Global Journal of Computer Science and Technology: D
Neural & Artificial Intelligence
Volume 23 Issue 1 Version 1.0 Year 2023
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Entity Matching for Digital World: A Modern Approach using Artificial Intelligence and Machine Learning

By K. Victor Rajan & Edward Lambert

Atlantic International University

Abstract- Entity matching is the field of research solving the problem of identifying similar records which refer to the same real-world entity. In today's digital world, business organizations deal with large amount of data like customers, vendors, manufacturers, etc. Entities are spread across various data sources and failure to correlate two records as one entity can lead to confusion. Relationships and patterns would be missed. Aggregations and calculations won't make any sense. It is a significant data integration effort that often arises when data originate from different sources. In such scenarios, we understand the situation by linking records and then track entities from a person to a product, etc. There is appreciable value in integrating the data silos across various industries.

Keywords: *entity matching, entity resolution, record linkage, de-duplication, machine learning.*

GJCST-D Classification: *FOR Code: 170203*

Strictly as per the compliance and regulations of:

Entity Matching for Digital World: A Modern Approach using Artificial Intelligence and Machine Learning

K. Victor Rajan [α] & Edward Lambert [σ]

Abstract- Entity matching is the field of research solving the problem of identifying similar records which refer to the same real-world entity. In today's digital world, business organizations deal with large amount of data like customers, vendors, manufacturers, etc. Entities are spread across various data sources and failure to correlate two records as one entity can lead to confusion. Relationships and patterns would be missed. Aggregations and calculations won't make any sense. It is a significant data integration effort that often arises when data originate from different sources. In such scenarios, we understand the situation by linking records and then track entities from a person to a product, etc. There is appreciable value in integrating the data silos across various industries. For example, if a customer record is listed multiple times with purchases across two different store databases due to different spellings of name or a typing error in the phone number, a duplicate email from the company would only be a missed sale opportunity, or worse could cause the customer to mark the company's marketing emails as spam. Organizations put huge effort in deduplication through record matching as it really helps in operational excellence. Traditional approaches include manually checking the names, phone numbers etc., and marking duplicate entries. This is time consuming and might involve huge labor cost. Often computer software is used to automate this process by comparing the attributes and applying rule-based techniques. Traditional programs can assess equality and perform mathematical comparisons but cannot understand fuzzy matching on their own. In this research paper, we present how Artificial Intelligence and Machine Learning (AI/ML) can be used for entity matching and propose a taxonomy of machine learning algorithms for entity matching. We propose a two-step methodology for entity matching. In the first step, we apply fuzzy matching techniques and generate feature vector of similarity scores. This produces a data set of feature vectors under two classes namely Duplicate (D) and Unique (U). In the second step, a well-trained machine learning model tries to predict the unknown real-world entities using supervised learning. Our experiments show highly accurate results and can be used in many practical use cases like customers de-duplication across e-commerce sites, retail stores, hospitals, pharmacies, etc.

Keywords: entity matching, entity resolution, record linkage, de-duplication, machine learning.

Author α σ: Department of Computer Engineering, Atlantic International University, Hawaii, 96813. USA. e-mail: victor@jts.co.in

I. Introduction

Our world is moving towards digitized business. This opens up numerous avenues to increase revenue through digital marketing, sales forecast, etc. Huge amount of historical data is available to analyze customer behavior, buying patterns and make predictions for future. However, it also comes with challenges along the way. A substantial amount of the value to be harvested from digitization depends on successful integration of large volume of data from different sources. Unfortunately, many of the existing data sources do not share a common frame of reference. For example, let us say, a marketing team wants to use statistics from retail stores, e-commerce sites etc., to find out potential buyers for a product. Sadly, these two systems do not refer to customers in the same way – i.e., there are no common identifiers or names across the two systems. Duplicate emails or messages may be sent to same customer again and again unless customer records are tagged uniquely. Recommendations to a customer and an effective marketing scheme cannot be performed based on distinct data silos. A group of similar problems has been studied for a long time in a variety of fields under different names like entity resolution, de-duplication etc. Entity matching is the field of research dedicated to solving the problem of matching which records refer to the same real-world entity. Organizations often struggle with a plethora of customer data captured multiple times in different sources by various people in their own ways. Despite having been studied for decades, entity matching remains a challenging problem in practice. In general, there are several factors that make it difficult to solve:

Poor Data Quality: Real-world data is seldom completely structured, cleansed, and homogeneous. Data originating from manual insertion may contain alternative spellings, typos, or fail to comply with the schema (e.g., mixing of first and last name).

Dependency on Human Knowledge: Same data may be represented in different formats by various users like abbreviations, suffixes, prefixes, etc. To perform matching, our solution must interact with human experts

and make use of their knowledge. Human interaction in itself is a complex domain.

For example, let's look at a customer table from which analyst is trying to identify distinct customers.

Table 1: Customer Records with Duplicates

No.	Name	Address	Email
1	Alexander Great	2/13, Philip Street, Paris, France	alex.gr@gmail.com
2	Alexander G	2/13, Philip Street, Paris	n/a
3	Alexander Graham	10, Middle Street, New York	alex.gr@yahoo.com

Without manual inspection and good understanding of geographical locations, it is difficult to guess whether record 2 is duplicate of 1 or 3. Somewhat ironically, as often pointed out, entity matching suffers from the problem of being referenced by different names, some referring to the exact same problem, while others are slight variations, generalizations, or specializations. In addition, the names are also not used completely consistently. Deduplication or duplicate detection is the problem of identifying records in the same data source that refer to the same entity and can be seen as the special case 1 = 2. Given such representation variations, an unprecedented number of permutations and combinations, the entity matching would be a herculean job when we handle large volume of data. Artificial intelligence and machine learning has become an essential part of multiple research fields in recent years, most notably in natural language processing and computer vision, which are concerned with unstructured data. Its most prominent advantage over systematic approaches is its ability to learn features instead of relying on step-by-step calculations.

a) Problem Definition

Researchers have already realized the potential advantage of machine learning for entity matching. In this paper, we aim to propose a machine learning model for entity matching.

Let E be a data source containing entities. E has the attributes ($A1$,$A2$, ...,An), and we denote entities as e = (e1, e2, ..., en) ∈ E. A data source is a set of records, and a record is a tuple having a specific schema of attributes. An attribute is defined by the intended semantics of its values. So, entities ei = ej if and only if attributes ai of ei are intended to carry the same information as attributes aj of ej, and the specific syntactics of the attribute values are irrelevant. Attributes can also have metadata (like a name) associated with them, but this does not affect the equality between them.

The goal of entity matching is to find the largest possible binary relation $M \subseteq E \times E$ such that a and b refer to the same entity for all (a, b) ∈ M. In other words, we would like to find all record pairs across data source that refer to the same entity. We define an entity to be something of unique existence. Attribute values are often assumed to be strings, but that is not always the case. The records are assumed to operate with the same taxonomic granularity. In this research, we will stick to the definition of deduplication (or duplicate detection) as the problem of identifying which records in the same data source refer to the same entity.

The remainder of this paper is organized as follows. We discuss related work in section 2. In Section 3, we formally formulate the problem and propose our methodology. Section 4 describes how our approach is used to detect similarity in a real-world data set and the results of our experiment are explained. Finally, the paper is concluded in Section 5.

II. Related Work

Entity resolution, record linkage, deduplication and entity matching are frequently used for more or less the same problem as we mentioned earlier. It is a technique to identify data records in a single data source or across multiple data sources that refer to the same real-world entity and to correlate the records together. In entity matching, the strings that are nearly identical, but not exactly the same, are matched without explicitly having a unique identifier. Entity matching is crucial as it matches non-identical records despite all the data inconsistencies without the constant need for formulating rules. By combining databases using fuzzy matching, we can refine the data and analyze the information. Comparing big data records having non-standard and inconsistent data from diverse sources that do not provide any unique identifier is a complex problem. In this section, we present an overview of the previous work done by researchers in entity matching. Researchers use two major techniques as shown below:

Rule-Based: Rule-based systems perform matching based on a set of manually crafted rules. To match any two records of the same entity, various string-based comparison rules are defined. Each record then would run with every other record on all these rules to decide if the two are identical.

Automatic: These systems rely on machine learning algorithms to learn from data. Computers first learn from data provided for training so that they can later make predictions on unknown input data items.

Usually, a rule-based system uses a set of human-crafted rules to help identify subjectivity. As the number of records increases, the number of comparisons increases exponentially in rule-based systems. With large volume of records, rule-based data matching becomes computationally challenging and unscalable. Automatic methods, contrary to rule-based systems, do not rely on manually crafted rules but on machine learning algorithms. There has been an uptick in interest on machine learning as a solution for entity

matching in recent years. We note that this process is machine-oriented and does not highlight any iterative human interactions or feedback loops. First, there are several books that provide an overview. Christen [15] is a dedicated and comprehensive source on entity matching. Anhai Doan et al. [2] and Talburt [10] introduce entity matching in the context of data quality and integration. Quite early on, statisticians dominated the field of entity matching. Probabilistic methods were first developed by Newcombe et al. [15]. A solid theoretical framework was presented by Fellegi and Sunter [9]. Blocking, which is surveyed by Papadakis et al. [8, 9], is considered an important subtask of entity matching. This is meant to tackle the quadratic complexity of potential matches. Christophides et al. [24] specifically review entity matching techniques in the context of big data. Significant research has gone into active learning approaches by Arvind [3], Jungo [11] and Kun [12]. Interestingly, Jungo et al. [11] use a deep neural network in their active learning approach. Such human-in-the-loop factors are often crucial for entity matching in practice as analyzed by Anhai et al. [2]. Many state-of-the-art models for natural language processing are based on deep learning networks. Central to all these approaches is how text is transformed to a numerical format suitable for a neural network. This is mainly done through embeddings, which are translations from text units to a vector space – traditionally available in a lookup table. The text units will usually be characters or words. An embeddings lookup table may be seen as parameters to the network and can be learned together with the rest of the network end-to-end. That way the network is able to learn good, distributed character or word representations for the problem at hand. The words used in a data set are often not unique to that data set, but rather just typical words from some language. Therefore, one may often get a head start by using pretrained word embeddings like word2vec, GloVe or fastText, which have been trained on enormous general corpora. One particular influential recent trend is the ability to leverage huge pretrained models that have been trained unsupervised for language modeling on massive text corpora similar to what the computer vision community has done for image recognition. They produce contextualized word embeddings that consider the surrounding words. These contextual embeddings can be used as a much more powerful variant of the classical word embeddings, but as popularized by BERT. However, with neural networks, the actual line between the initial feature extraction part and the rest is an artificial one and not necessarily indicative of how the networks actually learn and work. But they do reflect design decisions to a certain degree and help us compare them in that regard. Often these approaches use pre-built word embeddings for a specific set of values. Our research focuses on entity matching based on attributes where the number of attributes may vary from one use case to another. Also, we try to address the problem of multiple domains, i.e., the machine learning model must be suitable for entities from various categories like customers, products, vendors, etc. In this paper, we present a machine learning model which will perform attribute-based matching of entities. The type, number of attributes may vary over the time, but our approach does not require re-design. Merely a re-training of the model on the new data set will suffice. The model is robust enough to handle slight variations in ordinality and type of the attributes.

III. Methodology

Most neural network-based methods perform entity matching by producing so-called knowledge graph embeddings, embeddings of entries which incorporate information about their relationship with other entries. The embeddings work mainly at word level or character level. Embeddings offer neural networks an initial mapping from the actual input to a suitable numeric representation. When we surveyed the earlier methods, we found that researchers focus on explicit levels of representation of entities into single word or text. However, we try to address two problems mainly,

- How to perform matching of entities containing attributes of different data types, say string, boolean, and categorical?
- Will the machine learning algorithm continue to work even if the number of attributes change over the time?

Let's say there are few entities in a data set as shown in Table 1. It has two duplicates. Following is a generalized notation.

Table 2: Labelled Entities with Multiple Attributes

Entity	Attribute1	Attribute2	Attribute3	Label
e1	a11	a12	a13	Duplicate (e1 = e2)
e2	a21	a22	a23	
e3	a31	a32	a33	Unique

The entities e1 and e2 are same, though they might vary slightly in their attribute values but have similar meanings. Our aim is to design an approach which will combine the attribute level similarity and artificial intelligence to classify entities as unique or duplicate. We propose a two-step methodology where the first step involves calculating attribute level similarity scores and the second step is classification using supervised learning. Feature extraction involves use of a distance function for every pair of attributes. It transforms every pair of entities into numerical vector. For any give pair of attributes (a_{ij}, a_{kj}), the distance function δ produces a numerical value such that

$$0 <= \delta(a_{ij}, a_{kj}) <= 1$$

If the two attributes are exactly same, then the distance metric is zero. If they are completely unrelated, then the distance is 1. Partial match will result in value between 0 and 1. We call it as similarity score of the attributes.

A sample set of vectors for a set of three entities will be as shown below.

Table 3: Feature Extraction using Similarity Score

Entity Pair	Score1	Score2	Score3	Label
e1,e2	$\delta(a_{11}, a_{21}) = 0.8$	$\delta(a_{12}, a_{22}) = 0.6$	$\delta(a_{13}, a_{23}) = 1$	D
e2,e3	$\delta(a_{21}, a_{31}) = 0.5$	$\delta(a_{22}, a_{32}) = 0.6$	$\delta(a_{23}, a_{33}) = 0$	U
e1,e3	$\delta(a_{11}, a_{31}) = 0.6$	$\delta(a_{12}, a_{32}) = 0.4$	$\delta(a_{13}, a_{33}) = 1$	U

The extracted values correspond to two class labels duplicate (D) and unique (U). If we extract feature vectors of a data set and plot the points in a 3-dimensional space, then we will see two clusters as shown below.

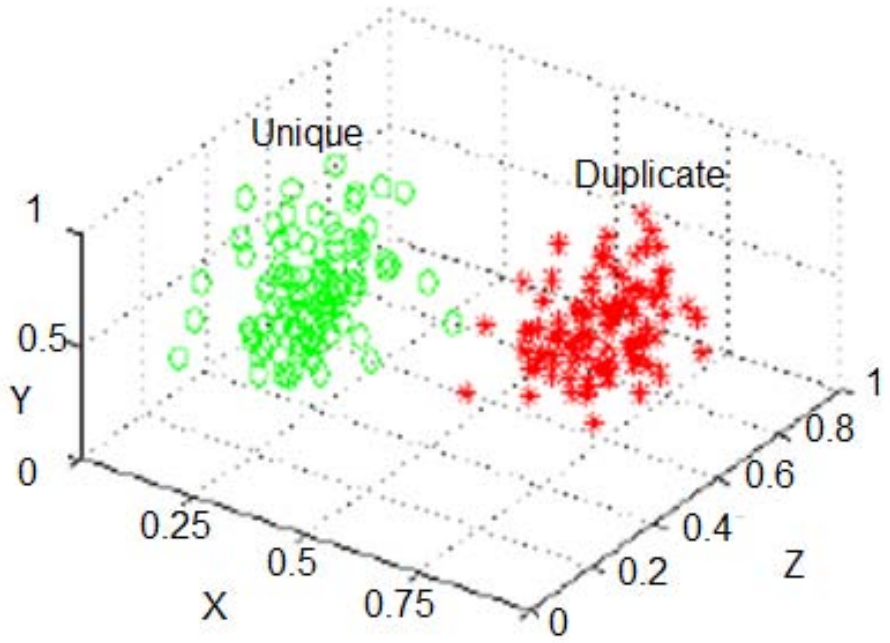

Figure 1: Feature Vectors Plotted in 3-D Space

For m entities having *n* attributes, after feature extraction, we will get m *x n* values under the two labels. Now, the entity matching problem is reduced to a binary classification problem, where the objective is to predict a pair of entities as unique or duplicate. Feature extraction involves attribute level comparison using fuzzy matching algorithms. The produced output is a labelled data set which can be used to train a model using supervised learning algorithm. A well-trained model will make predictions over the incoming data point. Points which lie around the boundary or away from the cluster centroid might require manual stevedoring. Following diagram shows the architecture of our machine learning based entity matching system.

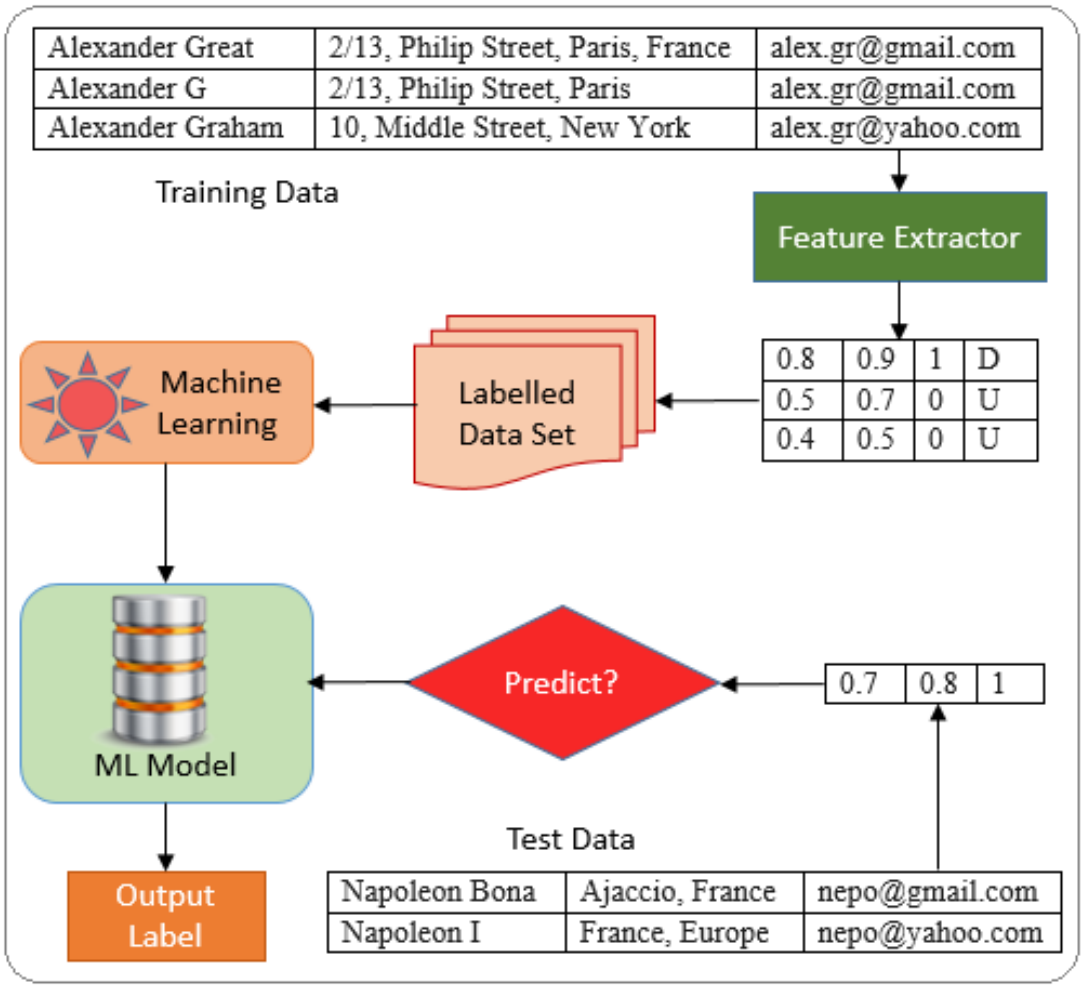

Figure 2: Architecture of Entity Matching System

Our approach takes every pair of entities and produces a numerical vector. This is in turn fed to a machine learning algorithm for classification. We use supervised learning algorithm for classification. The ML model learns from the training data set and makes accurate predictions on the incoming test data.

a) Feature Extraction using Similarity Score

The first step in ML modeling is data preprocessing, which is usually a crucial step in many data analytics tasks. Typical transformations involve lowercasing all letters, removing excess punctuation, normalizing values, and tokenizing. There are two other major steps in our process. Second one being the feature vector construction using similarity score and last step is machine learning. One might also view second step as feature extraction, since records are transformed to a feature space. The success of this entity matching systems depends upon careful selection of right algorithms. Attributes are often assumed to be strings, but that is not the case always. Attributes of an entity may be of any data type like string, numeric, categorical, boolean etc. One single function will not be able to calculate similarity score for various attributes to attribute. It is useful to compare various functions available for similarity score and pick the right choice. To this end, we present a high-level overview of few popular algorithms.

Table 4: Similarity Score Functions

No.	Data Type	Similarity Function
1	Single Word String	Exact Match
2		Levenshtein Distance
3		Jaro Distance
4		Jaro-Winkler Distance
5		Jaccard Similarity

6	2-to-5 Words String	Cosine Similarity
7		Levenshtein Distance
8		Jaccard Similarity
9		Needleman-Wunsch Algorithm
10		Smith-Waterman Algorithm
11		Monge-Elkan Algorithm
12	Long String (>5 Words)	Cosine Similarity
13		Levenshtein Distance
14		Jaccard Similarity
15		Monge-Elkan Algorithm
16	Number	Exact Match
17		Absolute Difference
18	Categorical	Exact Match
19	Boolean	Exact Match

For example, consider a similarity function Levenshtein Distance. The Levenshtein distance between 'new yrk' and 'new york' is one since it needs at least one edit (insertion, deletion, or substitution) to transform from 'new yrk' to 'new york'. It is advisable to normalize the similarity scores between 0 and 1 for improved accuracy of the machine learning algorithm.

b) Classification using Supervised Learning

The matching phase aims to develop the prediction model, which takes a candidate pair as input and predicts whether they are matching or non-matching. Figure 2 illustrates that the model predicts an output label Duplicate (D) or Unique (U). This is a binary classification problem. Data scientists need to decide which algorithm is most suitable for their classification task. Based on our study and experiments, we found three classification algorithms suitable for this task.

i. *K-Nearest Neighbors (KNN)*

K-Nearest Neighbors (KNN) is an algorithm that learns all available cases from data set and classifies new data item by a majority vote of its K neighbors. A case assigned to the data is majority of its K nearest neighbors measured by a distance (metric) function. The metric functions include Euclidean, Manhattan, Minkowski, and Hamming distances. KNN can be used for both regression and classification problems. However, it is widely used in classification problems in the industry.

ii. *XG Boost*

XG Boost stands for Extreme Gradient Boosting. It is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. It provides parallel tree boosting and is the leading machine learning library for regression, and classification problems.

iii. *Support Vector Machines (SVM)*

Support Vector Machine is a supervised algorithm in which the learning algorithm analyzes data and recognizes patterns. We plot the data as points in an n-dimensional space. The value of each feature is then tied to a particular co-ordinate, making it easy to classify the data.

And finally, we need to tune hyper-parameters in order to get the best model performance.

IV. Experiments and Results

Automatic entity matching makes the life of commercial organizations easier. A company that maintains thousands of customer records cannot afford to employ many people to verify manually and identify duplicates. Artificial Intelligence based entity matching is an efficient and cost-effective analytics tool for operational efficiency. We used open-source data sets for our experiments. While several open-source datasets are available, we picked up few commercial data sets for analysis. In this section, we describe the evaluation tasks, the data sets used, and the experimental results of our approach.

Evaluation Tasks:

1. We evaluate our approach on real-world data set.
2. We evaluate our approach on popular benchmarks.

Our goal is to provide real-life solution using our approach. We aim to evaluate the quality of entity matching. The empirical result is compared with real-time data to harness the accuracy. The results show promising output.

a) Data Set

We conducted extensive experiments on real-world benchmark entity datasets to evaluate the performance of approach. Following are few open-source data sets available for evaluating entity matching algorithms.

Table 5: Entity Matching Data Sets

No.	Dataset	Description	Training Size	Testing Size	No. of Attributes
1	Fodors-Zagats	Customer records with name, address, city, phone, type, and category code.	757	189	6
2	iTunes-Amazon	Records of songs with song name, artist name, album name, genre, etc.	430	109	8
3	DBLP-ACM	Publication dataset with paper title, author, venue etc.	9890	2473	4
4	DBLP-Scholar	Publication dataset with title, authors, venue, and year.	22965	5742	4
5	Amazon-Google	Software product dataset with attributes product title, manufacturer, and price.	9167	2293	3
6	Walmart-Amazon	Electronic product dataset with attributes product name, category, brand, model number, etc.	8193	2049	5
7	Abt-Buy	Product dataset with attributes product name, price, and description.	7659	1916	3

Many commercial organizations are nowadays struggling with customer de-duplication. Automatic de-duplication has significance in various sectors like Banking and Finance, Insurance, Telecom, Retail, etc. Hence our results mainly focus on the evaluation metrics accuracy on the customer data set.

b) Popular Metrics

In this section, we first describe a set of metrics commonly used for evaluating the performance of our classification model. Then we present a quantitative analysis of the performance using popular benchmarks.

Accuracy and Error Rate: These are primary metrics to evaluate the quality of a classification model. Let TP, FP, TN, FN denote true positive, false positive, true negative, and false negative, respectively. The classification Accuracy and Error Rate are defined in Equation 1.

$$\textbf{Accuracy} = \frac{(TP + TN)}{N}, \quad \textbf{Error rate} = \frac{(FP + FN)}{N} \tag{1}$$

where N is the total number of samples. Obviously, we have Error Rate = 1 – Accuracy.

Precision, Recall, and F1 Score: These are also primary metrics and are more often used than accuracy or error rate for imbalanced test sets. Precision and recall for binary classification are defined in Equation 2. The F1 score is the harmonic mean of the precision and recall, as in Equation 2. F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$\textbf{Precision} = \frac{TP}{TP + FP}, \quad \textbf{Recall} = \frac{TP}{TP + FN}, \quad \textbf{F1-score} = \frac{2 * Prec * Rec}{Prec + Rec} \tag{2}$$

For multi-class classification problems, we can always calculate precision and recall for each class label and analyze the individual performance on class labels or average the values to get the overall precision and recall. In our case, the average for the two labels Duplicate (D) and Unique (U) were calculated and the following diagram is the pictorial representation of the metrics.

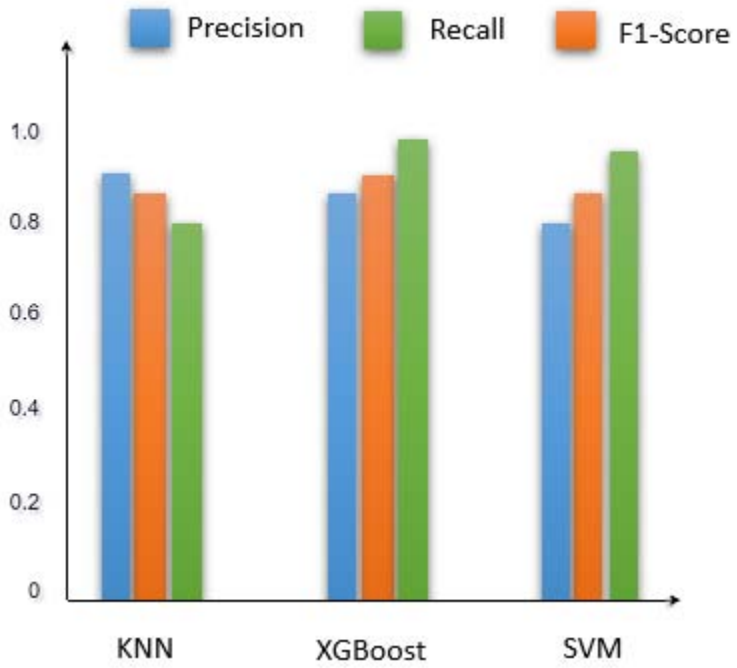

Figure 3: Quantitative Metrics Analysis

From the above results, we observe that XGBoost has highest F1-Score and best suited for the entity matching problem. Following table shows the final metrics of experiments conducted using various similarity score and classification algorithms over Fodors-Zagats dataset.

Table 6: F1-Score of Various Algorithms

		CLASSIFICATION ALGORITHM		
		XGBoost	KNN	SVM
SIMILARITY SCORE	Jaccard Similarity	87%	86%	83%
	Levenshtein Distance	84%	83%	81%
	Cosine Similarity	83%	81%	80%

c) *Empirical Results*

We aim to use our entity matching systemin real world applications like retail, e-commerce etc. We analyzed the presence of duplicate customer data and results showed more that 80% accuracy in read-world data sets. Following is a set of predictions made by our system from Fodors-Zagats dataset.

Table 7: Entities Matched using Automated System

No.	Name	Address	City	Phone	Label
1	restaurant ritz-carlton atlanta	181 Peachtree st.	Atlanta	404/659 -0400	D
2	ritz-carlton restaurant	181 Peachtree st.	Atlanta	404/659 -0400	
3	posterior	545 post st.	San Francisco	415/776 -7825	D
4	postrio	545 post street.	San Francisco	415/776 -7825	
5	tavern on the green	in central park at 67th st	New York	212/873 -3200	D
6	tavern on the green	central park west	New York	212/873 -3200	
7	carey's	1021 cobb pkwy . se	marietta	770-422-8042	U

8	carey's corner	1215 powers ferry rd.	marietta	770-933-0909	
9	chops	70 w. paces ferry rd.	atlanta	404-262-2675	U
10	chopstix	4279 roswell rd.	atlanta	404-255-4868	

From the above table, we observe that customers, vendors can easily get their ambiguities resolved using automatic entity matching system. AI-based entity matching is an alternative to traditional manual or other text analysis-based tools, and it is cost-effective solution for decision-makers.

V. Conclusion and Future Work

The proposed method accomplished superior performance in terms of time and cost. The overall benefits of AI-based entity matching include:

Sorting Data at Scale: Manually screening thousands of customer records, or product details is complex and time-consuming. AI-based entity matching helps businesses process large amount of data in an efficient and cost-effective way.

Real-Time Analysis: The automatic entity matching can help organizations quickly identify duplicates on real-time basis and act swiftly before duplicate marketing or promotional offers are sent out.

Though many deep learning models are being developed nowadays for entity matching, we propose a supervised learning model for few major reasons.

Explainability and Ease of Debugging: For many applications, it is crucial to trust the data source, and try to understand why something does not work is key. Unfortunately, deep learning models are notoriously hard to interpret. As steps in the entity matching process increasingly coalesce into a large neural network, we get fewer checkpoints along the way in the process that can easily be inspected. We can't see the output from each step in the same way anymore. Therefore, figuring out why two records where matched or not matched is usually nontrivial while inspecting deep learning models. There are a few techniques that are already used, such as looking at alignment scores, but we are still far away from a comprehensive way of debugging neural networks for entity matching. Our model addresses the challenges of explainability, running time in interactive settings, and the large need for training examples. Explainability of our supervised learning algorithm helps researchers to improve accuracy through inspection, comparison of algorithms and meet the real-world demands. We also see a lot of opportunities in trying to develop more open datasets, standardized benchmarks, and publicly available pretrained models for entity matching.

References Références Referencias

1. A Arasu, S Chaudhuri, and R Kaushik. 2008. Transformation-based Framework for Record Matching. *In 2008 IEEE 24th International Conference on Data Engineering*.ieeexplore.ieee.org, 40–49.
2. Anhai Doan, Adel Ardalan, Jeffrey Ballard, Sanjib Das, Yash Govind, Pradap Konda, Han Li, Sidharth Mudgal, Erik Paulson, G C Paul Suganthan, and Haojun Zhang. 2017. Human-in-the-loop challenges for entity matching: *A midterm report. In Proceedings of the 2Nd Workshop on Human-In-the-Loop Data Analytics (HILDA'17)*. dl.acm.org, New York, NY, USA, 12:1–12:6.
3. Arvind Arasu, Michaela Götz, and Raghav Kaushik. 2010. On Active Learning of Record Matching Packages. *In Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data (SIGMOD '10*). ACM, New York, NY, USA, 783–794.
4. A K Elmagarmid, P G Ipeirotis, and V S Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Trans. Knowl. Data Eng. 19,* 1 (Jan. 2007), 1–16.
5. Cheng Fu, Xianpei Han, Le Sun, Bo Chen, Wei Zhang, Suhui Wu, and Hao Kong. 2019. End-to-end Multi-perspective Matching for Entity Resolution. *In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI'19). AAAI Press*, Macao, China, 4961–4967.
6. Ekaterini Ioannou, Nataliya Rassadko, and Yannis Velegrakis. 2013. On Generating Benchmark Data for Entity Matching. *Jr. Data Semant. 2, 1*, 37–56, March 2013.
7. Felix Naumann and Melanie Herschel. 2010. An Introduction to Duplicate Detection. *Morgan and Claypool Publishers.*
8. George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. *A Survey of Blocking and Filtering Techniques for Entity Resolution*. arXiv:cs.DB/1905.06167, May 2019.
9. George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. Comparative Analysis of Approximate Blocking Techniques for Entity Resolution. *Proceedings VLDB Endowment* 9, 684–695, May 2016.
10. Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of Entity Resolution Approaches on

Real-world Match Problems. *Proceedings VLDB Endowment 3, 1-2* (Sept. 2010), 484–493.
11. Ivan P Fellegi and Alan B Sunter. 1969. A Theory for Record Linkage. *Jr. Am. Stat. Assoc. 64, 328* (Dec. 1969), 1183–1210.
12. John R Talburt. 2011. Entity Resolution and Information Quality. *Elsevier*.
13. Jungo Kasai, Kun Qian, Sairam Gurajada, Yunyao Li, and Lucian Popa. 2019. Low-resource Deep Entity Resolution with Transfer and Active Learning. *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, Stroudsburg*, PA, USA, 5851–5861.
14. Kun Qian, Lucian Popa, and Prithviraj Sen. 2017. Active Learning for Large-Scale Entity Resolution. *In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM '17). ACM*, New York, NY, USA, 1379–1388.
15. Lise Getoor and Ashwin Machanavajjhala. 2012. Entity Resolution: Theory, Practice & Open Challenges. *Proceedings VLDB Endowment 5, 12* (Aug. 2012), 2018–2019.
16. Muhammad E, Saravanan T, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *Proceedings VLDB Endowment 11,* 11 (July 2018), 1454–1467.
17. Newcombe, H B, J M Kennedy, S J Axford, and A P James. 1959. Automatic linkage of vital records. Science 130, 3381, 954–959, Oct. 1959.
18. Nihel Kooli, Robin Allesiardo, and Erwan Pigneul. 2018. Deep Learning Based Approach for Entity Resolution in Databases. *In Intelligent Information and Database Systems.* Springer International Publishing, 3–12.
19. Peter Christen. 2012. Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection. *Springer Science & Business Media*.
20. P Christen. 2012. A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. *IEEE Trans. Knowl. Data Eng*. 24, 9 (Sept. 2012), 1537–1555.
21. Ram Deepak Gottapu, Cihan Dagli, and Bharami Ali. 2016. Entity Resolution Using Convolutional Neural Network. *Procedia Comput. Sci. 95* (Jan. 2016), 153–158.
22. Thomas N Herzog, Fritz J Scheuren, and William E Winkler. 2007. *Data Quality and Record Linkage Techniques*. Springer Science & Business Media.
23. Ursin Brunner and Kurt Stockinger. 2020. Entity matching with transformer architectures-a step forward in data integration. *In International Conference on Extending Database Technology, Copenhagen*, 30 March-2 April 2020.
24. Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2019. End-to-End Entity Resolution for Big Data: A Survey. (May 2019). arXiv:cs. DB/1905.06397.
25. Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Anhai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. (April 2020). arXiv:cs.DB/2004.00584.