Locality Sensitive Hashing with Temporal and Spatial Constraints for Efficient Population Record Linkage

Charini Nanayakkara*
charini.nanayakkara@anu.edu.au
The Australian National University
Canberra, Australia

Peter Christen*
peter.christen@anu.edu.au
The Australian National University
Canberra, Australia

ABSTRACT

Record linkage is the process of identifying which records within or across databases refer to the same entity. Min-hash based Locality Sensitive Hashing (LSH) is commonly used in record linkage as a blocking technique to reduce the number of records to be compared. However, when applied on large databases, min-hash LSH can yield highly skewed block size distributions and many redundant record pair comparisons, where only few of those correspond to true matches (records that refer to the same entity). Furthermore, min-hash LSH is highly parameter sensitive and requires trial and error to determine the optimal trade-off between blocking quality and efficiency of the record pair comparison step. In this paper, we present a novel method to improve the scalability and robustness of min-hash LSH for linking large population databases by exploiting temporal and spatial information available in personal data, and by filtering record pairs based on block sizes and min-hash similarity. Our evaluation on three real-world data sets shows that our method can improve the efficiency of record pair comparison by 75% to 99%, whereas the final average linkage precision can be improved by 28% at the cost of a reduction in the average recall by 4%.

CCS CONCEPTS

• **Information systems → Entity resolution**.

KEYWORDS

Scalability, personal data, spatial constraint, temporal constraint.

ACM Reference Format:
Charini Nanayakkara and Peter Christen. 2022. Locality Sensitive Hashing with Temporal and Spatial Constraints for Efficient Population Record Linkage. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3511808.3557631

1 INTRODUCTION

Record linkage (RL), also known as entity resolution, is the process of identifying records that refer to the same real-world entity within or across databases [3]. Due to its widespread applicability, over the past five decades researchers have developed various methods for conducting RL [29]. Recent research in RL has focused on improving the scalability of linkage algorithms [6, 8, 28, 36], and the applicability of learning based techniques [18, 22, 24, 26, 32, 35].

While RL methods are generally applicable in different domains, they are of significant value in linking data about people (which we refer to as population data [4, 23]) such as those available in the health and government sectors [5, 25]. However, commonly used data sets for RL research are from domains related to publications, consumer products, or movies rather than about people [11, 13]. This is primarily due to privacy concerns which prevent making population data publicly available [9]. Furthermore, population data contain information about *complex entities* [19] that change over time, which requires the development of novel methods for linking such data. As a result, research which explores how RL methods can be applied for public or social good [5] have been limited.

Even those works that do explore population RL often do not exploit characteristics that are inherent to personal data. These include time and space related constraints that can help improve linkage quality [27], such as two siblings cannot be born four months apart, twins are with very high likelihood born at the same location, and the death of a person must occur after their birth and marriage. Another example is the implausibility for a student to concurrently attend two schools in different cities for full-time studies. Developing RL techniques that are tailored to linking population data is important because such types of data often have high ambiguity, while also having shorter attribute values with highly skewed frequency distributions and low data quality [10, 18].

In this paper, we focus on the problem of enhancing the scalability of population RL with temporal and/or spatial constraints available. Different blocking and indexing techniques for RL have been proposed over the years [30], where their aim is to efficiently remove as many true non-matches as possible while retaining (almost) all true matching pairs [7]. Even though numerous 'hand-crafted' techniques exist for blocking, these techniques require domain expertise and extensive knowledge about suitable attributes in the databases to be linked. Such information is often unavailable or limited in real-world applications. Therefore, attribute agnostic methods originally developed for other large-scale data applications such as Web search engines have been adopted for blocking in RL [30]. One widely used approach is min-hash based locality sensitive hashing (LSH) [16, 21], which has been employed for RL blocking for over a decade [12, 17, 33].

Even though min-hash LSH is an effective blocking method, tuning its parameters, the band size and number of bands [21], is difficult due to their sensitivity with regard to how many blocks are

*The authors are also affiliated with the Scottish Centre for Administrative Data Research (SCADR) at the University of Edinburgh, UK.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
CIKM '22, October 17–21, 2022, Atlanta, GA, USA
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9236-5/22/10.
https://doi.org/10.1145/3511808.3557631

Table 1: Blocking quality achieved when applying basic min-hash LSH on the data sets described in Table 2. Because ground truth data are not available for the BHIC data set, its number of false negatives (FN) cannot be calculated.

LSH number of bands	LSH band size	Isle of Skye (IOS)			Kilmarnock (KIL)			Brabant Historical Information Center (BHIC)	
		Num. pairs	Num. FN	Max. / avr. block sizes	Num. pairs	Num. FN	Max. / avr. block sizes	Num. pairs	Max. / avr. block sizes
25	4	7,650,418	251	849 / 5.0	75,301,860	566	4,661 / 6.4	> 2 Billion	37,704 / 15.3
25	6	1,302,842	888	483 / 2.9	8,245,291	980	1,418 / 3.1	482,695,380	7,068 / 5.1
25	10	90,637	3,686	54 / 2.1	330,702	2,691	249 / 2.1	9,390,284	722 / 2.8
50	4	15,217,162	52	1,669 / 5.2	101,684,899	479	3,429 / 6.5	> 3 Billion	25,447 / 13.6
50	6	2,423,487	502	625 / 2.9	17,844,234	616	1,645 / 3.2	479,332,067	5,972 / 4.7
50	10	147,222	2,282	71 / 2.1	747,008	1,769	202 / 2.1	15,918,082	784 / 2.8
100	4	22,666,733	15	2,129 / 5.1	179,175,757	457	5,169 / 6.9	> 6 Billion	29,059 / 13.6
100	6	4,183,730	188	713 / 2.9	33,255,513	538	1,636 / 3.2	1,088,031,531	6,335 / 4.7
100	10	265, 950	1,494	117 / 2.1	1,407,488	1,304	305 / 2.1	23,545,938	1,085 / 2.8

Table 2: Data sets used for experiments in Section 3. The GT column indicates which data sets contain ground truth data.

Data set	Description	GT
Isle of Skye (IOS)	A Scottish population data set from the Isle of Skye containing 17,613 birth records over the period from 1861 to 1901	✓
Kilmarnock (KIL)	A Scottish population data set from the town of Kilmarnock containing 37,121 birth records over the period from 1861 to 1901	✓
BHIC	A Dutch population data set from the Brabant Historical Information Center containing 830,616 birth records from 1762 to 1919	×

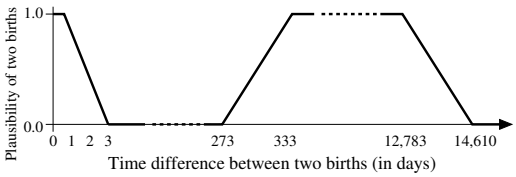

Figure 1: Temporal constraints as the plausibility for a mother to be able to give birth to two children. The vertical axis shows the plausibility of two births by the same mother for a certain time difference, as we discuss in Section 2.

Algorithm 1: Blocking and iterative classification

Input: $\mathbf{D}$: Data set with records to be linked
$\mathbf{T}$: List of plausible time ranges
$\mathbf{P}_T, \mathbf{P}_S$: Temporal and spatial plausibility indices
b, r: Number of bands and band size for min-hash LSH
ρ, ρ': Thresholds for block filtering ($\rho \geq \rho'$)
Δ: Threshold for score filtering
δ_v, δ_m: Thresholds for similarity filtering ($\delta_v \geq \delta_m$)
α, β: Weights for temporal and spatial plausibilities
Output: $\mathbf{V} \cup \mathbf{M}$: Set of classified matches obtained with blocking

$\mathbf{V} = \{\}, \mathbf{M} = \{\}$
$\mathbf{L} = \mathit{MinHashLSHIndexing}(\mathbf{D}, b, r)$; $\mathbf{R} = \mathit{GenInvRecIndex}(\mathbf{L})$
$\mathbf{L}, \mathbf{R} = \mathit{BlockSizeFiltering}(\mathbf{L}, \mathbf{R}, \rho)$; $\mathbf{C} = \mathit{BlockSimFiltering}(\mathbf{L}, \mathbf{R}, \rho')$
for $(t_{start}, t_{end}) \in \mathbf{T}$ **do**
  $\mathbf{C}' = \mathit{GetPairsInTempRange}(\mathbf{C}, t_{start}, t_{end})$
  for $(r_i, r_j) \in \mathbf{C}'$ **do**
    if $\exists r_x \in \mathbf{D} : ((r_i, r_x) \in \mathbf{V}$ ***and*** $\mathit{IsNonTemporal}((r_j, r_x), \mathbf{P}_T))$ ***or*** $((r_j, r_x) \in \mathbf{V}$ ***and*** $\mathit{IsNonTemporal}((r_i, r_x), \mathbf{P}_T))$ **then continue**
    if $\exists r_y \in \mathbf{D} : ((r_i, r_y) \in \mathbf{V}$ ***and*** $\mathit{IsNonSpatial}((r_j, r_y), \mathbf{P}_S))$ ***or*** $((r_j, r_y) \in \mathbf{V}$ ***and*** $\mathit{IsNonSpatial}((r_i, r_y), \mathbf{P}_S))$ **then continue**
    $p_t, p_s = \mathit{GetTempAndSpatialPlausibility}((r_i, r_j), \mathbf{P}_T, \mathbf{P}_S)$
    $p_a = \mathit{CalcBlockSim}((r_i, r_j), \mathbf{C}', b)$
    if $(\alpha \cdot p_t + \beta \cdot p_s + (1 - (\alpha + \beta)) \cdot p_a) \geq \Delta$ **then**
      $s_{i,j} = \mathit{GetPairwiseSimilarity}(r_i, r_j)$
      if $s_{i,j} \geq \delta_v$ **then** $\mathbf{V}.add((r_i, r_j))$
      else if $s_{i,j} \geq \delta_m$ **then** $\mathbf{M}.add((r_i, r_j))$

generated and their sizes [1]. As we show in Table 1, for different LSH parameter settings we obtain vastly different blocking results with considerable variations in the number of record pairs generated, the number of missed true matches (false negatives), and the range of block size distributions. These are due to the highly skewed frequency distributions of values (such as names and addresses) in population data, that can result in a very large numbers of record pairs even for small data sets in the population RL context.

To overcome this challenge, we develop a novel min-hash LSH blocking method which filters record pairs based on block size and min-hash similarity, and incorporates temporal and spatial constraints that are commonly available in population data.

2 IMPROVED MIN-HASH LSH FOR BLOCKING

While min-hash based LSH is an efficient blocking technique that can produce blocks with high recall because it inserts each record into multiple blocks [21], for large data sets this can result in very large blocks. Furthermore, LSH is a technique that can be highly sensitive to parameter settings [1].

To mitigate these limitations of min-hash LSH, we incorporate temporal and spatial constraints, which indicate the plausibility for a record pair to be linked based on time (such as the biologically possible time ranges for births by the same mother), and geographic distance related constraints (such as the higher plausibility for the families of a bride and groom to live in proximity), as commonly available in population data [27]. We employ two techniques to reduce the number of record pair comparisons in the blocks generated by LSH. In the first, we remove each record from a given proportion of the largest blocks it occurs in [31]. The intuition here is that larger blocks more often contain redundant record pairs which are also included in smaller blocks. In the second technique, we only retain those record pairs which occur together in multiple blocks (have high block or min-hash similarity), since this indicates higher attribute similarity of record pairs [2].

Algorithm 1 outlines our overall method. As input to the algorithm, we provide a population data set $\mathbf{D}$ with records to be linked, a list of time ranges $\mathbf{T}$ which are deemed to be plausible according to the linkage application, and indexes $\mathbf{P}_T$ and $\mathbf{P}_S$ containing the temporal and spatial plausibility values respectively. Figure 1 shows an example for temporal constraints, where the plausible time ranges is the list $\mathbf{T} = [(0, 3), (273, 14610)]$ of day differences. The temporal plausibility index $\mathbf{P}_T$, however, contains more fine-grain plausibility

Algorithm 2: Size filtering (function *BlockSizeFiltering*)

Input: **L**: Min-hash based LSH blocks
R: Inverted index of LSH blocks and their sizes per record
ρ: Number of smallest blocks to retain records in
Output: **L**, **R**: Filtered min-hash LSH blocks and inverted index

1 **for** $r \in \mathbf{R}$ **do**
2 $\quad \mathbf{k} = GetLargestBlockKeys(\mathbf{R}[r], |\mathbf{R}[r]| - \rho)$
3 $\quad RemoveRecFromBlocks(\mathbf{L}, \mathbf{k}, r)$
4 $\quad RemoveBlocksForRec(\mathbf{R}[r], \mathbf{k})$

Algorithm 3: Similarity filtering (function *BlockSimFiltering*)

Input: **L**: Min-hash based LSH blocks filtered by size
R: Filtered inverted index
ρ': Threshold number of blocks per record pair
Output: **C**: Candidate record pair index

1 $\mathbf{C} = \{\}, \mathbf{O} = \{\}$
2 **for** $\mathbf{b} \in \mathbf{L}$ **do**
3 $\quad$ **for** $(r_i, r_j) \in \mathbf{b}, i < j$ **do**
4 $\quad\quad$ **if** $r_j \notin \mathbf{O}[r_i]$ **and** $r_j \notin \mathbf{C}[r_i]$ **then**
5 $\quad\quad\quad c = |\mathbf{R}[r_i] \cap \mathbf{R}[r_j]|$
6 $\quad\quad\quad$ **if** $c \geq \rho'$ **then** $\mathbf{C}[r_i] = \mathbf{C}[r_i] \cup \{(r_j, c)\}$
7 $\quad\quad\quad$ **else** $\mathbf{O}[r_i] = \mathbf{O}[r_i] \cup \{r_j\}$

values, such as a plausibility of 1.0 for 0 day difference, 0.66 for 1 day difference, 0.33 for 2 days difference, and so on. Similarly, the spatial plausibility index $\mathbf{P}_S$ contains linkage plausibilities corresponding to geographic distances between the records in a pair (such as the distance of birth locations of siblings).

Input parameters b and r specify the number of bands and band-sizes to be used in the LSH blocking algorithm. We also provide threshold values (ρ, ρ'), Δ, and (δ_v, δ_m) for block filtering, score filtering, and average attribute value similarity filtering, respectively, and weights (α, β) to be assigned to temporal and spatial plausibility values in the overall score calculation, as we discuss below.

In line 1 of Algorithm 1, two sets **V** and **M** (where **V**∪**M** comprises the output of the algorithm) are initialised to contain the matches with *very high* and *moderately high* confidence. We use two similarity thresholds δ_v and δ_m to classify matches, where $\delta_v (> \delta_m)$ helps to filter the very high confident matches **V**. These very high confident matches, together with temporal and spatial constraints, are used for early identification of likely non-matches which are eliminated from the comparison step to improve efficiency. In line 2 we apply min-hash based LSH to generate a blocking index **L**, and generate an inverted record index **R** for **L** which for each record $r \in \mathbf{D}$ contains their blocking keys and the corresponding block sizes. With b bands used for LSH, each record is placed into b blocks with very high probability [21].

Next, we apply block size filtering, where we retain each record in the $\rho \leq b$ smallest blocks, and update the LSH block index **L** and the record index **R** as we discuss in Algorithm 2. We then apply filtering based on block similarity as we describe in Algorithm 3, where only those record pairs that occur in at least $\rho' \leq b$ common blocks in **L** are returned as candidate record pairs in the set **C**.

The iterative processing of record pairs in **C** is then conducted in lines 4 to 14. To improve the scalability of our method to large data sets, in each iteration we retrieve a subset of candidate record pairs $\mathbf{C}'$ (line 5) within a given plausible time range $(t_{start}, t_{end}) \in \mathbf{T}$. Next, starting from line 6, we process each record pair $(r_i, r_j) \in \mathbf{C}'$, where in lines 7 and 8 we initially discard pairs which are inconsistent with regard to temporal or spatial constraints, and with previously identified very high confidence matches in **V**.

For each consistent record pair (r_i, r_j), in line 9 we then calculate the temporal and spatial plausibility values p_t and p_s, based on pre-generated indexes reflecting a domain expert's knowledge of overall plausibilities (such as the plausibility of siblings being born twenty years apart to be much less likely than them being born two years apart). In line 10 we calculate the block (min-hash) similarity p_a of a record pair based on the number of blocks they occur in common, and in line 11 we check whether the weighted overall score of p_t, p_s and p_a is at least a given threshold Δ. If this score threshold condition is satisfied, in line 12 we calculate the average attribute value similarity $s_{i,j}$ for the record pair (r_i, r_j). The constraint-based checks in lines 7 and 8, and the plausibility and block similarity calculations in lines 9 and 10 are inexpensive, index-based computations. Given the record pair comparison step is computationally expensive [7, 29], reducing the number of comparisons in lines 7 and 8 using constraints, and the score check in line 11, substantially improve the efficiency of our method.

In lines 13 and 14, we classify a record pair as a very high confidence match, **V**, if its similarity $s_{i,j}$ is at least δ_v, or a moderately high confidence match, **M**, if its similarity $s_{i,j}$ is at least δ_m.

Algorithm 2 outlines the functionality of the *BlockSizeFiltering* function used in line 3 of Algorithm 1. As input we provide an index of min-hash LSH blocks **L**, the corresponding inverted record index **R**, and the threshold number of blocks to retain records in, ρ.

We iterate over each record $r \in \mathbf{R}$, and in line 2 obtain the set of keys **k** of the largest $|\mathbf{R}[r]| - \rho$ blocks (where $|\mathbf{R}[r]| = b$) corresponding to r. In line 3, we remove record r from all blocks in the LSH blocking index **L** with blocking keys in the set **k**. Similarly, in line 4, we remove the blocking key and block size pairs from the record index **R** that correspond to the blocking keys in **k**.

Algorithm 3 details the function *BlockSimFiltering* as used in line 3 of Algorithm 1. As input we provide the filtered indices **L** and **R** (output of Algorithm 2), and ρ' indicating the threshold number of blocks a record pair should appear in for it to be considered a candidate. In line 1 we initialise two empty indices, **C** and **O**, to hold the candidate and non-candidate record pairs, respectively. In lines 2 to 7, we iteratively process each block $\mathbf{b} \in \mathbf{L}$, and every record pair in a block $(r_i, r_j) \in \mathbf{b}$ (where $i < j$). If this record pair has not been processed previously, in line 5 we obtain the number of common blocks c in which records r_i and r_j appear. If the two records occur together in at least ρ' blocks, then the pair is added to the candidate index **C** with the common block count c in line 6. Otherwise, the record pair is added to the non-candidate index **O**.

3 EXPERIMENTAL EVALUATION

We conducted an evaluation of the quality and efficiency of our proposed method using the three data sets shown in Table 2, where the task is to link all birth records by the same mother [4, 34].

For the basic min-hash LSH step in Algorithm 1, we set the number of bands $b = 50$ and the band size $r = 4$ for the IOS and KIL data sets since these settings resulted in few false negatives (FN)

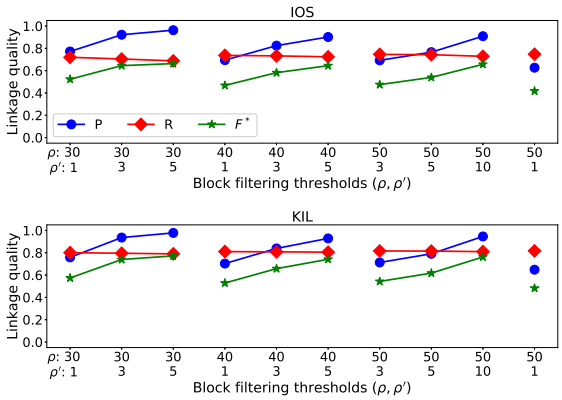

Figure 2: Final linkage quality achieved with different block filtering thresholds, as discussed in Section 3.

while generating substantially less record pairs compared to using the settings $b = 100$ and $r = 4$ which produced the smallest number of FN, as shown in Table 1. However, for the BHIC data set we used $b = 100$ and $r = 6$ since the $b = 50$ and $r = 4$ settings generated more than 3 billion record pairs, as can be seen in Table 1.

We use precision, recall, and the F^* measure (an interpretable variation of the F-measure [14]) to assess the linkage quality, where $F^* = F/(2 - F)$ [15]. Figure 2 shows the final linkage quality achieved with our method for different block filtering threshold pairs (ρ, ρ'), where $\rho = 50$ and $\rho' = 1$ presents the quality achieved with the baseline method excluding block filtering (line 3 of Algorithm 1). As shown in Figure 2, the final linkage quality exceeds or is on par with the quality of the baseline, while achieving a significant improvement in the efficiency as is evident from the reduction in the number of pair-wise comparisons shown in Table 3. The reduction of redundancy with block filtering led to an increase of precision (less false positives) while recall stayed stable.

As we show in Figure 3, our method is robust to changes in the score filtering threshold Δ, and the plausibility weights α, β, and γ, as indicated by the limited changes in the corresponding F^* values. We achieved the best linkage quality for both the IOS and KIL data sets with the similarity thresholds $\delta_m = 0.7$ and $0.8 \leq \delta_v \leq 0.9$. As Table 3 shows, we achieved a significant reduction in the number of record pair comparisons (of at least 75%) generated by our method ($|\mathbf{C}|$) compared with the numbers resulting from basic min-hash LSH. A further 11% reduction in comparisons can be achieved with the iterative temporal and spatial filtering conducted in Algorithm 1, as per the results obtained for the BHIC data set.

Based on our evaluation, we can recommend choosing values in the range [0.7, 0.9] for the similarity filtering thresholds δ_v and δ_m, with $\delta_v > \delta_m$. Our approach performs well when the threshold number of blocks per record pair ρ' is within the range [2, 6].

4 DISCUSSION AND FUTURE WORK

We have presented a novel method for improved min-hash LSH aimed at linking population data. We exploit both temporal and spatial constrains to substantially improve the scalability of RL applications of such types of data by filtering redundant as well as less similar record pairs, and preventing many comparisons of likely non-matches. Our experiments on two real data sets have shown that our method can reduce the number of record pair comparisons around 10-fold with no decline in the average recall, or more than 100-fold with a drop of the average linkage recall by only 4%.

In the future we aim to explore how errors in temporal and spatial constraints can impact the performance of our method. We also plan to conduct an ablation study to explore how the different filtering techniques contribute to the overall efficiency enhancement.

We presented initial work in an ongoing project that aims to develop scalable RL techniques for the automated (unsupervised) linking of personal data at the scale of full populations. The full reconstruction of (historical) populations [4] will allow a breadth of research studies that are currently impossible, such as genetic studies of hereditary diseases over many generations [20].

Table 3: The numbers of unique record pair comparisons and corresponding percentage reductions achieved with our blocking technique compared to basic min-hash LSH.

Block filtering thresholds (ρ, ρ')	IOS $(b = 50, r = 4)$ (LSH: 15,217,162)	KIL $(b = 50, r = 4)$ (LSH: 101,684,899)	BHIC $(b = 100, r = 6)$ (LSH: 1,088,031,531)
(100, 2)	-	-	121,037,748 (89%)
(50, 2)	3,804,882 (75%)	21,159,017 (79%)	1,554,921 (99%)
(40, 1)	2,157,206 (86%)	12,095,120 (88%)	2,213,576 (99%)
(30, 1)	552,020 (96%)	2,793,411 (97%)	1,180,213 (99%)
(50, 5)	371,148 (98%)	855,492 (99%)	1,025,202 (99%)
(40, 5)	77,839 (99%)	123,795 (99%)	924,654 (99%)
(30, 5)	43,652 (99%)	72,471 (99%)	849,550 (99%)

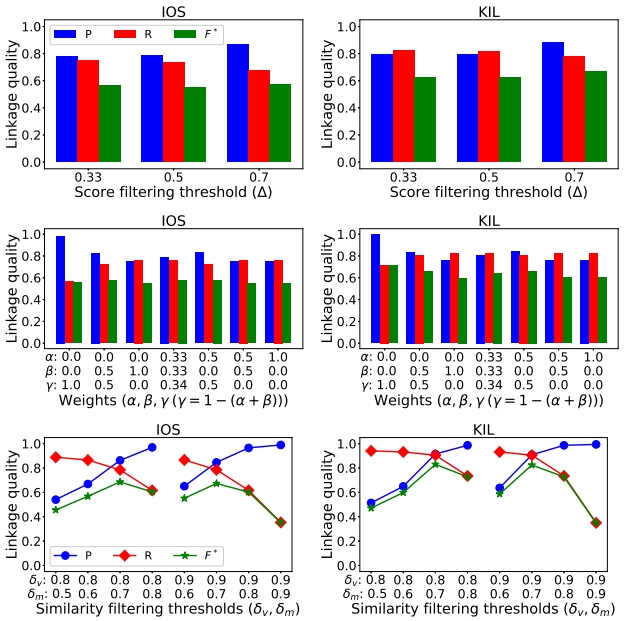

Figure 3: Final linkage quality achieved with different score and similarity filtering thresholds, and different weights.

ACKNOWLEDGMENTS

This work was partially funded by the UK Economic and Social Research Council under grant number ES/W010321/1.

REFERENCES

[1] Özgür Akgün, Alan Dearle, Graham Kirby, and Peter Christen. 2018. Using metric space indexing for complete and efficient record linkage. In *PAKDD*. Springer, Melbourne, 89–101.

[2] Roberto J. Bayardo, Yiming Ma, and Ramakrishnan Srikant. 2007. Scaling up all pairs similarity search. In *WWW*. ACM, Banff, Canada, 131–140.

[3] Olivier Binette and Rebecca C Steorts. 2022. (Almost) all of entity resolution. *Science Advances* 8, 12 (2022), eabi8021.

[4] Gerrit Bloothooft, Peter Christen, Kees Mandemakers, and Marijn Schraagen. 2015. *Population Reconstruction.* Springer, Heidelberg.

[5] Emma L. Brook, Diana L. Rosman, and C. D'Arcy J. Holman. 2008. Public good through data linkage: measuring research outputs from the Western Australian data linkage system. *Australian and New Zealand Journal of Public Health* 32, 1 (2008), 19–23.

[6] L. Soares Caldeira, Guilherme Dal Bianco, and Anderson A Ferreira. 2021. Experimental Evaluation Among Reblocking Techniques Applied to the Entity Resolution. In *European Conference on Advances in Databases and Information Systems*. Springer, Tartu, Estonia, 229–243.

[7] Peter Christen. 2012. *Data Matching – Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Springer, Heidelberg.

[8] Peter Christen, Ross Gayler, and David Hawking. 2009. Similarity-Aware Indexing for Real-Time Entity Resolution. In *CIKM*. ACM, Hong Kong, 1565–1568.

[9] Peter Christen, Thilina Ranbaduge, and Rainer Schnell. 2020. *Linking Sensitive Data.* Springer, Heidelberg.

[10] Peter Christen and Rainer Schnell. 2021. Common Misconceptions about Population Data. *arXiv preprint arXiv:2112.10912* (2021).

[11] Xin Luna Dong. 2018. Challenges and Innovations in Building a Product Knowledge Graph. In *SIGKDD*. ACM, London, 2869.

[12] Elizabeth Ashley Durham. 2012. *A Framework for Accurate, Efficient Private Record Linkage.* Ph. D. Dissertation. Faculty of the Graduate School of Vanderbilt University, Nashville, TN.

[13] Anja Gruenheid, Xin Luna Dong, and Divesh Srivastava. 2014. Incremental Record Linkage. *Proc. VLDB Endowment* 7, 9 (2014), 697–708.

[14] David Hand, Peter Christen, and Nishadi Kirielle. 2021. F*: an interpretable transformation of the F-measure. *Machine Learning* 110 (03 2021), 451–456.

[15] David J Hand and Peter Christen. 2018. A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing* 28, 3 (2018), 539–547.

[16] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*. ACM, Dallas, 604–613.

[17] Dimitrios Karapiperis, Aris Gkoulalas-Divanis, and Vassilios S. Verykios. 2017. FEDERAL: A Framework for Distance-Aware Privacy-Preserving Record Linkage. *Transactions on Knowledge and Data Engineering* 30, 2 (2017), 292–304.

[18] Nishadi Kirielle, Peter Christen, and Thilina Ranbaduge. 2022. TransER: Homogeneous Transfer Learning for Entity Resolution. In *EDBT*. Edinburgh, 118–130.

[19] Nishadi Kirielle, Peter Christen, and Thilina Ranbaduge. 2022. Unsupervised Graph-Based Entity Resolution for Complex Entities. *Transactions on Knowledge Discovery from Data* (2022), 29 pages.

[20] Nishadi Kirielle, Charini Nanayakkara, Peter Christen, Chris Dibben, Lee Williamson, Eilidh Garrett, and Clair Manson. 2022. Unsupervised Graph-based Entity Resolution for Accurate and Efficient Family Pedigree Search. In *EDBT*. Edinburgh, 498–510.

[21] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2014. *Mining of Massive Datasets.* Cambridge University Press, Cambridge.

[22] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proc. VLDB Endowment* 14, 1 (2020), 50–60.

[23] Kim McGrail, Kerina Jones, Ashley Akbari, Tellen D Bennett, Andy Boyd, et al. 2018. A Position Statement on Population Data Science: The science of data about people. *International Journal of Population Data Science* 3, 1 (2018), 11 pages.

[24] Venkata V. Meduri, Lucian Popa, Prithviraj Sen, and Mohamed Sarwat. 2020. A Comprehensive Benchmark Framework for Active Learning Methods in Entity Matching. In *SIGMOD*. ACM, Portland, 1133–1147.

[25] Miranda J. Mourby, James Doidge, Kerina H. Jones, Ruth Gilbert, Stergios Aidinlis, Hannah Smith, Jessica Bell, Peter Dutey-Magni, and Jane Kaye. 2019. Health Data Linkage for Public Interest Research in the UK: Key Obstacles and Solutions. *International Journal of Population Data Science* 4, 1 (2019), 13 pages.

[26] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, et al. 2018. Deep learning for entity matching: A design space exploration. In *SIGMOD*. ACM, Houston, 19–34.

[27] Charini Nanayakkara, Peter Christen, and Thilina Ranbaduge. 2018. Temporal graph-based clustering for historical record linkage. In *MLG, held at SIGKDD*. ACM, London, 9 pages.

[28] Charini Nanayakkara, Peter Christen, and Thilina Ranbaduge. 2021. Active Learning Based Similarity Filtering for Efficient and Effective Record Linkage. In *PAKDD*. Springer, Delhi, 321–333.

[29] George Papadakis, Ekaterini Ioannou, Emanouil Thanos, and Themis Palpanas. 2021. The Four Generations of Entity Resolution. *Synthesis Lectures on Data Management* 16, 2 (2021), 1–170.

[30] George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. 2020. Blocking and Filtering Techniques for Entity Resolution: A Survey. *ACM CSUR* 53, 2 (2020), 1–42.

[31] George Papadakis, Jonathan Svirsky, Avigdor Gal, and Themis Palpanas. 2016. Comparative analysis of approximate blocking techniques for entity resolution. *Proc. VLDB Endowment* 9, 9 (2016), 684–695.

[32] Anna Primpeli and Christian Bizer. 2021. Graph-boosted active learning for multi-source entity resolution. In *ISWC*. Springer, Virtual, 182–199.

[33] Thilina Ranbaduge, Dinusha Vatsalan, Peter Christen, and Vassilios S. Verykios. 2016. Hashing-Based Distributed Multi-party Blocking for Privacy-Preserving Record Linkage. In *PAKDD*. Springer, Auckland, 415–427.

[34] Alice Reid, Ros Davies, and Eilidh Garrett. 2002. Nineteenth-century Scottish demography from linked censuses and civil registers: A 'sets of related individuals' approach. *History and Computing* 14, 1–2 (2002), 61–86.

[35] Kai-Sheng Teong, Lay-Ki Soon, and Tin Tin Su. 2020. Schema-Agnostic Entity Matching Using Pre-Trained Language Models. In *CIKM*. ACM, Galway, 2241–2244.

[36] Wei Zhang, Hao Wei, Bunyamin Sisman, Xin Luna Dong, Christos Faloutsos, and Davd Page. 2020. AutoBlock: A Hands-off Blocking Framework for Entity Matching. In *WSDM*. ACM, Houston, 744–752.

APPENDIX: COMPLEXITY ANALYSIS

The basic min-hash LSH indexing conducted in line 2 of Algorithm 1 has a time complexity of $O(|\mathbf{D}| \cdot b)$ since each record is permuted b times. The inverted index generation step in line 3, and the block size filtering step in line 4 have a maximum time complexity of $O(|\mathbf{D}|)$ each. The time complexity of the block similarity filtering step is $O(|\mathbf{b}^2| \cdot |\mathbf{L}|)$ where $\mathbf{b} \in \mathbf{L}$. The complexity of the for loop from line 6 to 16 is $O(|\mathbf{T}| \cdot |\mathbf{C}'|)$, since each step from line 9 to 16 has constant time complexity $O(1)$. The total time complexity of our algorithm is therefore $O(|\mathbf{D}| \cdot b + |\mathbf{b}^2| \cdot |\mathbf{L}| + |\mathbf{T}| \cdot |\mathbf{C}'|)$.