Scalable Similarity Joins for Fast and Accurate Record Deduplication in Big Data

Ondrej Rozinek[1], Monika Borkovcova[2], and Jan Mares[1,3]

[1] University of Pardubice, Department of Process Control, Studentska 95, 532 10 Pardubice, Czech Republic,
ondrej.rozinek@gmail.com,
[2] University of Pardubice, Department of Information Technology, Studentska 95, 532 10 Pardubice, Czech Republic,
[3] University of Chemistry and Technology Prague, Department of Mathematics, Informatics and Cybernetics, Technicka 5, 166 28 Prague, Czech Republic

Abstract. Record linkage is the process of matching records from multiple data sources that refer to the same entities. When applied to a single data source, this process is known as deduplication. With the increasing size of data source, recently referred to as big data, the complexity of the matching process becomes one of the major challenges for record linkage and deduplication. In recent decades, several blocking, indexing and filtering techniques have been developed. Their purpose is to reduce the number of record pairs to be compared by removing obvious non-matching pairs in the deduplication process, while maintaining high quality of matching. Currently developed algorithms and traditional techniques are not efficient, using methods that still lose significant proportion of true matches when removing comparison pairs. This paper proposes more efficient algorithms for removing non-matching pairs, with an explicitly proven mathematical lower bound on recently used state-of-the-art approximate string matching method - Fuzzy Jaccard Similarity. The algorithm is also much more efficient in classification using Density-based spatial clustering of applications with noise (DBSCAN) in log-linear time complexity $\mathcal{O}(|\mathcal{E}|\log(|\mathcal{E}|))$.

Keywords: record deduplication, Q-gram filter, record linkage, entity resolution, similarity space, bipartite matching, similarity join

1 Introduction

Record deduplication, a critical process in data management, refers to the identification and removal of duplicate records in databases. This process is essential for maintaining data quality and integrity, especially in large databases where duplicate records can lead to inconsistent, misleading, or erroneous data analysis [1, 4].

The growing volume of digital data poses significant challenges to data deduplication. Traditional methods, which rely primarily on rule-based and exact

matching techniques, are often unable to cope with the complexity of today's datasets, which include natural language variations.

Recent advances in machine learning and natural language processing have opened up new opportunities for more sophisticated deduplication strategies. [1, 8, 9]. These approaches use complex algorithms to identify duplicates with higher accuracy, even in data sets with high variability and noise.

This article aims to extend the model of similarity join methods for bipartite record matching, incorporating a newly developed count Q-gram filter, with an application to record deduplication.

2 Related Work

Recent developments in Q-gram count filters have had a significant impact on methods for approximate string matching and data deduplication. Ukkonen's seminal work [12] introduced Q-grams to string processing and laid the foundation for subsequent algorithmic advances. This approach has been further refined, as shown by Yang et al. [17], to be suitable for large-scale data environments.

The introduction of the Ed-join algorithm by Xiao et al. (2008) [16] marked a notable advance in algorithmic development for similarity joins with edit distance constraints, using Q-grams to optimise performance. At the same time, the survey by Yu et al. (2016) [18] emphasized the effectiveness of Q-gram-based techniques in string similarity joins, highlighting their role in balancing accuracy and computational efficiency.

Hybrid approaches that integrate Q-gram count filters with other computational methods have demonstrated improved effectiveness in data deduplication. Jiang et al. (2014) [6] illustrated the benefits of combining Q-gram filters with token-based methods, showing improved results on datasets characterised by typographical variations.

Challenges of scalability and data noise continue to drive innovation in the field. Vernica et al. (2010) [13] addressed scalability with a parallelized approach to set-similarity joins using Q-gram methods. Koudas et al. (2006) focused on the use of Q-gram filters in flexible string matching against large, diverse databases.

Recent research, such as that of Papadakis et al. (2020) [8], indicates a trend towards integrating machine learning with Q-gram count filters to improve entity resolution in deduplication processes. This integration represents a potential shift towards more sophisticated data processing capabilities.

3 Problem Formulation

At its core, deduplication involves the concept of entities (or entity profiles) [8], which provide a uniquely identified description of a real-world object in the form of name-value pairs. Two entities e_i and e_j match, $e_i \equiv e_j$, if they refer to the same real-world entity. Matching entities are also called duplicates. The task of entity resolution is to find all matching entities within an entity collection or

across two or more entity collections. The term entity is also interchangeable with the term record, which is mainly used in the fields of databases and data storage.

Definition 1 (Deduplication). *Deduplication is a process represented by a function $\mathcal{D}\colon \mathcal{E} \to \mathcal{C}$, where $\mathcal{E}$ is a collection of entities, and $\mathcal{C}$ is a collection of clusters of duplicate entities within $\mathcal{E}$. Each cluster in $\mathcal{C}$ consists exclusively of entities that are considered equivalent (duplicates) under a specified equivalence relation $\equiv$. Formally, the function is defined as:*

$$\mathcal{D}(\mathcal{E}) = \mathcal{C} = \{\{e_i, \ldots, e_j\}\colon e_i, \ldots, e_j \in \mathcal{E}, \forall e_i \equiv e_j, i \neq j\}. \tag{1}$$

The definition based on the family of sets $\mathcal{C}$ imposes the necessity to have the output as a cluster of entities with the same entity resolution. This is the main difference with the introduced definition [8].

The use of clusters requires the use of Euclidean or metric spaces, which provide the fundamental properties of most clustering methods. These spaces define distance metrics, which are crucial optimization criteria for many clustering algorithms. Working in non-metric spaces presents challenges such as difficulties in point localization, distance measurement, algorithm convergence, and identifying cluster shapes. However, a wide range of similarity functions, such as the Jaccard index, Tanimoto coefficient and edit similarity, have been shown to be dual to metric spaces [11]. We refer to this dual construct as a 'similarity space', which shapes a different axiomatic system. We therefore focus on similarity spaces with proven duality to metric spaces [11].

Definition 2 (Similarity Space [10, 11]). *Given a non-empty set $\mathcal{X}$, a function $s\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a similarity metric if for all elements $x, y, z \in \mathcal{X}$, it satisfies the following axioms:*

(S1) $s(x, y) = s(y, x)$ (symmetry),
(S2) $s(x, z) + s(y, y) \geq s(x, y) + s(y, z)$ (triangle inequality),
(S3) $s(x, x) = s(x, y) = s(y, y) \iff x = y$ (identity of indiscernibles),
(S4) $s(x, y) \geq 0$ (non-negativity),
(S5) $s(x, y) \leq \min\{s(x, x), s(y, y)\}$ (bounded by self-similarity).

A similarity space is an ordered pair $(\mathcal{X}, s)$.

Theorem 1 (Generalized Rozinek Distance [11]). *Suppose given a normalized similarity metric s_n and an arbitrary similarity metric s. The Generalized Rozinek distance $d_R\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is the distance metric derived from an arbitrary normalized similarity metric $s_n(x, y)$ and self-similarities $s(x, x)$ and $s(y, y)$ by*

$$d_R(x, y) = \frac{1 - s_n(x, y)}{1 + s_n(x, y)}(s(x, x) + s(y, y)). \tag{2}$$

Proof. [11]

Definition 3 (Self-Join in Similarity Space). *Given an entity collection $\mathcal{E}$, a similarity metric $s\colon \mathcal{E}^2 \to \mathbb{R}$, and a similarity threshold α, a similarity join identifies all pairs of entity in $\mathcal{E}$ that have similarity at least α*

$$\mathcal{E} \bowtie_\alpha \mathcal{E} = \{(e_i, e_j) \in \mathcal{E}^2 \colon s(e_i, e_j) \geq \alpha, i \neq j\}. \tag{3}$$

The effectiveness of deduplication lies in its ability to detect true positive duplicates, while the efficiency is related to the computational cost of these detections, usually measured by the number of comparisons or the computational time complexity $\mathcal{O}(D(\mathcal{E}))$. The brute-force approach, involving all pairwise comparisons within an entity collection $\mathcal{E}$, leads to a quadratic complexity $\mathcal{O}(D(\mathcal{E})) = \mathcal{O}(|\mathcal{E}|^2)$. For deduplicating a single structured data source with $|\mathcal{E}|$ entities, the maximum number of comparisons is equal to half the entries in a $|\mathcal{E}| \times |\mathcal{E}|$ symmetric matrix, excluding the diagonal. This results in a final time complexity of $\mathcal{O}(D(\mathcal{E})) = (|\mathcal{E}|^2 - |\mathcal{E}|)/2$, since each entity may have to be compared with all others.

To avoid an exhaustive pairwise comparison, the similarity join typically consists of two steps

- *Filtering* is a function $\mathcal{F}_\alpha\colon \mathcal{E}^2 \to \mathcal{E}^2$ that returns a set of candidates for each entity e_i, excluding all those that do not match e_i.
- *Matching* is a function $\mathcal{M}_\alpha\colon \mathcal{E}^2 \to \mathcal{E}^2$.

Self-Join in similarity space could be decomposed into functional composition as follows

$$\mathcal{E} \bowtie_\alpha \mathcal{E} = \mathcal{M}_\alpha \circ \mathcal{F}_\alpha. \tag{4}$$

4 Self-Join in Similarity Space

In this generalization, we consider two entities e_i, e_j each comprising token sets $\mathcal{X}$ and $\mathcal{Y}$, which are matched in a bipartite graph using Fuzzy Jaccard Similarity. The variable $|\mathcal{M}|$ represents the maximum number of connected token pairs in the optimal combinatorial assignment problem.

Definition 4 (Fuzzy Jaccard Similarity [14, 15]). *At our disposal are two sets of tokens, $\mathcal{X}$ and $\mathcal{Y}$. Write $\mathcal{X} \widetilde{\cap} \mathcal{Y}$ for the fuzzy overlap of $\mathcal{X}$ and $\mathcal{Y}$: Fuzzy Jaccard Similarity, $s_n(\mathcal{X}, \mathcal{Y})$,*

$$s_n(\mathcal{X}, \mathcal{Y}) = \frac{|\mathcal{X} \widetilde{\cap} \mathcal{Y}|}{|\mathcal{Y}| + |\mathcal{X}| - |\mathcal{X} \widetilde{\cap} \mathcal{Y}|}. \tag{5}$$

In the articles [14, 15], the incident edge for the token pair is considered only if $s_n(X_i, Y_j) \geq \delta$. In our definition, we have dropped the second threshold δ for a more fuzzy approach, because applying δ would essentially create a binary classifier determining whether a token pair (X_i, Y_j) is classified as a match. Furthermore, the original paper violates the triangle inequality $S2$ due to the normalization of edit distance $d_n(X_i, Y_j) = \frac{d(X_i, Y_j)}{\max |x|, |y|}$.

Example 1. Consider the strings $X = "ab"$, $Y = "abc"$, and $Z = "bc"$. Then we obtain

$$\begin{aligned} &d(X,Z) \leq d(X,Y) + d(Y,Z), \\ &\frac{d(X,Z)}{max\{|X|,|Z|\}} \leq \frac{d(X,Y)}{max\{|X|,|Y|\}} + \frac{d(Y,Z)}{max\{|Y|,|Z|\}}, \\ &\frac{2}{2} \not\leq \frac{1}{3} + \frac{1}{3}. \end{aligned} \tag{6}$$

Theorem 2 (Threshold of Normalized Edit Similarity Metric). *Let the edit distance be $d(X,Y)$, the worst case of the expected distance function be $d(\alpha,|X|,|Y|)$, and write the floor function by $\lfloor . \rfloor$. Then*

$$s_n(X,Y) \geq \alpha \Longleftrightarrow d(X,Y) \leq d(\alpha,|X|,|Y|) = \left\lfloor \frac{1-\alpha}{1+\alpha}(|X|+|Y|) \right\rfloor, \tag{7}$$

where $\alpha \in [0,1] \subset \mathbb{R}$ is a threshold of the normalized similarity metric given by $s_n(X,Y) \geq \alpha$.

Proof. According to Theorem 1 and substituting for self-similarities which equal the corresponding cardinality of the sets, $s(X,X) = |X|$, $s(Y,Y) = |Y|$, the expected distance $d(\alpha,|X|,|Y|)$ reaches a maximum just when the similarity is minimal under the lowest similarity given by the threshold α. This happens if and only if we substitute $s_n(X,Y) = \alpha$:

$$\begin{aligned} d(X,Y) = \lfloor d_R \rfloor &= \left\lfloor \frac{1-s_n(X,Y)}{1+s_n(X,Y)}(s(X,X)+s(Y,Y)) \right\rfloor \\ &\leq \sup_{\alpha} d(X,Y) = \left\lfloor \frac{1-\alpha}{1+\alpha}(|X|+|Y|) \right\rfloor = d(\alpha,|X|,|Y|). \end{aligned} \tag{8}$$

The edit distance is an integer, hence we use the floor function to remove the undefined fractional part.

4.1 Optimal Count Q-gram Filter

In [7], a lower bound relationship is established between edit distance and the Q-gram method for a pattern string X of length $|X|$ and a text string Y of length $|Y|$. This lower bound is crucial for string similarity search and similarity join algorithms, which are widely applied in data cleaning, search engines, and data integration [18]. These algorithms, going beyond traditional exact search methods, handle data errors and inconsistencies. Their importance lies in speeding up similarity joins and minimizing exhaustive pairwise comparisons.

Theorem 3 (Q-gram Count Filtering [7, 17]). *Let X and Y be strings with the edit distance $d(X,Y)$. Then, the Q-gram similarity $|Q_X \cap Q_Y|$ of the token X and Y is at least*

$$t = \inf_{d}\{|Q_X \cap Q_Y|\} = \max\{|X|,|Y|\} - q + 1 - qd(X,Y), \tag{9}$$

where t is a Q-gram similarity threshold with respect to $d(X, Y)$.

We refine the Q-Gram Count filter model by introducing more precise assumptions for the token sets $\mathcal{X}, \mathcal{Y}$ and derive an optimal filter that ensures no comparison pair with a Fuzzy Jaccard Similarity higher than α is lost.

We assume that the matching token pairs provided by $\mathcal{M}$ are known, while their edit distances are unknown. However, these distances can be predicted based on expected edit distances.

Theorem 4 (Optimal Count Q-gram Filter for Bipartite Matching). *Let $\mathcal{X}$ and $\mathcal{Y}$ be records representing a set of tokens. Then the Q-gram similarity in bipartite matching of $\mathcal{X}$, $\mathcal{Y}$ and cardinality $|\mathcal{M}|$ for a given threshold $s_n(\mathcal{X}, \mathcal{Y}) \geq \alpha$ is at least*

$$\begin{aligned} t_{\mathcal{M}} &= \inf_{\alpha} \{|Q_{\mathcal{X}} \cap Q_{\mathcal{Y}}|\} \\ &= \underbrace{\sum_{(i,j)\in\mathcal{M}} \max\{|X_i|, |Y_j|\} - |\mathcal{M}|q + |\mathcal{M}|}_{\textit{maximum shared Q-grams}} \underbrace{- q \max_{\alpha} \sum_{(i,j)\in\mathcal{M}} d(\alpha_{i,j}, |X_i|, |Y_j|)}_{\textit{loss function}}, \end{aligned} \tag{10}$$

containing a linear combination of

$$d(\alpha_{i,j}, |X_i|, |Y_j|) = \frac{1 - \alpha_{i,j}}{1 + \alpha_{i,j}} (|X_i| + |Y_j|) \tag{11}$$

under the constraint $\alpha = \frac{\sum_{(i,j)\in\mathcal{M}} \alpha_{i,j}}{|\mathcal{X}|+|\mathcal{Y}|-\sum_{(i,j)\in\mathcal{M}} \alpha_{i,j}}$ for which the linear combination is maximized.

Proof. Consider the sum over connected pairs of tokens with cardinality $|\mathcal{M}|$

$$\begin{aligned} &\inf_{\alpha} \{|Q_{\mathcal{X}} \cap Q_{\mathcal{Y}}|\} = \inf_{\alpha} \left\{ \sum_{(i,j)\in\mathcal{M}} |Q_{X_i} \cap Q_{Y_i}| \right\} = \sum_{(i,j)\in\mathcal{M}} \inf_{\alpha_{i,j}} \{|Q_{X_i} \cap Q_{Y_j}|\} \\ &= \sum_{(i,j)\in\mathcal{M}} \inf_{\alpha_{i,j}} \{\max\{|X_i|, |Y_j|\} - q + 1 - q d(X, Y)\} \\ &= \sum_{(i,j)\in\mathcal{M}} \{\max\{|X_i|, |Y_j|\} - q + 1 - q \sup_{\alpha_{i,j}} d(X, Y)\} \\ &= \sum_{(i,j)\in\mathcal{M}} \max\{|X_i|, |Y_j|\} - |\mathcal{M}|q + |\mathcal{M}| - q \max_{\alpha} \sum_{(i,j)\in\mathcal{M}} d(\alpha_{i,j}, |X_i|, |Y_j|). \end{aligned} \tag{12}$$

Each $\alpha_{i,j}$ is the distributed minimum similarity for each token, giving a threshold vector that should maximize the sum of the expected distances $d(\alpha_{i,j}, |X_i|, |Y_j|)$ so that $s_n(\mathcal{X}, \mathcal{Y}) \geq \alpha$ holds for t_M. Formalizing this, we get the task

$$\begin{aligned} &\text{maximize} \sum_{(i,j)\in\mathcal{M}} d(\alpha_{i,j}, |X_i|, |Y_j|), \\ &\text{subject to} \sum_{(i,j)\in\mathcal{M}} \alpha_{i,j} \geq \frac{\alpha}{1+\alpha} (|\mathcal{X}| + |\mathcal{Y}|) \quad \alpha \in [0, 1],\, i = 1, \ldots, |\mathcal{M}|, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \alpha_{i,j} \in [0, 1],\, j = 1, \ldots, |\mathcal{M}|. \end{aligned}$$

This leads to an integer linear programming task equivalent to the Knapsack problem, solvable in $\mathcal{O}(nb)$ time. The optimization algorithm determines the maximum expected edit distance distribution across tokens, maintaining the similarity threshold $s_n(\mathcal{X}, \mathcal{Y}) \geq \alpha$.

We aim to merge the objective function and its constraint into a single expression using the Lagrange multiplier method

$$\begin{aligned}&\mathcal{L}(\alpha_{i,j}, ..., \alpha_{|\mathcal{M}|}, \lambda)\\ &= \sum_{(i,j)\in\mathcal{M}} \frac{1-\alpha_{i,j}}{1+\alpha_{i,j}}(|X_i|+|Y_j|) - \lambda\left(\frac{\alpha}{1+\alpha}(|\mathcal{X}|+|\mathcal{Y}|) - \sum_{(i,j)\in\mathcal{M}} \alpha_{i,j}\right).\end{aligned}$$

and solve $\nabla_{\alpha_{i,j},...,\alpha_{|\mathcal{M}|},\lambda}\mathcal{L}(\alpha_{i,j}, ..., \alpha_{|\mathcal{M}|}, \lambda) = 0$. Differentiating with respect to a specific $\alpha_{i,j}$ and setting the derivative to zero:

$$\frac{\partial\mathcal{L}}{\partial\alpha_{i,j}} = -\frac{2(|X_i|+|Y_j|)}{\alpha_{i,j}^2+2\alpha_{i,j}+1} - \lambda = 0. \tag{13}$$

Differentiating with respect to λ and setting this derivative to zero:

$$\frac{\partial\mathcal{L}}{\partial\lambda} = \frac{\alpha}{1+\alpha}(|\mathcal{X}|+|\mathcal{Y}|) - \sum_{(i,j)\in\mathcal{M}} \alpha_{i,j} = 0. \tag{14}$$

Solving these equations will yield the values of $\alpha_{i,j}$ and the optimal λ for the given optimization problem. From this, we can solve for λ as follows

$$\lambda = -\frac{2(|X_i|+|Y_j|)}{\alpha_{i,j}^2+2\alpha_{i,j}+1}. \tag{15}$$

4.2 Approximate Count Q-gram Filter

Considering the analytical intractability of the optimal count Q-gram filter, our goal is to establish a suitable approximation that maintains accuracy while ensuring computational efficiency, achieving a constant time complexity of $O(1)$.

Our approach is predicated on the following assumptions:

- The cardinalities of the sets $\mathcal{X}$, $\mathcal{Y}$, and the matching set $\mathcal{M}$ are equivalent, i.e., $|\mathcal{X}| = |\mathcal{Y}| = |\mathcal{M}|$.
- For every token pair $(i, j) \in \mathcal{M}$, the length of Y_j is assumed to be the expected value of the lengths of tokens in $\mathcal{Y}$, represented as $|Y_j| = \mathbb{E}[Y_j]$.
-
- The similarity threshold α is uniformly applied across all token pairs, such that $\mathbb{E}[\alpha_{i,j}] = \alpha$, and hence $\mathbb{E}[\mathcal{L}(\alpha_{i,j}, ..., \alpha_{|\mathcal{M}|}, \lambda)] = \mathcal{L}(\alpha, \lambda)]$.
- The expected number of destroyed Q-grams, $\mathbb{E}[q_{i,j}]$, is estimated based on the assumption that the edit distance $d = 1$ is a uniformly distributed random variable over the token. This estimation is represented by the formula:

$$\mathbb{E}[q_{i,j}] = q \cdot \frac{\sup|Q_{X_i} \cap Q_{Y_j}|}{\max\{|X_i|, \mathbb{E}[|Y_j|]\}} = q \cdot \frac{\max\{|X_i|, \mathbb{E}[|Y_j|]\} - q + 1}{\max\{|X_i|, \mathbb{E}[|Y_j|]\}} < q. \tag{16}$$

Specifically, by establishing that $\mathbb{E}[q_{i,j}] < q$, the filter criteria become more stringent and so enhancing the selectivity of the filter.
- The mean value of the expected number of destroyed Q-grams across all tokens $\mathcal{X}$ is denoted $\mathbb{E}[\mathbb{E}[q_{i,j}]]$.
- Due to certain simplifications and estimations made across the collection of records, we introduce the filter sensitivity factor γ within the range $[0, 1]$. This factor is empirically set to be slightly weaker, allowing a balance between high efficiency and precision of the filter.

Under these considerations, the approximation for the Q-gram similarity is derived as follows:

$$\begin{aligned}\mathbb{E}[t_{\mathcal{M}}] &= \sum_{i\in\mathcal{X}} \max\{|X_i|, \mathbb{E}[Y_j]\} - |\mathcal{X}|q + |\mathcal{X}| - \mathcal{L}(\alpha, \lambda) \\ &= \sum_{i\in\mathcal{X}} \max\{|X_i|, \mathbb{E}[Y_j]\} - |\mathcal{X}|q + |\mathcal{X}| - \sum_{i\in\mathcal{X}} \mathbb{E}[q_{i,j}] d(\alpha, |X_i|, \mathbb{E}[Y_j]) \\ &\quad - \mathbb{E}[\mathbb{E}[q_{i,j}]] \cdot \gamma \cdot \lambda \cdot \left(\frac{\alpha(1-\alpha)}{1+\alpha}|\mathcal{X}|\right).\end{aligned} \tag{17}$$

The loss function $d(\alpha, |X_i|, |Y_j|)$, considering the expected value of Y_j, is defined as $d(\alpha, |X_i|, \mathbb{E}[Y_j]) = \frac{1-\alpha}{1+\alpha}(|X_i| + \mathbb{E}[Y_j])$ and Lagrange multiplier $\lambda = -\frac{2(\mathbb{E}[|X_i|]+\mathbb{E}[Y_j])}{\alpha^2+2\alpha+1}$.

This approximation greatly simplifies the original problem. By standardizing the length of tokens in $\mathcal{Y}$ to their expected value and using a consistent similarity threshold, we optimize the Q-gram similarity calculation to $\mathcal{O}(1)$. This method is especially useful when the exact lengths of $|Y_j|$ are unknown or when computational efficiency is a priority.

5 Experiments

In our experiments, we estimate the expected token length as the average length across all $\mathcal{E}$ records, denoting $\mathcal{Y} \in \mathcal{E}$ as $\mathbb{E}[Y_j] = \overline{Y}$ and $\mathcal{X}$ as $\mathbb{E}[X_i] = \overline{X}$. We set the factor γ to 0.75 empirically.

The precision and recall metrics are based on the concepts of true positives, false positives, and false negatives:

- *True Positives (TP):* Pairs of records that are correctly placed in the same clusters and belong to the same entity.
- *False Positives (FP):* Pairs of records that are incorrectly placed in the same cluster but belong to different entities.
- *False Negatives (FN):* Pairs of records that belong to the same entity but are incorrectly placed in different clusters.

Based on the defined terms, Precision and Recall are calculated using the formulas: Precision $= \frac{TP}{TP+FP}$ and Recall $= \frac{TP}{TP+FN}$. The F-Score, which is the harmonic mean of Precision and Recall, is given by: $F-Score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}}$.

These equations represent the standard approach to calculating the accuracy of a process such as clustering, balancing the trade-off between precision and recall. The maximum F-Score is calculated over all thresholds α in range $[0, 1]$. For our clustering analysis, we employ DBSCAN [5] and Nearest Neighbors (NN) as demonstrated in Table 1. The Q-Gram filter achieved a precision of 100%, indicating that no true positive (TP) comparison pairs were erroneously removed, while maintaining a recall of 52%. This high efficiency and filtering capability of the Q-Gram filter are evident, as filter passes only twice as many candidate pairs compared to the fraction of TP comparison pairs.

Similarity	**DBSCAN Max F-score**	**NN Max F-score**
2-Gram Filter + Fuzzy Jaccard	**0.8520**	0.6979
3-Gram Filter + Fuzzy Jaccard	**0.8520**	0.6979
Fuzzy Jaccard	**0.8520**	0.6979
2-Gram Jaccard	0.8248	0.6061
Jaro	0.7917	0.7163
2-Gram Overlap	0.6586	0.7917
3-Gram Jaccard	0.7886	0.6619
3-Gram Overlap	0.6720	0.7405
Jaro-Winkler	0.7333	0.6169
Levenshtein	0.6859	0.5516

Table 1. Sorted Comparison of Max F-scores for DBSCAN and Nearest Neighbor Clustering Algorithms on Labelled Vauniv Dataset (116 Records and 15 Clusters) [2]

6 Conclusion

In this work, we extend Ukkonen's lemma by incorporating an edit constraint for bipartite matching, a notable field contribution. Our main contribution outperforms existing models with an optimal Q-Gram Count filter for bipartite matching. This development, from a mathematically derived lower bound, ensures no loss of true positive (TP) comparison pairs. Given its analytical intractability, we propose a precise estimation method for this filter that operates in constant time complexity $\mathcal{O}(1)$. In our tests, this approach achieved 100% precision in filtering with a high filtering capability. Altogether, the proposed extended count Q-gram filter significantly speeds up the process of similarity join while maintaining high filter efficiency and precision. The record deduplication was efficiently conducted using DBSCAN clustering, which has the significant advantage of being able to form clusters of arbitrary shape while maintaining fast performance, characterized by a time complexity of $\mathcal{O}(|\mathcal{E}| \log(|\mathcal{E}|))$ [5]. It is worth noting that scalability can be achieved by using one of the parallel versions of DBSCAN, as discussed in [3].

Acknowledgment

It was supported by SGS FEI UPCE 2024 and the Erasmus+ project: Project number: 2022-1-SK01-KA220-HED-000089149, Project title: Including EVERyone in GREEN Data Analysis (EVERGREEN) funded by the European Union. Views and opinions

expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the Slovak Academic Association for International Cooperation (SAAIC). Neither the European Union nor SAAIC can be held responsible for them.

References

1. Christen, P.: A survey of indexing techniques for scalable record linkage and deduplication. IEEE transactions on knowledge and data engineering 24(9), 1537–1555 (2012)
2. Cohen, W.W., Ravikumar, P., Fienberg, S.E., et al.: A comparison of string distance metrics for name-matching tasks. In: IIWeb. vol. 3, pp. 73–78 (2003)
3. Dafir, Z., Lamari, Y., Slaoui, S.C.: A survey on parallel clustering algorithms for big data. Artificial Intelligence Review 54, 2411–2443 (2021)
4. Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S.: Duplicate record detection: A survey. IEEE Transactions on knowledge and data engineering 19(1), 1–16 (2007)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: kdd. vol. 96, pp. 226–231 (1996)
6. Jiang, Y., Li, G., Feng, J., Li, W.S.: String similarity joins: An experimental evaluation. Proceedings of the VLDB Endowment 7(8), 625–636 (2014)
7. Jokinen, P., Ukkonen, E.: Two algorithms for approxmate string matching in static texts. In: International Symposium on Mathematical Foundations of Computer Science. pp. 240–248. Springer (1991)
8. Papadakis, G., Skoutas, D., Thanos, E., Palpanas, T.: Blocking and filtering techniques for entity resolution: A survey. ACM Computing Surveys (CSUR) 53(2), 1–42 (2020)
9. Papadakis, G., Svirsky, J., Gal, A., Palpanas, T.: Comparative analysis of approximate blocking techniques for entity resolution. Proceedings of the VLDB Endowment 9(9), 684–695 (2016)
10. Rozinek, O., Borkovcova, M.: Theorems for boyd–wong contraction mappings on similarity spaces. Mathematics 11(20), 4359 (2023)
11. Rozinek, O., Mareš, J.: The duality of similarity and metric spaces. Applied Sciences 11(4) (2021), https://www.mdpi.com/2076-3417/11/4/1910
12. Ukkonen, E.: Approximate string-matching with q-grams and maximal matches. Theoretical computer science 92(1), 191–211 (1992)
13. Vernica, R., Carey, M.J., Li, C.: Efficient parallel set-similarity joins using mapreduce. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data. pp. 495–506 (2010)
14. Wang, J., Li, G., Fe, J.: Fast-join: An efficient method for fuzzy token matching based string similarity join. In: 2011 IEEE 27th International Conference on Data Engineering. pp. 458–469. IEEE (2011)
15. Wang, J., Li, G., Feng, J.: Extending string similarity join to tolerant fuzzy token matching. ACM Transactions on Database Systems (TODS) 39(1), 1–45 (2014)
16. Xiao, C., Wang, W., Lin, X.: Ed-join: an efficient algorithm for similarity joins with edit distance constraints. Proceedings of the VLDB Endowment 1(1), 933–944 (2008)
17. Yang, Z., Yu, J., Kitsuregawa, M.: Fast algorithms for top-k approximate string matching. In: Twenty-Fourth AAAI Conference on Artificial Intelligence (2010)
18. Yu, M., Li, G., Deng, D., Feng, J.: String similarity search and join: a survey. Frontiers of Computer Science 10(3), 399–417 (2016)