# The Use of Reference Graphs in the Entity Resolution of Criminal Networks

David Robinson(✉)

Inland Revenue, Wellington, New Zealand
david.robinson@ird.govt.nz

**Abstract.** Entity resolution (ER) is the detection of duplicated records within a dataset representing the same real-world entity. The importance of ER is amplified within law enforcement as criminal data, or criminal networks, has inherent uncertainty and ER inaccuracy incurs a high cost. Commercial ER solutions focus on fast and scalable resolution of obvious pairs of entities, rather than the more complex non-obvious pairs which are so critical to law enforcement. Here we outline the use of proper names represented as reference graphs - generated from an algorithm that conducts name similarity, logic-based pruning, and classification using community detection and a proper name origin algorithm. The resultant classes are used at indexing and decision management stages within an ER model to support the detection of non-obvious duplicate entities. Utility is clearly demonstrated through the application of the approach on three real-world datasets of varying origin, size, topology, and heterogeneity.

**Keywords:** Entity resolution · Record linkage · Reference graph · Criminal networks · Indexing · Decision management · Community detection

## 1 Introduction

Criminal networks - graph representations focusing on criminal actors - present significant challenges in terms of deriving an accurate representation that mimics the real-world. Incompleteness, data heterogeneity, non-intentional error, intentional misinformation, and bias all contribute to increase the uncertainty of the data. At the core of this uncertainty and variance is accurately and reliably resolving duplicate entities within the data representation which in fact represent the same real world entity [1] - entity resolution.

Whether the problem is derived from the integration of multiple heterogeneous datasets or focuses on one homogeneous dataset the domain dictates the nature and complexity of the problem, which in turn places specific demands of an entity resolution solution. This complexity can be driven from artefacts of the source(s) of data and their representation or the wider domain where data error is generated from not only incidental variance but variance derived from specific intent. Interestingly within the criminal domain the very entities that are the source of intentionally poor quality data are often the very entities that are of most interest.

M. Chau et al. (Eds.): PAISI 2016, LNCS 9650, pp. 3–18, 2016.
DOI: 10.1007/978-3-319-31863-9_1

The criminal context provides an additional layer of complexity and uncertainty due to the motivation of entities to actively supply misinformation with the goal to reduce the effectiveness of entity resolution. For this reason entity resolution and link discovery are often deployed in concert to enhance the quality of the graph through making the data as explicit as possible and discover latent knowledge. This paper however is limited to entity resolution, and in particular the use of reference graphs in entity resolution. A critical element though to highlight is that entity resolution in the criminal domain must be able to contend with not just missing nodes and edges, but the existence of fake nodes, nodes that are in the dataset but do not exist in the real world, and spoof nodes, where a real world node will be represented in multiple nodes within the dataset [2]. And of course in instances of high incompleteness, the presence of fake and spoof nodes (and edges), and high uncertainty the generation of false positives can significantly obfuscate the graph.

Therefore, inexpensive, accurate and scalable approaches to entity resolution that go beyond identifying the obvious matches (deduplication) and can also detect the non-obvious matches are of critical importance. Current “state of the art” commercial entity resolution products are often focused on markets that require generic scalable fast deduplication solutions and do not place the requisite emphasis on the detection of the complex low-signal non-obvious matches. Responding to this need the reference graph algorithm has been designed to support the detection of non-obvious duplicate entities.

A reference graph, in this instance, is defined as a graph constructed from a set of proper names whose pairwise distance is calculated using a variety of concepts, including string similarity and co-occurrence, and represented as a graph that can be improved over time to enhance ER performance. Improvement to the reference graph can be derived from improvements in the algorithm that constructs the graph, the integration of additional data, or the manual annotation by human experts. The reference graph algorithm generates meta-data that can be used in both indexing and decision management stages of an entity resolution model that out-performs more traditional algorithms on typical criminal networks, is scalable (≈4 million nodes) and “fast enough”.

From an applied perspective two main elements of entity resolution are critical to an effective and performant solution: indexing (otherwise known as blocking or key-generation) and decision management - making a decision on whether to resolve a pair of entities or not. These two components of entity resolution will be briefly introduced to create context required before reference graphs are explained further.

Indexing is essentially the creation of subsets of records or entities based on some notion of similarity. The comparison of all pairs is an intractable problem and hence indexing has been a pragmatic solution to avoid exhaustive comparison and reduce the computational expense by breaking the initial set into multiple sub-sets or blocks. The number, size and the “similarity” between entities within each sub-set determine the quality of the indexing, as each block, or cluster of blocks, will serve as the set that pairwise similarity will be measured. The quality in combination with runtime, scalability, ease of optimization, and versatility (how well the indexing performs across a range of different scenarios) determine the utility of the indexing. Many approaches have been used to generate blocks including the use of phonetic algorithms like Soundex [3], Double-Metaphone [4], and Metaphone 3 [5] which generate keys based on the phonetic sound of the name (e.g. Metaphone 3 of "Robinson" == "RPNS"). Although many of the latest generation algorithms are proprietary, making them problematic to benchmark against, a wide range of quality algorithms exist and are freely available to apply. Truncation approaches similarly generate keys off a predetermined number of letters from the front of the Family Name (e.g. "Robinson" == "ROB"). Suffix Array is another blocking approach that is used, which creates an integer key based on lexicographically sorted string suffixes [6]. Meta-blocking approaches are another class of approach that takes the output from a blocking strategy and attempts to optimize given the tolerance for error and speed. A good example of a meta-blocking strategy is that outlined by Hernandez and Stolfo [7, 8] and McCallum, Nigam and Ungar [9] using windows or canopies to effectively create overlapping classes to reduce computational expense and yet retain accuracy.

Decision management is about making decisions under uncertainty, given the context of the purpose of making those decisions and validation metrics [10]. The first element within ER to achieve this is the discovery of a number of relevant pairwise similarity metrics (e.g. Name similarity, Date of Birth similarity, and distance), measuring key concepts. The pairwise similarity metrics fall under two concepts, those helping to measure congruence and commonality. Congruence refers to the holistic assessment of how similar the pair is based on name features, other attribute features (e.g. Date of Birth; Gender) and contextual features such (e.g. graph distance; community membership) which have all shown to significantly improve performance [11–13]. Commonality refers to how often these features are represented in different entities within the bounded context of the comparison. If the pairwise similarity is conducted globally across the entire dataset without any notion of distance (for example, geographical or social relationship) between the pair then the commonality measurement has to be based on a global assessment. However, incorporating the notion of distance can create a bounded local context which significantly alters the measurement of commonality. For example, the certainty of the pairwise assessment of whether "Joan Mary SMITH" and "Joan Mary SMITH" are indeed the same real-world entity is significantly increased if it is known that they reside in the same suburb or community detection has identified they are members of the same community. The second component is factoring in the context of what the ER is being conducted for (for example tax evasion detection or counter terrorism). Central concepts from a contextual perspective include how rare the class of events are that are the focus of detection and measurement, and the size of impact, and then how that translates into the cost of false positives and false negatives. The third aspect is the generation, retention, and use of contextual validation metrics of the decision made - a critical element to decision management [14]. In this case the ER model computes transitivity providing a useful guide to the measurement of accuracy of the overall model, each ER function, and at a pair level. Indeed, generating transitivity metrics creates the opportunity for fine-grained transitive closure based approaches to enhance performance [15]. All relevant discovered metrics are then output in graph format enabling robust validation and exploitation of the model.

As alluded to earlier a core feature of the broader entity resolution model developed is the use of reference graphs to drive indexing and support decision management.

Reference graphs are an explicit representation of proper name knowledge derived from the data and from external knowledge sources. Knowledge representations are often used as a “deterministic” adjunct to bolster accuracy through the provision of a set of relationships between names based on synonyms and hypocorisms [16]. Here the novel generation of reference graphs from proper names, derived from both the data source and external proper name based edge-lists of hypocorisms, has been used to generate blocking-keys from a simple partitioning of the reference graphs using non-overlapping community detection coupled with classification derived from a proper name co-occurrence graph, which are used at both indexing and decision management stages. An important feature of the indexing is that the keys form a graph, which creates the opportunity for deploying meta-blocking strategies to optimize performance [17].

The details of how the reference graphs are constructed and implemented are covered before the experimental conditions are outlined and the results thereof are examined. A discussion of the results, conclusions and extensions complete the investigation.

## 2 Generation and Implementation of Reference Graphs

### 2.1 Reference Graph Generation

The proper name reference graphs, Family Name Reference Graph (FNRG) and Given Name Reference Graph (GNRG), are generated by measuring the string distance, using the Jaro-Winkler algorithm [18], between all names of the same class, whether that is the Family name class or Given name class. The proper names are sourced from the target unresolved dataset, and potentially any other source. Dependent on the size of the target unresolved dataset there may need to be an intermediate blocking phase to generate the reference graph. The intermediate blocking phase is implemented by a two-step algorithm that firstly blocks the names by the first letter of each name and compares all names beginning with that letter and uses the string distance to start building the reference graph. The second step of the algorithm generates a sample of names and conducts string distance on all pairs from the sample. The complete graph derived from this process is then pruned via using a simple threshold to remove edges and enable the assessment of relationship strength between names starting with a specific letter (see Fig. 1). This derived graph served as the basis in the second stage of the algorithm to select meta-blocks (as per the circled clusters) from which to base additional string distance comparisons.

The result of this intermediate blocking phase is an approximately complete graph based on the distance between proper names, either Family names as in the FNRG or Given names as in the GNRG.

The next step is to turn the distance graph into a similarity graph (so more intuitive) and only retain edges between proper names that are useful to the ultimate goal of conducting community detection that is both accurate and inexpensive. A simple thresholding method could be used to delete edges under a certain weight however that would mean that unique names are more likely to be isolated and not part of larger blocks and therefore resulting in poor blocking performance. So, a core goal is to ensure that communities of names are larger than one or two and obviously at the other end of the

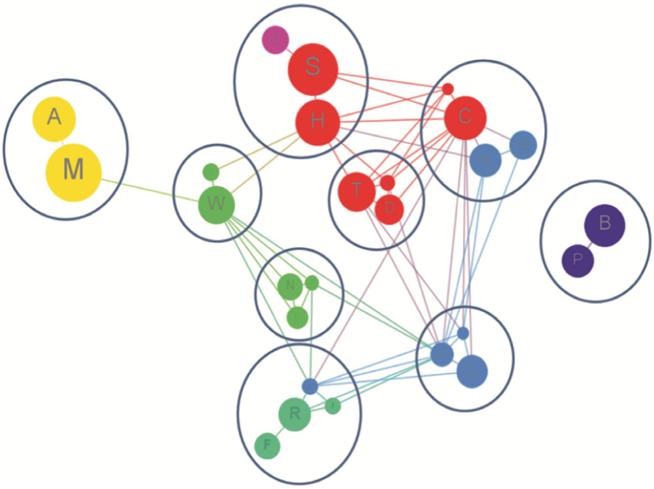


**Fig. 1.** This figure depicts a derived contracted graph indicating the relationship between classes of names based on their first letter. This graph is used in the second stage of the intermediate blocking phase to improve the accuracy and completeness of the reference graphs.

spectrum are small enough so the communities/blocks enable scalable deployment. To do this the two highest (as the graph is now a similarity based graph rather than a distance based graph) weighted incident edges of every node (i.e. name) is retained to ensure the graph retains a giant single component. It is important to select the two highest as simply using the single highest weighted edge can result in isolated dyads. Using this approach significantly reduces the number of components and ensures the smallest component is a triad. Furthermore, all edges above a pre specified threshold were retained.

An alternate source of names and their variants (derived from transcription, hypocorisms) is then introduced. This secondary source of proper name variation is important from a human centered systems perspective as it creates the opportunity for experts to add their explicit knowledge into the entity resolution model and derive instant performance improvements. Knowledge is added to both create relationships between names and negate relationships between names that do not exist. For example, the transcription of the Chinese family name 韩 into the Latin alphabet is dependent on the dialect - China's pinyin system converts this to "Han", in Cantonese "Hon", and in Hainan "Hang". Furthermore, negation is critical in situations where names are very similar but are actually distinct proper names. String matching cannot discriminate between these sets of names easily, so other methods are required to buttress performance (see Fig. 2.). The use of deterministic sources also reinforces the fact that the reference graphs are indeed assets that can be developed and curated to support a range of endeavors in addition to identity and entity resolution such as named entity recognition.

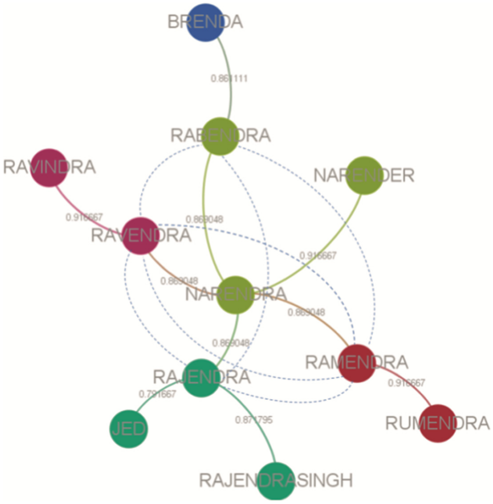


**Fig. 2.** This figure provides an example of Given Name Reference Graph (GNRG) annotation. The dashed blue lines represent the relationships that have been manually removed to ensure the four proper names "Rajendra", "Ravendra", "Rabendra", and "Ramendra" are discriminated between appropriately. Note how the community detection appropriately ascribes a different membership to each of these four names (Color figure online).

Next a non-overlapping community detection algorithm was deployed to partition the graph of names into classes (see Fig. 3. for an illustration of a subgraph of the FNRG). In this case the multilevel algorithm was used [19] due to its relative speed and performance.

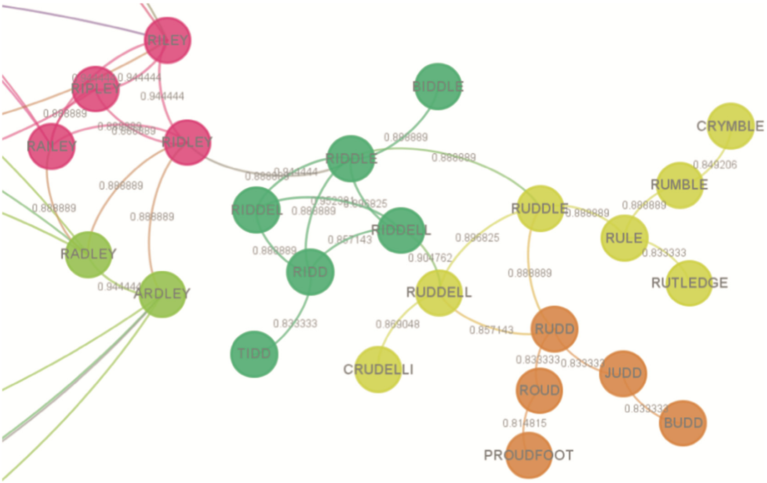


**Fig. 3.** This figure illustrates a subgraph of the Family Name Reference Graph (FNRG), including the membership classes derived from the community detection algorithm which are used in blocking.

Subsequent to the use of a community detection algorithm the classification derived from a proper name co-occurrence graph (for example, the name “John Edward Smith” would create a triad of “John– Edward; Edward – Smith; and John – Smith”) derived from the target unresolved data, which is used to deterministically predict the origin of proper names using a coarse classification, is then used to allocate new hybrid classes.

Importantly the frequency of each name within the target unresolved dataset is retained as a node attribute to enable efficient assessment of the block sizes. The three levels of blocks in addition to the frequency of names from each block represented in the target unresolved dataset enables a degree of optimization, dependent on the domain specific context (e.g. the level of incompleteness and uncertainty in the data is significant), the business context (e.g. the cost of missing a match is high and real time assessment is not required so batch processing is preferable) and other factors such as hardware, software etc.

At this point the FNRG can be represented as a simple table of nodes with a membership integer, and the GNRG the more complex representation of a ragged array of integers, due to the nature of people having multiple given names.

## 2.2 Reference Graph Implementation

The entity resolution function with which the reference graphs were deployed within has four modules (see Fig. 4.).

The first module (see Fig. 4.) is Pre-Indexing which selects (e.g. Persons) and constrains (e.g. only those persons that have a Family name, Given name, and Date of Birth) the nodes to be compared for entity resolution. A range of parameters are used to configure this module.

The Equivalence Assessment module (see Fig. 4.) takes the set of entities from the previous module and creates sub-sets or blocks (Indexing), based on the algorithm selected as a parameter, to ensure the quadratic assessment of pairs is done in a scalable manner yet retaining as much accuracy as possible. Then Approximate String Matching (ASM), based on the algorithm (Jaro-Winkler or Cosine) selected as a parameter, is performed on each pair.

The Decision Management module (see Fig. 4.) takes the output of the previous module and a range of attributes from the target unresolved dataset (g) and makes a decision on whether the pair are a match or not. This decision is made on the basis of two concepts. Congruence – how similar the pair are in terms of the metrics available, and commonality – how unique the set of attributes are. These two concepts create the basis to not only decide whether the pair of entities are in fact the same real world entity in a probabilistic way but additionally how much certainty exists so optimized decision making can be made, given the domain context, of whether decisions can be made with a reduced set of attributes in a relatively inexpensive fashion, or whether uncertainty is high enough, given the domain context, to require an enriched set of attributes and a higher standard of proof. Decision making is conducted via a rule-based approach.

The Graph Contraction module (see Fig. 4.) manages the contraction of the graph using a range of methods including provenance to select what data to retain as the primary attributes. The meta-data derived from the ER model is retained.

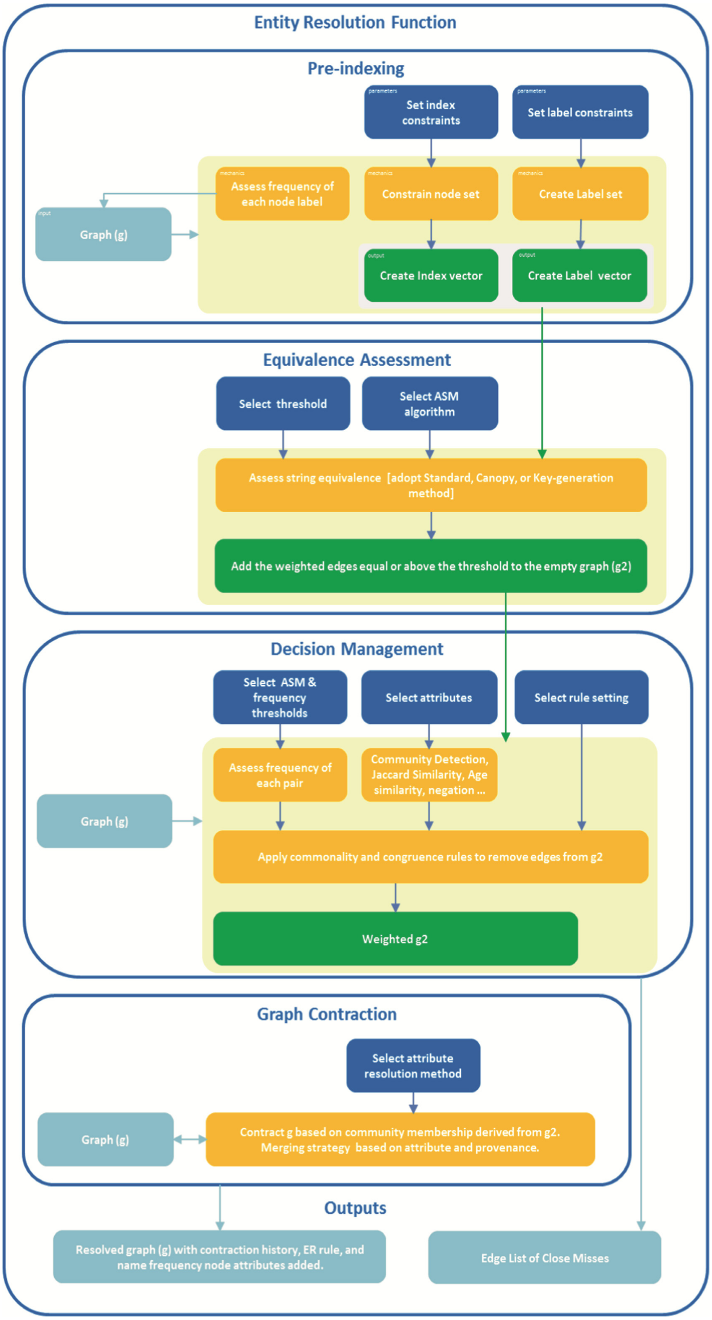

Entity Resolution Function
Pre-indexing
parameters
Set index constraints
parameters
Set label constraints
mechanics
Assess frequency of each node label
mechanics
Constrain node set
mechanics
Create Label set
input
Graph (g)
output
Create Index vector
output
Create Label vector
Equivalence Assessment
Select threshold
Select ASM algorithm
Assess string equivalence [adopt Standard, Canopy, or Key-generation method]
Add the weighted edges equal or above the threshold to the empty graph (g2)
Decision Management
Select ASM & frequency thresholds
Select attributes
Select rule setting
Assess frequency of each pair
Community Detection, Jaccard Similarity, Age similarity, negation ...
Graph (g)
Apply commonality and congruence rules to remove edges from g2
Weighted g2
Graph Contraction
Select attribute resolution method
Graph (g)
Contract g based on community membership derived from g2. Merging strategy based on attribute and provenance.
Outputs
Resolved graph (g) with contraction history, ER rule, and name frequency node attributes added.
Edge List of Close Misses

**Fig. 4.** This figure is a graphical representation of the design of the entity resolution function which was used to implement the experimental design. The indexing and decision management of the various strategies were conducted within the Equivalence Assessment and Decision Management modules respectively.

The reference graphs are used at both Equivalence Assessment (blocking) and Decision Management modules. Within the Equivalence Assessment module the FNRG community membership is used to create blocks or subsets of similar nodes to ensure the implementation is scalable. The granularity of blocks is dependent on the size and nature of the entity resolution task. The GNRG is not implemented within the Equivalence Assessment module at this point in time due to the added complexities of the overlapping nature of community membership.

However, both FNRG and GNRG are used in the Decision Management module. The FNRG is represented as a simple binary marker of whether the Family names of the pair of persons share the same class or not.

The GNRG is similarly implemented however rather than a binary approach each of the pairs community membership vectors (which can be of varying length dependent on how many given names they have) are compared and the number of matches is output.

## 3 Experiment

The utility of the reference graphs is measured on their ability to augment indexing and decision management. Performance measurement is based on the computational speed, scalability, and the number of true positive pairs identified through the application of one run of an optimally set entity resolution function. Four blocking strategies are compared. The reference graph strategy and three alternate blocking strategies have been deployed within the Equivalence Assessment module to serve as performance benchmarks. Each of the four blocking strategies is compared on the performance metrics under two decision management states – not using reference graph attributes and using reference graph attributes to support decision making. All other parameters are kept constant at near optimal levels for each dataset to replicate real-world conditions.

Importantly, the relative contextual performance of the overall ER model, which is comprised of multiple ER functions in a range of configurations, is briefly compared against a range of proprietary and non-proprietary entity resolution models that were benchmarked by Ferrante and Boyd [20]. This was done to provide comparative insight into how the reference graph algorithm contributes to the overall real world performance of the ER model.

### 3.1 Indexing Strategies

The “reference graph” strategy is implemented using the Family Name Reference Graph, and the classes derived from this graph.

The “Meta-Blocking Canopy” strategy [7–9] amalgamates blocks in a logical way to enhance accuracy with concomitant performance degradation. It is an effective optimizing strategy when the cost of inaccuracy is high, and computational expense can be sacrificed. The implementation here uses string length as the underpinning blocking strategy and then takes clusters of these blocks to form meta-blocks. The purpose for using the canopy strategy as a benchmark was to firstly to create a quasi-gold standard, and secondly to introduce the concept from a block-optimizing perspective to provide the explicit extension potential of using such strategies in tandem with reference graphs.

The second benchmark blocking strategy is a simple "Truncation" strategy. The implementation here is in two forms. For the small and two big datasets key-generation is derived from the truncation of each person entities family name at the first two letters (e.g. "DerekRobinson" > "RO") and first three letters (e.g. "DerekRobinson" > "ROB"), respectively. The reason for changing the implementation between small and large datasets is due to scalability and real world application.

The third strategy is the "Phonetic" strategy implemented using the Metaphone 3 (v2.5.4) algorithm [5] generating keys from each Person entities Family Name.

### 3.2 Decision Management States

The two decision management states implemented here consist of holding all other parameters constant and applying the use of the Given Name Reference Graph (GNRG) community membership attribute, or not.

### 3.3 Context

As discussed three real world target unresolved datasets are used to test the conditions previously outlined.

The first dataset is centered on Suspicious Transactions and is a small heterogeneous directed multiplex graph of approximately 40,000 nodes and 51,000 edges, comprised largely of manually annotated data, involving a vast range of relationships including familial, business, and financial transactions. The data contains high data incompleteness meaning there will be a high number of missing edges and nodes, and attributes thereof. Furthermore, fake and spoof nodes will also be present. Fake nodes are nodes that exist in the dataset that do not exist in the real world, and spoof nodes is where a real world entity is represented as multiple nodes in the data [2]. Both fake and spoof nodes will be derived from instances where real world entities intend to provide misinformation and where instances simply derive from human error. The cost of false positives and false negatives is high and therefore accuracy is paramount.

The second and third datasets are drawn from shareholding and directorship of NZ Companies. The Partial Companies graph is derived from the giant component from this data, and the Complete Companies graph is the complete graph. They are homogeneous multiplex weighted graphs of approximately 1.1 million nodes and 2.4 million edges, and 2.2 million nodes and 3.8 million edges respectively. The relationships consist of two types – shareholding and directorship. The data is drawn from the companies register and therefore is relatively complete; however there is a relatively low threshold of entity validation. Indeed it is highly probable that again fake and spoof nodes are present. The cost of false positives and false negatives is not as high, but scalability and run time become increasingly important considerations.

Here we are testing one entity resolution function to generate performance metrics from which to assess the utility of reference graphs. Of course within real world application the entity resolution function would be run, with alternate parameters, multiple times and as a collective form the implemented entity resolution model. In this experiment the performance metrics is couched within the context of the entity resolution model's performance and the context of each of the three target unresolved datasets.

It is also important to explicitly state that the entity resolution model was designed for batch process rather than real time, and the strength of the model is its extensibility and ability to be configured to incorporate a range of different methods at the indexing phase within the Equivalence Assessment module (e.g. Canopy, Phonetic, FNRG, Community Detection) and within the Decision Management module (e.g. hypocorisms, graph distance, reference graphs) for the purpose of entity resolving non-obvious latent entity pairs in the criminal context. Reference graphs are designed to ally the entity resolution of non-obvious pairs of entities and as such it is important to create the conditions where this is the goal. So, deduplication is conducted first to ensure all obvious entity resolutions are completed so as to leave only the non-obvious pairs.

In terms of the contextual performance of the ER model, of which the use of reference graphs is but one feature, both runtime and quality metrics (precision, recall, and f-measure) have been benchmarked against other ER models using Ferrante and Boyd's [20] comparison of software using synthetic data. Using the comparative performance results from Ferrante and Boyd's [20] study the ER model used here performs in the "slow" and "moderate" brackets for the small and large record sets respectively. However the quality metrics derived from the ER model used here (precision: 0.999, recall: 0.994, f-measure: 0.996) on the Suspicious Transactions Graph significantly outperform all benchmarked competitors (the top ranked software attained precision: 1.0, recall: 0.79, f-measure: 0.88), and using Ferrante and Boyd's classification would be classed as "very good". The limitations of this comparator are obvious but give a useful guide to performance for contextual purposes.

### 3.4 Performance Metrics

The following performance metrics have been used to assess the utility of the reference graphs within the experiment; computational expense; scalability and accuracy.

Computational expense consists of the measurement of each of the four blocking strategies across Equivalence Assessment (indexing) and Decision Management modules in both states (reference graph used in decision management or not) within the context of the entity resolution function and the overall entity resolution model.

Scalability is measured within the Equivalence Assessment module as the optimization of blocking remains a key research and applied problem. The primary metric is the number of computations required; however the number of blocks and the maximum number of entities within the blocks are important metrics to give a sense of block distribution, and how that distribution translates into computational performance.

Accuracy is measured simply by the number of pairs correctly resolved (true positives). The introduction of error through incorrectly resolving two entities within the context of the three data-sets generates a high cost and therefore the simple metric of counting the number of true positives is core. By-products of using a simple metric are that it was relatively simple to ensure each strategy was compared like for like, and the possibility of bias was reduced, and also the assessment of which strategy was most performant is made straight forward.

Diversity of the blocking approaches as a collective is another important concept to measure as in the real world the strategies are simply implemented as an alternate parameter setting from which the user can select and tune a specific entity resolution function. Not just one entity resolution function, but a number of entity resolution functions that together as a collective comprise the entity resolution model. Having said this however the partitions derived from the experiment indicate a nested structure, so those with a greater number of correctly resolved pairs are more diverse.

Table 1 outlines the scalability and accuracy of the reference graph and competing algorithms. The measurement of scalability was achieved by measuring the number of blocks generated, the maximum block size, the highest number ASM of computations conducted on a block, and total ASM computations. The measurement of accuracy was conducted by measuring the number of matches when the Given Name Reference Graph was not used in the Decision Management module, and the number of matches when the Given Name Reference Graph was used in the Decision Management module.

Table 2 illustrates the computational expense of the reference graph and competing algorithms across the three datasets for pre-processing, Equivalence Assessment (indexing) and Decision Management modules, the total run time for the Equivalence Assessment and Decision Management modules, and the average run time for each ER function.

## 3.5 Experimental Results

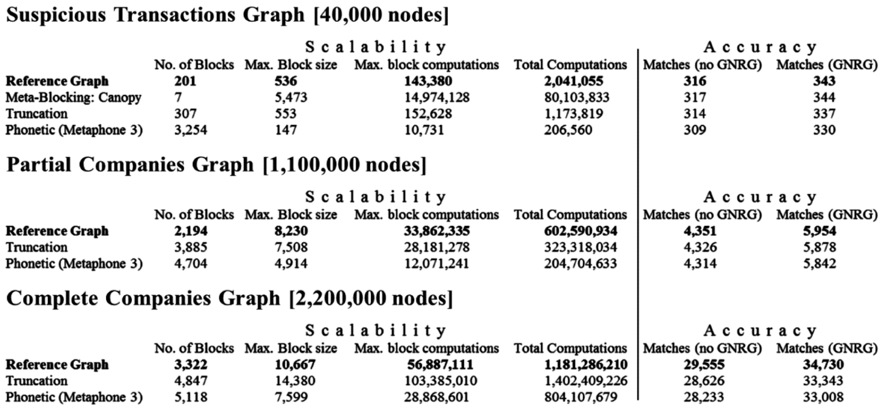
From Tables 1 and 2 the performance profiles of each strategy is evident. The Meta-Blocking Canopy strategy, as implemented, is not scalable as highlighted within scalability metrics and particularly the large number of total computations (80,103,833) but due to the near exhaustive equivalence assessment is most accurate on the Suspicious Transactions Graph with 317 (not using the GNRG) and 344 (using the GNRG) matches. In terms of its speed the Meta-Blocking Canopy strategy is slower, however the high accuracy of this approach means it remains a viable niche strategy on small graphs.

The Truncation strategy is scalable but performance will drop as the size of the dataset increases into the millions of nodes as either the number of computations increases sharply relative to the other algorithms or the truncation strategy is adjusted and accuracy suffers. Performance is good on the Suspicious Transactions Graph both in terms of expense and accuracy. This strategy is simple to implement, is fast on small graphs and relatively scalable.

The Phonetic (Metaphone 3) strategy, surprisingly, is consistently the poorest performer from an accuracy perspective, but the most scalable. The accuracy may be due to the very diverse set of names contained within the Suspicious Transactions graph, however the Companies graphs are less diverse and still the algorithm underperforms. Run times are slow on small graphs but relatively quicker on the larger graphs.

The reference graph strategy has a relatively expensive pre-processing time, however this one-off cost is offset under the context of the entity resolution model that will run multiple entity resolution functions together as a cluster to perform the resolution. Otherwise, the reference graph strategy consistently out-performed the other algorithms on both run time and accuracy. In terms of scalability the reference graph showed it is scalable, and due to being represented in a graph optimization is possible.

**Table 1.** Experimental results: Scalability and accuracy.



**Suspicious Transactions Graph [40,000 nodes]**

| | Scalability | | | | Accuracy | |
|---|---|---|---|---|---|---|
| | No. of Blocks | Max. Block size | Max. block computations | Total Computations | Matches (no GNRG) | Matches (GNRG) |
| **Reference Graph** | **201** | **536** | **143,380** | **2,041,055** | **316** | **343** |
| Meta-Blocking: Canopy | 7 | 5,473 | 14,974,128 | 80,103,833 | 317 | 344 |
| Truncation | 307 | 553 | 152,628 | 1,173,819 | 314 | 337 |
| Phonetic (Metaphone 3) | 3,254 | 147 | 10,731 | 206,560 | 309 | 330 |

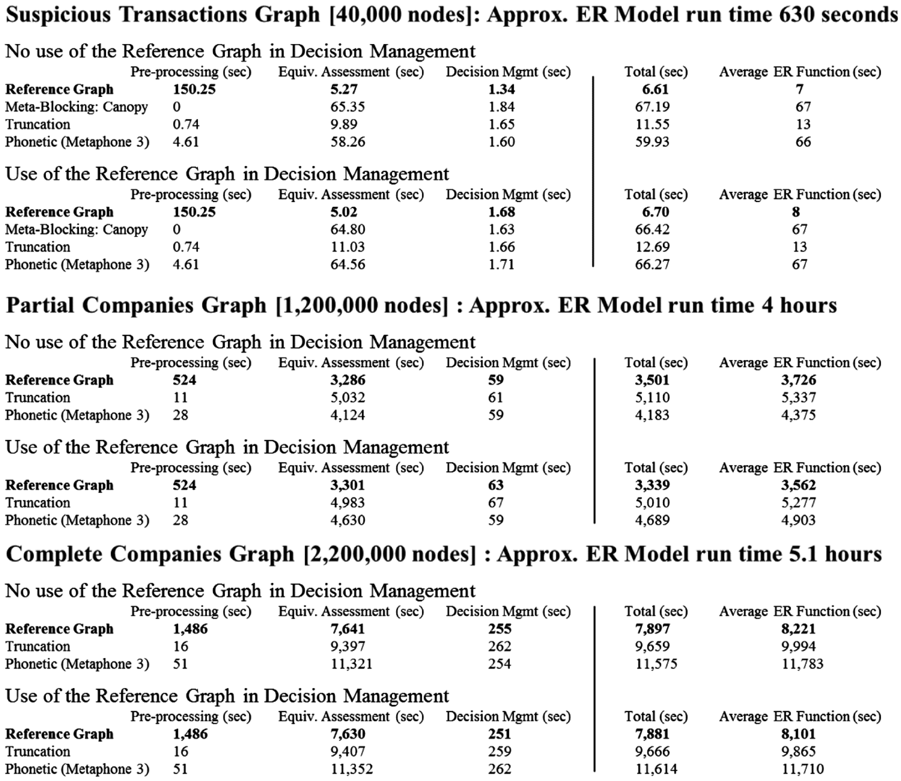
**Partial Companies Graph [1,100,000 nodes]**

| | Scalability | | | | Accuracy | |
|---|---|---|---|---|---|---|
| | No. of Blocks | Max. Block size | Max. block computations | Total Computations | Matches (no GNRG) | Matches (GNRG) |
| **Reference Graph** | **2,194** | **8,230** | **33,862,335** | **602,590,934** | **4,351** | **5,954** |
| Truncation | 3,885 | 7,508 | 28,181,278 | 323,318,034 | 4,326 | 5,878 |
| Phonetic (Metaphone 3) | 4,704 | 4,914 | 12,071,241 | 204,704,633 | 4,314 | 5,842 |

**Complete Companies Graph [2,200,000 nodes]**

| | Scalability | | | | Accuracy | |
|---|---|---|---|---|---|---|
| | No. of Blocks | Max. Block size | Max. block computations | Total Computations | Matches (no GNRG) | Matches (GNRG) |
| **Reference Graph** | **3,322** | **10,667** | **56,887,111** | **1,181,286,210** | **29,555** | **34,730** |
| Truncation | 4,847 | 14,380 | 103,385,010 | 1,402,409,226 | 28,626 | 33,343 |
| Phonetic (Metaphone 3) | 5,118 | 7,599 | 28,868,601 | 804,107,679 | 28,233 | 33,008 |

**Table 2.** Experimental results: Computational expense (run time in seconds).



**Suspicious Transactions Graph [40,000 nodes]: Approx. ER Model run time 630 seconds**

No use of the Reference Graph in Decision Management

| | Pre-processing (sec) | Equiv. Assessment (sec) | Decision Mgmt (sec) | Total (sec) | Average ER Function (sec) |
|---|---|---|---|---|---|
| **Reference Graph** | **150.25** | **5.27** | **1.34** | **6.61** | **7** |
| Meta-Blocking: Canopy | 0 | 65.35 | 1.84 | 67.19 | 67 |
| Truncation | 0.74 | 9.89 | 1.65 | 11.55 | 13 |
| Phonetic (Metaphone 3) | 4.61 | 58.26 | 1.60 | 59.93 | 66 |

Use of the Reference Graph in Decision Management

| | Pre-processing (sec) | Equiv. Assessment (sec) | Decision Mgmt (sec) | Total (sec) | Average ER Function (sec) |
|---|---|---|---|---|---|
| **Reference Graph** | **150.25** | **5.02** | **1.68** | **6.70** | **8** |
| Meta-Blocking: Canopy | 0 | 64.80 | 1.63 | 66.42 | 67 |
| Truncation | 0.74 | 11.03 | 1.66 | 12.69 | 13 |
| Phonetic (Metaphone 3) | 4.61 | 64.56 | 1.71 | 66.27 | 67 |

**Partial Companies Graph [1,200,000 nodes] : Approx. ER Model run time 4 hours**

No use of the Reference Graph in Decision Management

| | Pre-processing (sec) | Equiv. Assessment (sec) | Decision Mgmt (sec) | Total (sec) | Average ER Function (sec) |
|---|---|---|---|---|---|
| **Reference Graph** | **524** | **3,286** | **59** | **3,501** | **3,726** |
| Truncation | 11 | 5,032 | 61 | 5,110 | 5,337 |
| Phonetic (Metaphone 3) | 28 | 4,124 | 59 | 4,183 | 4,375 |

Use of the Reference Graph in Decision Management

| | Pre-processing (sec) | Equiv. Assessment (sec) | Decision Mgmt (sec) | Total (sec) | Average ER Function (sec) |
|---|---|---|---|---|---|
| **Reference Graph** | **524** | **3,301** | **63** | **3,339** | **3,562** |
| Truncation | 11 | 4,983 | 67 | 5,010 | 5,277 |
| Phonetic (Metaphone 3) | 28 | 4,630 | 59 | 4,689 | 4,903 |

**Complete Companies Graph [2,200,000 nodes] : Approx. ER Model run time 5.1 hours**

No use of the Reference Graph in Decision Management

| | Pre-processing (sec) | Equiv. Assessment (sec) | Decision Mgmt (sec) | Total (sec) | Average ER Function (sec) |
|---|---|---|---|---|---|
| **Reference Graph** | **1,486** | **7,641** | **255** | **7,897** | **8,221** |
| Truncation | 16 | 9,397 | 262 | 9,659 | 9,994 |
| Phonetic (Metaphone 3) | 51 | 11,321 | 254 | 11,575 | 11,783 |

Use of the Reference Graph in Decision Management

| | Pre-processing (sec) | Equiv. Assessment (sec) | Decision Mgmt (sec) | Total (sec) | Average ER Function (sec) |
|---|---|---|---|---|---|
| **Reference Graph** | **1,486** | **7,630** | **251** | **7,881** | **8,101** |
| Truncation | 16 | 9,407 | 259 | 9,666 | 9,865 |
| Phonetic (Metaphone 3) | 51 | 11,352 | 262 | 11,614 | 11,710 |

Perhaps the most significant finding was the clear advantage of using the Given Name Reference Graph to assist making decisions, especially considering there is no material computational expense involved.

## 4 Discussion

The experimental results clearly mark the applied utility of the reference graph strategy, and excitingly, the demonstrated applied utility is buttressed by a number of features that extends the real-world value of this strategy.

The experimental results clearly indicate the relative scalability, expense, and accuracy of the reference graph as a blocking strategy, across all graph types examined showing encouraging performance and generalized applicability. Furthermore, the reference graph has a number of features that extends this value under real-world conditions. From a human centered computing perspective the reference graph can be improved over time by the curation of human experts annotating the relationships between proper names, crucially including ensuring counter-factual relationships between proper names do not exist (e.g."Rabendra" ! = "Ravendra"), and validate the performance of community detection.

The flexibility of the coarseness of partitioning, or indeed potential for over-lapping classes, is another feature that enable meta-blocking like capabilities creating the opportunity to tune the reference graph dependent on the contextual requirements demanded by each individual set of instances.

As an adjunct to decision management the case for the use of reference graphs is compelling. Performance was significantly enhanced with little to no material increase in expense. From a criminal network perspective, as alluded to earlier, performance enhancements targeting the non-obvious pairs is the focus and a very complex and challenging problem. The results derived from the experiment are very encouraging from this perspective, both in terms of dealing with the higher uncertainty of the Suspicious Transactions Graph, and from a scalability perspective with the Companies Graphs. Of course these are only indicative findings and further comparison against a variety of "state of the art" algorithms using a diverse range of criminal datasets is required to further validate the utility of reference graphs.

From a real world perspective the application of the ER model using reference graphs within the criminal domain has significantly improved downstream models, designed to detect, measure and prioritize risk, that consume the output from the ER model. It was expected that the downstream benefit would be more amplified in anomaly detection approaches, however there has proven to be a significantly improved performance across the more generic models as well. Models such as shortest-path based models that identify subgraphs of clusters of entities involved in suspicious transactions that are linked to entities that generate illicit income (e.g. methamphetamine traffickers), and graph propagation models used across a range of criminal sub-domains. This provides initial tentative support to the earlier assertion that the latent non-obvious pairs are of most importance, and indeed this small set of latent actors may well be essential to the ongoing criminal structural fabric of criminal networks.

## 5 Extensions

A number of areas have been identified to extend the current implementation of reference graphs.

The use of overlapping community detection approaches will undoubtedly increase the accuracy of the reference graphs, however there is certainly a trade-off of making the approach more complex and potentially significantly more expensive even if data representations like hypergraphs are used.

The reference graphs are used as a binary attribute to drive blocking and support decision management. However, there is the simple extension of deploying the attribute as a graph-distance based metric that creates the opportunity for optimized blocking via a meta-blocking implementation and at the decision management phase creates the opportunity to use a more nuanced and sophisticated approach to support making decisions.

## 6 Conclusion

The use of reference graphs to bolster performance in entity resolution, at both indexing and decision management stages, has been clearly demonstrated within this paper, with both experimental results and the outlining of additional real-world benefits. This coupled to the reference graphs wide applicability, simple implementation, and numerous areas for extensions points to an entity resolution strategy that has great potential for generating real-world value.

Specifically within the criminal domain the use of reference graphs within entity resolution has been demonstrated to be both performant from an accuracy perspective, which is critical when targeting non-obvious instances, and also performant from a scalability perspective. This unlocks the ability to federate data between a criminal network hub and multiple large heterogeneous datasets (the spokes), in addition to providing quality accurate resolution with data characterised by incompleteness, high uncertainty, and the presence of fake and spoof nodes.

## References

1. Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J.: Swoosh: a generic approach to entity resolution. VLDB J. **18**(1), 255–276 (2009)
2. Maeno, Y.: Node discovery problem for a social network. Connections **29**, 62–76 (2009)
3. Odell, M., Russell, R.: The Soundex Coding System. US Patents 1261167 (1918)
4. Philips, L.: The double metaphone search algorithm. C/C ++ Users J. **18**(6), 38–43 (2000)
5. Philips, L.: Metaphone 3 version 2.5.4 (2015)
6. de Vries, T., Ke, H., Chawla, S., Christen, P.: Robust record linkage blocking using suffix arrays. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 305–314. ACM (2009)
7. Hernández, M.A., Stolfo, S.J.: The merge/purge problem for large databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data 1995, pp. 127–138. ACM, New York (1995)

8. Hernández, M.A., Stolfo, S.J.: Real-world data is dirty: data cleansing and the merge/purge problem. Data Min. Knowl. Discov. **2**(1), 9–37 (1998)
9. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 169–178. ACM (2000)
10. Taylor, J.: Decision Management Systems: A Practical Guide to Using Business Rules and Predictive Analytics. Pearson Education, Boston (2012)
11. Bhattacharya, I., Getoor, L.: Collective entity resolution in relational data. ACM Trans. Knowl. Discov. Data **1**(1), 1–36 (2007)
12. Bhattacharya, I., Getoor, L.: Entity Resolution in Graphs. In: Cook, D.J., Holder, L.B. (eds.) Mining Graph Data, pp. 311–344. Wiley, Hoboken (2006)
13. Köpcke, H., Rahm, E.: Frameworks for entity matching: a comparison. Data Knowl. Eng. **69**(2), 197–210 (2010)
14. Randall, S.M., Boyd, J.H., Ferrante, A., Bauer, J.K., Semmens, J.B.: Use of graph theory measures to identify errors in record linkage. Comput. Methods Programs Biomed. **115**(2), 55–63 (2014)
15. Zhou, Y., Talburt, J.R.: Strategies for large-scale entity resolution based on inverted index data partitioning. In: Yeoh, W., Talburt, J.R., Zhou, Y. (eds.) Information Quality and Governance for Business Intelligence, pp. 329–351. IGI Global, Hershey (2013)
16. Michalowski, M., Thakkar, S., Knoblock, C.A.: Exploiting secondary sources for unsupervised record linkage. In: Proceedings of the 30th VLDB Conference, Toronto, Canada (2004)
17. Papadakis, G., Koutrika, G., Palpanas, T., Nejdl, W.: Meta-blocking: taking entity resolution to the next level. IEEE Trans. Knowl. Data Eng. **26**(8), 1946–1960 (2014)
18. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: Proceedings of the Section on Survey Research Methods, American Statistical Association, pp. 354–359 (1990)
19. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech. Theory Exper. **2008**(10), P10008 (2008)
20. Ferrante, A., Boyd, J.: A transparent and transportable methodology for evaluating data linkage software. J. Biomed. Inform. **45**(1), 165–172 (2012)