Li, J., Suhara, Y., et al. (2023). Effective entity matching with transformers. *The VLDB Journal*. https://doi.org/10.1007/s00778-023-00779-z

Meduri, V. V., Popa, L., Sen, P., et al. (2020). A comprehensive benchmark framework for active learning methods in entity matching. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. ACM. https://doi.org/10.1145/3318464.3380597

Mozafari, B., Sarkar, P., Franklin, M., et al. (2014). Scaling up crowd-sourcing to very large datasets. *Proceedings of the VLDB Endowment, 8*(2), 125–136. https://doi.org/10.14778/2735471.2735474

Mudgal, S., Li, H., Rekatsinas, T., et al. (2018). Deep learning for entity matching. In: *Proceedings of the 2018 International Conference on Management of Data*. ACM. https://doi.org/10.1145/3183713.3196926

Nafa, Y., Chen, Q., Chen, Z., et al. (2022). Active deep learning on entity resolution by risk sampling. *Knowledge-Based Systems, 236*, 107729. https://doi.org/10.1016/j.knosys.2021.107729

Niknam, M., Minaei-Bidgoli, B., & Dianat, R. (2021). The role of transitive closure in evaluating blocking methods for dirty entity resolution. *Journal of Intelligent Information Systems, 58*(3), 561–590. https://doi.org/10.1007/s10844-021-00676-3

Papadakis, G., Ioannou, E., Thanos, E., et al. (2021). The Four Generations of Entity Resolution. *Springer International Publishing*. https://doi.org/10.1007/978-3-031-01878-7

Papadakis, G., Kirielle, N., Christen, P., et al. (2023). A critical re-evaluation of benchmark datasets for (deep) learning-based matching algorithms. https://doi.org/10.48550/ARXIV.2307.01231

Peeters, R., & Bizer, C. (2021). Dual-objective fine-tuning of BERT for entity matching. *Proceedings of the VLDB Endowment, 14*(10), 1913–1921. https://doi.org/10.14778/3467861.3467878

Petrovski, P., & Bizer, C. (2020). Learning expressive linkage rules from sparse data. *Semantic Web, 11*(3), 549–567. https://doi.org/10.3233/sw-190356

Petrovski, P., Primpeli, A., Meusel, R., et al. (2017). The wdc gold standards for product feature extraction and matching. In: D. Bridge, H. Stuckenschmidt (eds.) *E-Commerce and Web Technologies* (pp. 73–86). Springer International Publishing. https://doi.org/10.1007/978-3-319-53676-7_6

Primpeli, A., Bizer, C., & Keuper, M. (2020). Unsupervised bootstrapping of active learning for entity resolution. In: *The Semantic Web* (pp. 215–231). Springer International Publishing. https://doi.org/10.1007/978-3-030-49461-2_13

Qian, K., Popa, L., & Sen, P. (2017). Active learning for large-scale entity resolution. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM. https://doi.org/10.1145/3132847.3132949

Reyes-Galaviz, O. F., Pedrycz, W., He, Z., et al. (2017). A supervised gradient-based learning algorithm for optimized entity resolution. *Data & Knowledge Engineering, 112*, 106–129. https://doi.org/10.1016/j.datak.2017.10.004

Sarawagi, S., & Bhamidipaty, A. (2002). Interactive deduplication using active learning. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD'02*. ACM Press. https://doi.org/10.1145/775047.775087

Settles, B. (2012). Active Learning. *Springer International Publishing*. https://doi.org/10.1007/978-3-031-01560-1

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*(3), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Somepalli, G., Goldblum, M., Schwarzschild, A., et al. (2021). Saint: Improved neural networks for tabular data via row attention and contrastive pre-training. arXiv:2106.01342. https://doi.org/10.48550/ARXIV.2106.01342

Tejada, S., Knoblock, C. A., & Minton, S. (2002). Learning domain-independent string transformation weights for high accuracy object identification. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. https://doi.org/10.1145/775047.775099

Vieira, P. K. M., Lóscio, B. F., & Salgado, A. C. (2019). Incremental entity resolution process over query results for data integration systems. *Journal of Intelligent Information Systems, 52*(2), 451–471. https://doi.org/10.1007/s10844-019-00544-1

Wu, R., Chaba, S., Sawlani, S., et al. (2020). ZeroER: Entity resolution using zero labeled examples. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. ACM. https://doi.org/10.1145/3318464.3389743

Yan, Y., Meyles, S., Haghighi, A., et al. (2020). Entity matching in the wild: A consistent and versatile framework to unify data in industrial applications. In: *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*. ACM. https://doi.org/10.1145/3318464.3386143

Zhang, D., Guo, L., He, X., et al. (2018). A graph-theoretic fusion framework for unsupervised entity resolution. In: *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE. https://doi.org/10.1109/icde.2018.00070

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.