Computing and Artificial Intelligence 2025, 3(1), 1450.
https://doi.org/10.59400/cai.v3i1.1450

Article

Pre-trained models for linking process in data washing machine

Bushra Sajid[1,*], Ahmed Abu-Halimeh[2], Nuh Jakoet[3]

[1] Department of Computer Science, The University of Arkansas at Little Rock, AR 72204, United States
[2] Department of Information Science, The University of Arkansas at Little Rock, AR 72204, United States
[3] Department of Information Quality, The University of Arkansas at Little Rock, AR 72204, United States
*** Corresponding author:** Bushra Sajid, bxsajid@ualr.edu

CITATION

Sajid B, Abu-Halimeh A, Jakoet N. Pre-trained models for linking process in data washing machine. Computing and Artificial Intelligence. 2025; 3(1): 1450. https://doi.org/10.59400/cai.v3i1.1450

ARTICLE INFO

Received: 17 June 2024
Accepted: 19 October 2024
Available online: 1 November 2024

Abstract: Entity Resolution (ER) has been investigated for decades in various domains as a fundamental task in data integration and data quality. The emerging volume of heterogeneously structured data and even unstructured data challenges traditional ER methods. This research mainly focuses on the Data Washing Machine (DWM). The DWM was developed in the NSF DART Data Life Cycle and Curation research theme, which helps to detect and correct certain types of data quality errors automatically. It also performs unsupervised entity resolution to identify duplicate records. However, it uses traditional methods that are driven by algorithmic pattern rules such as Levenshtein Edit Distances and Matrix comparators. The goal of this research is to assess the replacement of rule-based methods with machine learning and deep learning methods to improve the effectiveness of the processes using 18 sample datasets. The DWM has different processes to improve data quality, and we are currently focusing on working with the scoring and linking processes. To integrate the machine model into the DWM, different pre-trained models were tested to find the one that helps to produce accurate vectors that can be used to calculate the similarity between the records. After trying different pre-trained models, distilroberta was chosen to get the embeddings, and cosine similarity metrics were later used to get the similarity scores, which helped us assess the machine learning model into DWM and gave us closer results to what the scoring matrix is giving. The model performed well and gave closer results overall, and the reason can be that it helped to pick up the important features and helped at the entity matching process.

Keywords: data quality; data curation; unsupervised machine learning; entity resolution and data linking

1. Introduction

Data integration plays a vital role in maximizing data quality. Different field sectors collect data from multiple sources, like health data, which contains information on electronic health references, genomics, toxicology, and drug databases, which are essential for the advances of precision medicine [1]. To deal with entity resolution (ER), which is a critical challenge in data integration because ER helps us find whether two references in an information system belong to the same or different objects [2]. From the perspective of the ER, the entity references of the data consist of a series of values for a person's identity attributes. If we have an example of the information for a particular person, then the person's name, date of birth, address, and SSN would be the attributes. So, considering this, the ER system takes those entities and the information like the entity references (ER) and decides linking based on how comparable the information of two references is about the same person [3]. There are many ways to describe ER, but entity resolution is also known as reference or record

linkage [4], reference matching [5], deduplication [6], and so on. There are many ER methods to deal with uncleaned data, and they have numerous applications.

Besides dealing with uncleaned data and improving the integrity and quality of data, ER has been applied to crime detection, national security, and fraud [7]. ER is usually seen as an ETL (extraction, transformation, and loading) tool component [8] because of how it plays its role in data integration. ER uses various methods like probabilistic matching [9], rule-based matching [10], and machine and deep learning [11].

After diving through different models like Word2Vec, GloVe, and BERT, the focus was entirely on the BERT model because BERT utilizes the Transformer encoder as its bidirectional framework. In the Transformer encoder, positional embeddings are typically added to each position in the input sequence. Unlike the original Transformer encoder, however, BERT employs learnable positional embeddings. The BERT input sequence embeddings comprise the combined token embeddings, segment embeddings, and positional embeddings. Several natural language processing (NLP) researchers prefer pre-trained BERT models over ones developed from scratch because of their numerous essential benefits. Pre-trained BERT models, like the ones created by Google, can acquire rich, contextualized word representations that capture intricate linguistic nuances since they have already been trained on large corpora, such as Wikipedia and Book Corpus, and that is what is needed in this use case. Resource efficiency is one of the main advantages of utilizing these models. It can take weeks to fully train a BERT model from scratch, which is computationally costly and time-consuming. It also requires a lot of hardware. On the other hand, researchers with limited resources can still fine-tune a pre-trained BERT model on a particular job with relative speed and substantially reduced processing demands [12].

Moreover, the data requirements for training BERT from scratch are significant, often requiring billions of words to learn effective representations. Pre-trained models relieve this burden as they have already been exposed to large datasets, allowing researchers to fine-tune them with smaller, task-specific datasets while still achieving excellent performance. This transferability is a key advantage, as BERT's architecture supports transfer learning, where knowledge acquired during pre-training is effectively applied to new tasks or domains [13]. This capability is particularly beneficial in specialized domains where annotated data may be scarce. Pre-trained BERT models have consistently demonstrated superior performance across a variety of NLP tasks, such as sentiment analysis, named entity recognition, and question answering, often outperforming models trained from scratch on limited data [14].

Since most applications require data integration from multiple sources, ER has become increasingly influential in today's big data world. Entity matching relies on extracting and comparing entity features, essential for tasks like knowledge management, data integration, and information retrieval. By encoding rich, context-aware representations of entities, which BERT captures, traditional techniques typically miss nuances in meaning and usage, thereby transforming this process. A more sophisticated comparison is made possible by this capacity to comprehend the context in which entities exist, which lowers the number of false matches and increases

the recall of genuine matches [12].

This research starts by evaluating pre-trained models, particularly BERT and its variants, alongside newer large language models, to explore their potential in the entity resolution process within the Data Washing Machine (DWM). By integrating these deep learning models into the unsupervised DWM process, the research aims to improve the clustering accuracy by addressing syntactic and semantic similarity issues. The use of attention mechanisms helps derive reference embeddings based on similarity score vectors, which are crucial for comparing entity records. Additionally, machine learning techniques are employed to compare the results with the scores generated by the data washing machine. Current ER approaches rely heavily on rule-based methods, and this study aims to introduce a novel method for handling data in the entity linking process of the DWM.

By utilizing the bidirectional framework of the BERT model, this study aims to enhance entity resolution by capturing rich, contextualized representations of entities, leading to improved matching accuracy. Pre-trained models like BERT, which have been trained on large corpora, are used to save computational resources and time [12]. These models offer resource efficiency, as they can be fine-tuned for specific tasks without requiring extensive training from scratch [13]. BERT's transfer learning capabilities allow it to adapt to various tasks, including those in specialized domains where annotated data may be scarce, making it an ideal candidate for entity resolution tasks in this study [13,14].

The rest of this paper is organized as follows: The Related Work section presents related and recent work, focusing on existing models like BERT and their applications in entity resolution. It also mentions DWM ER. The Method section outlines the proposed methodology, detailing how BERT embeddings are incorporated into the Data Washing Machine process and its effects on the entity resolution accuracy, and it also describes the experimental setup and presents results from applying the proposed method to various datasets with different data quality levels. The discussion section discusses the findings, comparing the performance of the new approach with traditional rule-based methods. Conclusion and future work conclude the paper, summarizing contributions and potential directions for future research.

2. Related work

A scalable approach, OYSTER, a supervised ER system for clustering equivalent references [15], helped to do the linking process along with blocking and clustering. This work uses frequency-based blocking and stop-word removal. The scalable implementation embraced in this research involved two main preprocessing phases. First, the frequency of each of the tokens is calculated, and second, all excluded blocking tokens and stop words are eliminated, leaving a skinny reference pair that will be compared for similarity using the scoring matrix.

The Data Washing Machine helps cover all the entity resolution steps by going through different processes, which are rule-based methods to solve error correction problems and improve data quality. The DWM was developed in the NSF DART Data Life Cycle and Curation research theme, which helps to detect and correct certain types of data quality errors automatically. **Figure 1** shows how the data washing

machine goes through the whole process. It uses ER as the first step using unsupervised blocking and stop word schemes based on token frequency. A variant of the Monge-Elkan comparator, a scoring matrix, is used to link unstandardized references. Linking is followed by an unsupervised process for evaluating the quality of the linking results based on a variation of Shannon entropy.

The DWM ER process is iterative, and the reference similarity threshold is increased in each iteration. The prototype was tested on 18 fully annotated test samples of primarily synthetic person data, which varied in two ways: good data quality versus poor data quality and a single record layout versus two different record layouts. The results demonstrated the feasibility of building an unsupervised ER engine to support data integration for good quality references while avoiding the time and effort to standardize reference sources to a standard layout and to design and test matching rules, design blocking keys, or test blocking alignment.

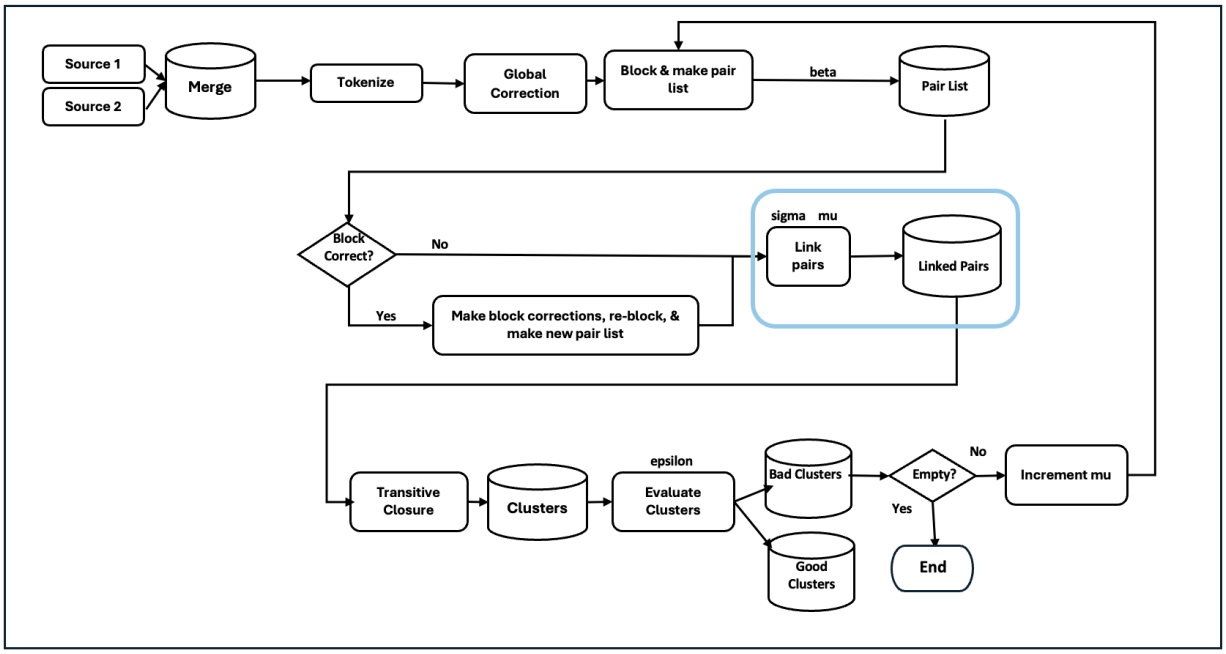

Figure 1. Flowchart of the data washing machine.

Embeddings have become a cornerstone in modern NLP applications, enabling more sophisticated and nuanced language models that can handle various complex tasks such as sentiment analysis, machine translation, and question-answering systems. This shift from sparse to dense representations reflects broader trends in NLP towards models that better approximate human language understanding [16].

Then, an unsupervised mechanism for estimating the optimal parameters is used in the DWM.

2.1. Probabilistic approach

In 1969, Fellegi-Sunter introduced the probabilistic technique. It calculates weights for various attributes based on their projected accuracy in distinguishing between matching and non-matching references. The weights determine the likelihood that two specified references correspond to the same entity [1].

The likelihood of a reference pair matching is determined by the similarity measure of the two references, quantified by a similarity value ranging from 0 to 1. A value of 0 signifies no similarity between the references, while 1 shows complete similarity. Various similarity measures are available for selection, such as Jaro [17], Jaro-Winkler [18], Jaccard [19], Q-Gram [20], and Levenshtein edit distance [21]. Various similarity metrics are appropriate for different data types, and no universal similarity measure applies to all forms of references. Elmagarmid et al. [22] categorize similarity metrics based on the level at which the comparison is made. The similarity measure methods mentioned above make reference-matching judgments using the probabilistic matching model. This model utilizes a Bayesian approach to categorize reference pairs into two classes: match (M) and non-match (N).

2.2. Machine learning approach

Typically, machine learning treats the pairwise-based entity resolution problem as a binary classification task. The learning-based technique can be implemented using supervised, semi-supervised, and unsupervised learning methods.

Supervised learning involves learning an ER model using training data and using the model to categorize new reference pairs. The training dataset is structured as follows: {(r1,1, r1,2, l1), (r2,1, r2,2, l2), ... (rn,1, rn,2, ln)}, where each data sample includes a reference pair (ri, rj) and a label with a value of 0 or 1, denoting whether the two references match or not. The binary classifier can be trained using several learning models with the training data. Bilenko and Mooney [6] utilized machine learning to acquire similarity estimates for text references instead of manually adjusting the similarity measures.

Effective implementation of supervised approaches relies on having a substantial amount of precisely labeled training data, which might need to be more readily available in practical scenarios, resulting in costly and labor-intensive entity recognition.

In ER research, semi-supervised and unsupervised learning methods are offered to tackle the issue of insufficient training data. Jurek-Loughrey and Deepak [1] comprehensively overview the current semi-supervised and unsupervised approaches. Semi-supervised learning involves using a limited number of labeled reference pairs during training. Unsupervised learning does not require labeled reference pairs to create the decision model.

Hou et al. [23] introduced a new machine-learning approach for entity recognition (ER) termed progressive machine learning. This method is designed to facilitate accurate machine labeling without manual labeling. The process starts with simple examples of tasks that the machine can accurately label automatically. It then progresses to labeling more difficult examples using iterative factor graph inference. Gradual machine learning involves incrementally labeling complex cases in a task in modest increments, guided by the estimated evidential certainty obtained by labeling easier instances. Prior work had been done in Al Sarkhi and Talburt's research [24,25] for a matrix comparator for linking equivalent references. The scoring matrix comparator underwent further development, which improved its capabilities and allowed it to do linking using conventional techniques. The unsupervised technique

suggested by Zeakis et al. [26] relies heavily on the embeddings and tries to find matching entities in a variety of data kinds, including text sentences and relational tuples. This adaptability enables the framework to function independently of previously collected labeled data, allowing it to be used in many domains.

Incorporating sentence similarity through the NLP tasks described by Ahmadi N et al. [27] improved the entity resolution processes and data cleaning capabilities. Several studies show embeddings generated by Transformer-based models to enhance the linking process in entity matching (EM), utilizing pre-trained models like BERT and RoBERTa, which are fine-tuned to improve matching quality significantly [28]. Embeddings can provide a more nuanced representation of the attributes of records. For instance, textual attributes can be transformed into vector representations that capture semantic meanings, which can enhance the model's ability to distinguish between matches and unmatches. This aligns with the paper's emphasis on the importance of similarity vectors in the entity resolution process [29].

3. Problem statement

One of the key challenges in entity resolution (ER) systems is the processing of multiple reference sources, each potentially in a different standardized format. Even when individual reference sources are already in a standardized format, they often differ from each other. This necessitates an additional preprocessing step in the ER process to transform each source into the standard the ER system expects. For instance, if the ER system is designed to compare person first names and last names as separate fields, a reference source where both names are in a single field must be pre-processed and reformatted to separate and properly classify the name words.

The clustering is used in a master data management system in the Data Washing Machine. It is now apparent that some records should have matched or should have been like each other. We may get into situations where the records match because of the SSN but not the address. So, there is a need for data to quantify and compare the data washing machine link matching and scoring with deep learning matching algorithms to see how different the results are. This research aims to explore designs for the matching process that operate effectively and help get better results in linking pairs of heterogeneously structured references.

4. Dataset

There are 18 sample data files with more than 10,000 samples to test our model. These datasets all have associated truth sets (annotations) that allow the user to check the accuracy of the clustering for a given set of parameter settings. Each dataset came with annotated truth sets, as depicted in **Table 1**, allowing for the validation of clustering accuracy under distinct parameter configurations. **Table 1** shows a comprehensive summary of the attributes of each testing dataset, such as file name, size, data characteristics, quality assessment, layout, and corresponding truth file. The datasets differed in size, ranging from 50 to 19,998 entries, and encompassed various data types like a person's name, addresses, and SSNs. Quality evaluations were given for every dataset, classified into "good" or "poor," with relevant truth files for examination. The linking process performance was assessed using precision, recall,

and F-measure metrics computed based on the truth file names specified under the "truth filename" parameter.

Table 1. Datasets used for data washing machine.

File Name	Size	Characteristics	Quality	Layout	Truth File Name
S1G.txt	50	Person name & address	Good	Single	truthABCgoodDQ.txt
S2G.txt	100	Person name & address	Good	Single	truthABCgoodDQ.txt
S3Rest.txt	868	Business name & address	Good	Single	truthRestaurant.txt
S4G.txt	1912	Person name & address	Good	Single	truthABCgoodDQ.txt
S5G.txt	3004	Person name & address	Good	Single	truthABCgoodDQ.txt
S6GeCo.txt	19,998	Person name & address	Good	Single	truthGeCo.txt
S7GX.txt	2912	Person name & address	Good	Mixed	truthABCgoodDQ.txt
S8P.txt	1000	Person name & address	Poor	Single	truthABCpoorDQ.txt
S9P.txt	1000	Person name & address	Poor	Single	truthABCpoorDQ.txt
S10PX.txt	2000	Person name & address	Poor	Mixed	truthABCpoorDQ.txt
S11PX.txt	3999	Person name & address	Poor	Mixed	truthABCpoorDQ.txt
S12PX.txt	6000	Person name & address	Poor	Mixed	truthABCpoorDQ.txt
S13GX.txt	2000	Person name & address	Good	Mixed	truthABCgoodDQ.txt
S14GX.txt	5000	Person name & address	Good	Mixed	truthABCgoodDQ.txt
S15GX.txt	10,000	Person name & address	Good	Mixed	truthABCgoodDQ.txt
S16PX.txt	2000	Person name & address	Poor	Mixed	truthABCpoorDQ.txt
S17PX.txt	5000	Person name & address	Poor	Mixed	truthABCpoorDQ.txt
S18PX.txt	10,000	Person name & address	Poor	Mixed	truthABCpoorDQ.txt

The data to train our model is synthetic data which mimic real-world people's reference. The following **Table 2** is an example of the dataset.

Table 2. Reference examples in dataset file (S1G.txt).

Reference ID	Reference
C787384	IAN AADLAND, LARS, 7715 ABINGTON DR, KERNERSVILLE, NC 27284, (361)-924-5829,1911/8/25
B996789	IAAN LARS AADLAND, 7715, ABINGTON DR, KERNERSVILLE, NC, 27284, 490-46-2048, 1911825
C787387	AANAI, HIKARI, F, 2165 MAURINE WAY, WINSTON SALEM, NC 27127, (483)-549-7645,
C787385	Kavassana Aanai, Hikari, F, 2165 MAURINE WAY, WINSTON SALEM, NC 27127, (483)-549-7645,1906/4/6
C787386	KAVASSANA AANAI, H, F, 2165 MAURINE WAY, WINSTON SALEM, NC 27127, (483)-549-7645,
A939042	JUDY, AANSTAD, 221 HARMON CT, WINSTON SALEM, NC, 27106, 555374439, (247)-793-3157

5. The method

When we run our raw dataset through a data washing machine, it goes through tokenization processes, calculating the frequencies of those tokens, error correction, and deduplication, and provides us with the pairs in the linking process. The pairs the machine gets after the blocking process used in the scoring matrix help to determine whether the machine will link them. Those same pairs are considered for assessing and using the machine learning method. After extracting their references, vectorization is done to continue the linking process by using cosine similarity.

When given a raw score vector input instead of a statistical characteristic input, neural networks perform better. We need to convert the token sequence into a sequence of numerical values because deep learning algorithms are unable to read the token sequence. While using the same algorithm, several transformation techniques may yield different outcomes. To find the most effective vectorization techniques for our design, state-of-the-art word embedding was examined.

DistilRoBERTa [30] outperformed all other pre-trained models following the application of several pre-trained models and analysis. A transformer-based language model called DistilRoBERTa was trained using a condensed variant of the BERT model's text. Although its primary applications are in text classification and question answering, we can take advantage of its powers to support entity resolution. When evaluating the integration of machine learning into the DWM (**Figure 2**), all other processes were running as exactly as they needed to run except the linking process; it seemed best to take linked pairs as a parameter and calculate the cosine similarity between them using the "all-distilroberta-v1" model.

DistilRoBERTa helps with vectorization by encoding textual data into dense vectors.

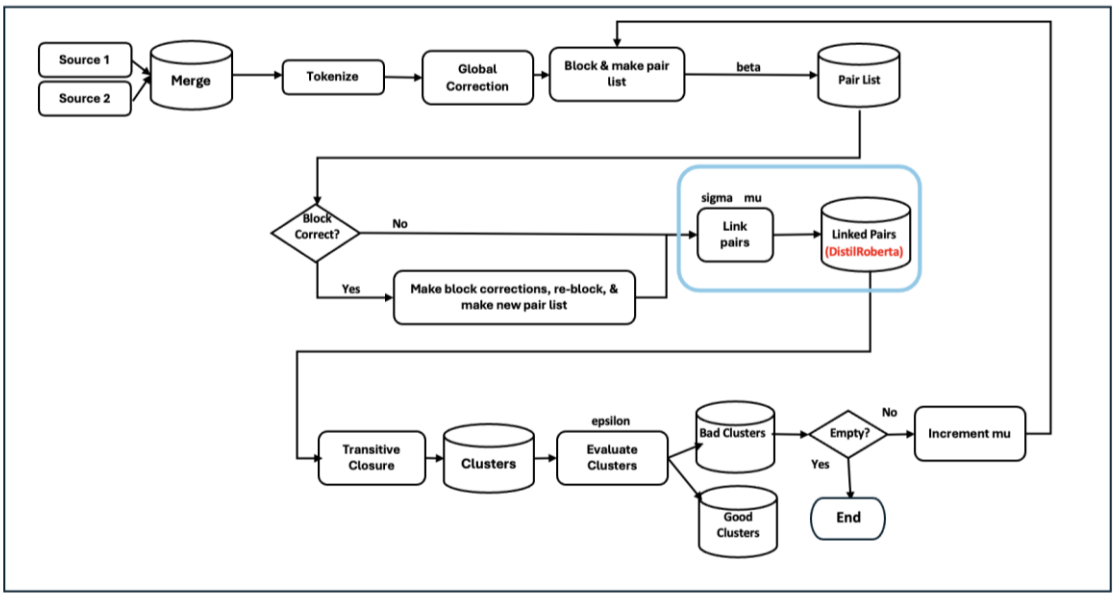

Figure 2. Flowchart of the addition to the data washing machine.

In the data washing machine, the blocking process uses the parameter beta, which represents the maximum frequency of a token that is considered a blocking token.

Shared single or shared double-blocking tokens help to create the pair list. In our method, the pair list gets accessed, and the references from that pair list are extracted. Thus, **Figure 3** shows that after extracting the references, the pre-trained model generates the vector embeddings because the machine learning model can only understand vectors instead of text. By using those vectors. We calculate the similarity score for each pair that the blocking process created.

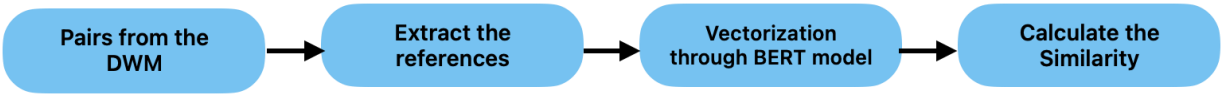

Figure 3. Flowchart of our method.

Since references often contain unique tokens or errors like typos in ER, the pre-trained word embedding does not necessarily cover all the tokens. Thus, we use the word vector for our specific pairwise matching task. The result embedding should place matched references and non-matched pairs close together in the vector space.

6. Results

After taking the linked pairs, doing the vectorization through the DistilRoBERTa (paper reference) model, and using those vectors to calculate the similarity, we found the results mentioned in **Table 3**. The results show the precision, recall, and F-measures of the DWM, where Mu represents the match threshold for linking two linked pairs each time in a datafile. The mu value must be a decimal value between 0.0 and 1.0.

Below are scores that were collected from the Data Washing Machine process while using a Levenshtein Edit Distance and the scoring matrix formula for the linking process.

Table 3. DWM linking results.

SAMPLE	PRECISION	RECALL	F-MEAS	MU
S1G.txt	1	1	1	0.6
S2G.txt	0.9231	1	0.96	0.7
S3Rest.txt	0.9043	0.9286	0.9163	0.67
S4G.txt	0.963	0.8939	0.9272	0.73
S5G.txt	0.9162	0.8879	0.9018	0.73
S6GeCo.txt	0.9812	0.9254	0.9254	0.82
S7GX.txt	0.9725	0.8678	0.9172	0.82
S8P.txt	0.7708	0.8495	0.8082	0.67
S9P.txt	0.8265	0.7226	0.7711	0.72
S10PX.txt	0.868	0.718	0.7859	0.74
S11PX.txt	0.8581	0.7384	0.7938	0.74
S12PX.txt	0.9009	0.7232	0.8023	0.73
S13GX.txt	0.9004	0.8861	0.8932	0.81
S14GX.txt	0.9103	0.8843	0.8971	0.81
S15GX.txt	0.9301	0.8568	0.8919	0.83

Table 3. (*Continued*).

SAMPLE	PRECISION	RECALL	F-MEAS	MU
S16PX.txt	0.7706	0.7731	0.7718	0.71
S17PX.txt	0.8228	0.725	0.7708	0.73
S18PX.txt	0.8194	0.7368	0.7759	0.73

The results below are collected using the pre-trained model to assess how machine learning would help find the pair's similarity. So, instead of using Levenshtein and the scoring matrix algorithm, the pre-trained model was used to convert the pairs, a text, into vectors that a model can understand. Then the cosine similarity formula was applied to those vectors to get the matching scores. Keeping the other steps, which include tokenization, blocking, and clustering, the same but only focused on linking, which is the primary process of the data washing machine.

Table 4. Linking results through ML.

SAMPLE	PRECISION	RECALL	F-MEAS	MU
S1G.txt	1	0.963	0.9812	0.87
S2G.txt	0.9333	0.875	0.9032	0.8
S3Rest.txt	0.9074	0.875	0.8909	0.7
S4G.txt	0.9207	0.798	0.855	0.85
S5G.txt	0.8741	0.8421	0.8578	0.8
S6GeCo.txt	0.7018	0.7368	0.7189	0.82
S7GX.txt	0.8134	0.8495	0.8311	0.82
S8P.txt	0.5562	0.5813	0.5685	0.65
S9P.txt	0.7544	0.3671	0.4199	0.78
S9P.txt	0.5815	0.2725	0.3711	0.72
S10PX.txt	0.5997	0.2856	0.3869	0.74
S11PX.txt	0.8262	0.2163	0.3428	0.8
S12PX.txt	0.7829	0.4196	0.5539	0.73
S13GX.txt	0.9004	0.8861	0.8932	0.81
S14GX.txt	0.6467	0.8744	0.7435	0.81
S15GX.txt	0.8688	0.7893	0.8271	0.83
S16PX.txt	0.6788	0.528	0.4187	0.71
S17PX.txt	0.6986	0.5818	0.4016	0.73
S18PX.txt	0.7549	0.4249	0.5543	0.73

As shown in **Table 4**, adding a Machine learning method results in almost the same results as we are getting in the DWM using the scoring matrix.

7. Discussion

The primary objective of this research was to assess the potential of using machine learning models within the Data Washing Machine (DWM) framework [31], particularly for the linking process. Previous entity resolution procedures for data cleaning, such as tokenization, blocking, linking, and clustering, have previously

shown strong performance. We had to limit our attention to the linking process in order to determine whether ML might outperform these conventional methods in terms of accuracy. One major obstacle to integrating ER in data integration is effectively managing uncleaned data. Traditional approaches frequently miss nuances in meaning and application, which makes it more difficult to remember accurate matches and more likely to recall false ones.

By integrating deep learning models into the process, we intend to increase cluster accuracy while assisting in overcoming syntactic and semantic similarity limitations. Machine learning models are highly versatile and can identify patterns or relationships that traditional methods would overlook, especially in datasets with varied topologies. Evaluation of pre-trained models, including BERT and its variants, is the aim of this work to improve cluster accuracy in an unsupervised data-washing machine process. Currently, DWM uses the Scoring Matrix technique for the linking process, which has been shown to improve data quality. But we want to assess if ML can make little improvements, and while this method performs well generally, we hypothesized that more improvements could be achieved by using machine learning, particularly when dealing with complex or ambiguous data records.

Although DWM's traditional methods now do most linking processes as effectively as they should, there is always room for improvement. The primary objective of this research is to ascertain the extent to which machine learning could improve results by providing a deeper understanding of the ways in which the linking process in DWM might be executed differently.

When missing data, like some missing records, presents a problem for traditional methods, machine learning can be useful in this case. It can facilitate the linking process without adding more computational complexity to the system. Furthermore, our experiment's results (**Table 3**) show that machine learning models can be used to selectively represent linkage accuracy, especially for poor datasets.

Incorporating pre-trained models into the traditional Entity Resolution (ER) tool, such as the Data Washing Machine (DWM), introduced an alternative and more flexible approach to the linking process. Unlike traditional methods that rely heavily on rule-based techniques—such as tokenization, blocking, and clustering—pre-trained models bring a more dynamic capability to detect patterns and relationships in data. These models, which have been trained on large datasets, can capture nuanced relationships between entities that may not be immediately apparent through deterministic methods. This makes it possible to implement a more complex linking mechanism, which is particularly useful when working with loud, complex, or lacking data.

To enhance the connection process, pre-trained models leveraged learned representations from enormous amounts of historical data [26]. Conventional ER methods sometimes have preset rules and matching requirements that restrict connection. When previously overlooked new patterns or minute variations in the data surface, these approaches might not be effective. By incorporating trained models, the ER tool offers a more robust and flexible linking mechanism that dynamically adjusts its connecting criteria based on contextual data obtained from previous datasets. This is especially helpful in cases where the data contains ambiguity or inconsistencies because the trained models can make more informed decisions by linking those

records better.

So, **Table 2** represents the linking process results when we use the scoring matrix for the linking process in the DWM, and **Table 3** represents the results when we apply the machine learning model in the DWM. As you can see, machine learning has performed well in our case, and the results are closer to what traditional methods represent. Our future plan would be to improve the linking process evaluation metrics using machine learning models.

Furthermore, the use of pre-trained models in DWM provided an alternative linking solution that was not only more accurate but also more scalable. Instead of relying solely on traditional rule-based methods, pre-trained models allowed the ER tool to perform linking in a more probabilistic and data-driven manner. This approach reduced the reliance on manually curated rules and thresholds, streamlining the overall process while improving the tool's ability to handle complex entity matching scenarios. As a result, DWM's data quality improved significantly, especially in scenarios where traditional methods would struggle, such as when dealing with missing or partial data.

8. Conclusion

Our main goal was to assess how including machine learning in the data washing machine could affect the clusters' accuracy and how adding the pre-trained models to vectorize and calculate the similarity would help us get similar results. The model then does take time to load, which makes the runtime a little bit more than the scoring matrix, Kris. Still, we can't deny that the scoring matrix is based on matrix calculations, which helps to make it run faster. However, working with a model is slightly different because all these models are trained on billions of parameters.

The BERT model gave some similar and some different results overall, and the reason can be that it helped to pick up the essential features and helped with the entity matching process. To further improve the model, training with more diverse data, including low-quality datasets with many spelling errors and null values, might be a potential direction.

9. Future work

This project focuses on the entity-matching process. The first goal was to assess using the pre-trained machine learning model in a data washing machine. In the future, we want to integrate our model into the data washing machine, add more steps to make it run successfully, and see how to improve the current tool. We have a list of Python libraries and large language models to apply further to do entity matching and compare our results with our old results and Data Washing Machine results. Our main goal is to fine-tune the models rather than build the model from scratch. During the assessment, we decided to keep comparing different methods and the newest models for the linking process to see what significant difference we can find in the accuracy of the DWM.

Author contributions: Conceptualization, BS, AAH and NJ; methodology, BS, AAH and NJ; software, BS and NJ; validation, BS, AAH and NJ; formal analysis, BS and

AAH; investigation, BS; resources, BS and AAH; data curation, BS; writing—original draft preparation, BS; writing—review and editing, BS and AAH; visualization, BS and NJ; supervision, AAH; project administration, AAH; funding acquisition, AAH. All authors have read and agreed to the published version of the manuscript.

Conflict of interest: The authors declare no conflict of interest.

Reference

1. Jurek-Loughrey A, P. D. Semi-supervised and unsupervised approaches to reference pairs classification in multi-source data linkage. In: Linking and Mining Heterogeneous and Multi-view Data. Springer, Cham; 2019. pp. 55-78.
2. Talburt JR. Entity Resolution Models. Entity Resolution and Information Quality. 2011: 63-101. doi: 10.1016/b978-0-12-381972-7.00003-8
3. Wang P, Pullen D, Wu N, Talburt JR. Iterative approach to weight calculation in probabilistic entity resolution. In: Proceedings: International Conference on Information Quality (ICIQ-19); August 2014; Xi'an, China. pp. 245-258.
4. Bhattacharya I, Getoor L. Iterative record linkage for cleaning and integration. Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. 2004: 11-18. doi: 10.1145/1008694.1008697
5. Pasula H, Marthi B, Milch B, et al. Identity uncertainty and citation matching. Available online: https://papers.nips.cc/paper_files/paper/2002/hash/d30960ce77e83d896503d43ba249caf7-Abstract.html (accessed on 3 May 2024).
6. Bilenko M, Mooney RJ. Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining; 2003. doi: 10.1145/956750.956759
7. Jonas J, Jim H. Effective counterterrorism and the limited role of predictive data mining. Washington DC: Cato Institute; 2006.
8. Christen P. Data Matching. Springer Berlin Heidelberg; 2012. doi: 10.1007/978-3-642-31164-2
9. Wang, P., Pullen, D., Talburt, J. R., Chen, C., “A method for match key blocking in probabilistic matching”, In Information Technology: New Generations, pp. 847-857, 2016.
10. Li, Lingli, Jianzhong Li, and Hong Gao. "Rule-based method for entity resolution." IEEE Transactions on Knowledge and Data Engineering 27, no. 1 (2015): 250-263.
11. Hou, Boyi, Qun Chen, Jiquan Shen, Xin Liu, Ping Zhong, Yanyan Wang, Zhaoqiang Chen, and Zhanhuai Li. "Gradual machine learning for entity resolution." In The World Wide Web Conference, pp. 3526-3530. ACM, 2019.
12. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. North American Chapter of the Association for Computational Linguistics. 2018.
13. Ruder S, Peters ME, Swayamdipta S, et al. Transfer Learning in Natural Language Processing. In: Proceedings of the 2019 Conference of the North; 2019. doi: 10.18653/v1/n19-5004
14. Wolf T, Debut L, Sanh V, et al. Transformers: State-of-the-Art Natural Language Processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations; 2020. doi: 10.18653/v1/2020.emnlp-demos.6
15. Al Sarkhi A, Talburt J. A scalable, hybrid entity resolution process for unstandardized entity references. J. Comput. Sci. Coll.2020; 35: 19-29.
16. Manning CD, Raghavan P, and Schütze H. Introduction to information retrieval. Cambridge University Press; 2008.
17. Jaro MA. Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American Statistical Association. 1989; 84(406): 414-420. doi: 10.1080/01621459.1989.10478785
18. William EW. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Reference Linkage. Available online: https://eric.ed.gov/?id=ED325505 (accessed on 3 May 2024).
19. Jaccard P. Distribution of alpine flora in the Dranses basin and some neighbouring regions (French). Bulletin de la Societe Vaudoise des Sciences Naturelles. 1901; 37(140): 241-72. doi:10.5169/seals-266440
20. Shannon CE. A Mathematical Theory of Communication. Bell System Technical Journal. 1948; 27(3): 379-423. doi: 10.1002/j.1538-7305.1948.tb01338.x
21. Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady. 1966; 10(8): 707-710.
22. Elmagarmid AK, Ipeirotis PG, Verykios VS. Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and

Data Engineering. 2007; 19(1): 1-16. doi: 10.1109/tkde.2007.250581
23. Hou B, Chen Q, Shen J, et al. Gradual Machine Learning for Entity Resolution. In: Proceedings of the World Wide Web Conference; 2019. doi: 10.1145/3308558.3314121
24. Al Sarkhi A, Talburt J. A scalable, hybrid entity resolution process for unstandardized entity references. J. Comput. Sci. Coll. 2020; 35: 19-29.
25. Al Sarkhi A, Talburt JR. Estimating the parameters for linking unstandardized references with the matrix comparator. J. Inform. Technol. Manag. 2018; 10: 12-26.
26. Zeakis A, Papadakis G, Skoutas D, et al. Pre-Trained Embeddings for Entity Resolution: An Experimental Analysis. In: Proceedings of the VLDB Endowment; 2023; pp. 2225-2238. doi: 10.14778/3598581.3598594
27. Ahmadi N, Sand H, Papotti P. Unsupervised Matching of Data and Text. In: Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE); 2022; pp. 1058-1070. doi: 10.1109/icde53745.2022.00084
28. Li Y, Li J, Suhara Y, et al. Effective entity matching with transformers. The VLDB Journal. 2023; 32(6): 1215-1235. doi: 10.1007/s00778-023-00779-z
29. Wu R, Chaba S, Sawlani S, et al. ZeroER: Entity Resolution using Zero Labeled Examples. In: Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data; 2020. doi: 10.1145/3318464.3389743
30. Sanh V, Debut L, Chaumond J, Wolf T. Distilbert, a distilled version of Bert: Smaller, faster, cheaper and lighter. Available online: https://arxiv.org/abs/1910.01108 (accessed on 3 May 2024).
31. Talburt JR, Al Sarkhi AK. (n.d.). An Iterative, Self-Assessing Entity Resolution System: First Steps toward a Data Washing Machine. Available online: https://par.nsf.gov/servlets/purl/10219479 (accessed on 3 May 2024).