Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

IMPROVING RECORD LINKAGE FOR COUNTER-THREAT FINANCE INTELLIGENCE WITH DYNAMIC JARO-WINKLER THRESHOLDS

Ian Kloo

Department of Systems Engineering
United States Military Academy
West Point, NY 10996, USA

Matthew F. Dabkowski, PhD

Department of Systems Engineering
United States Military Academy
West Point, NY 10996, USA

Samuel H. Huddleston, PhD

Biocore, LLC
Charlottesville, VA 22911, USA

ABSTRACT

Counter-Threat Finance Intelligence (CTFI) is a discipline within the U.S. intelligence enterprise that illuminates and prosecutes terrorist financiers and their supporting networks. Relying on voluminous, disparate financial data, efficient and accurate record linkage is critical to CTFI, as successful prosecutions and asset seizures hinge on the association of designated, nefarious entities with financial transactions falling under U.S. jurisdiction. The Jaro-Winkler (J-W) algorithm is a well-known, widely used string comparator that is often employed in these record linkage problems. Despite J-W's popularity, there is no academic consensus on the threshold score at which strings should be considered likely matches. In practice, J-W thresholds are selected somewhat arbitrarily or with little justification. This paper uses a simulative approach to identify optimal J-W thresholds based on an entity pair's string lengths, thereby improving the lead-discovery process for CTFI practitioners.

1 INTRODUCTION

Counter-Threat Finance Intelligence (CTFI) is defined as "the means and methods used by the US government and legitimate actors in the financial industry to discover, disrupt, and deter the financing of threats to US national security and global stability" (American Security Project). As a 2010 RAND study notes, "There is a wide-ranging acceptance across the U.S. Government that using intelligence to follow the finances of terrorists, drug traffickers, and weapons proliferators is a useful way to track these groups and may expose new ways to degrade their capabilities" (Bahney et al. 2010, p. 57). One significant tool for degrading threat networks is to invoke various authorities that allow law enforcement entities to prosecute individuals and seize or freeze assets within the global financial system when they can prove that a transactions occurred within US jurisdiction and involved designated entities or organizations. The fundamental analytical task in developing a successful case for prosecution or asset seizure of an individual, business, or group based on financial records is record linkage. The Jaro-Winkler (J-W) algorithm is a widely-used heuristic approach for string matching that plays a fundamental role in many record linkage applications.

The simulation study documented in this paper is motivated by a previous research project conducted by an interagency task force that developed a novel analytical method (and software) to support CTFI (Huddleston et al. 2016). The method developed in that project processes bulk data stores of financial transactions that are flagged as suspicious and, through a largely automated process, generates representations of financial activity that highlight transactions and actors that are good candidates for close investigation by

analysts (i.e., likely to be good candidates for prosecution and seizure investigations), providing a service that would otherwise require decades or centuries of analyst time. It has been well-received by the military's analytic community (Goerger et al. 2016), and it is effective (Cunningham 2017).

The core analytical tasks performed by Huddleston et al.'s (2016) software are the automated record linkage of financial transactions that involve the same entities, followed by their identity resolution against many different databases of "designated individuals and organizations," where attempts have often been made to hide the true identities of the entities involved in threat transactions. In practice, this method's performance hinges on the selection of predetermined thresholds (θ), which are applied to the results of J-W string matching across many different transaction attribute pairs. When θ is set too low, the representations of the financial networks generated are not accurate and analysts are overwhelmed with investigative leads to pursue, many of which are not valid (i.e., the algorithmic precision is too low). When θ is set too high, opportunities to successfully seize and prosecute important assets of criminal and terror organizations are missed (i.e., the algorithmic recall is too low). Huddleston et al. (2016) addressed this tradeoff by setting θ to maintain exceptional precision and then leveraging limited clerical reviews to improve recall.

This paper documents a simulation study designed to address this fundamental precision-recall trade-off by estimating ideal values for θ based on an entity pair's string lengths. It is organized as follows. Section 2 provides additional background on Huddleston et al.'s (2016) approach and entity deduplication in general, paying special attention to the danger of using string-length invariant thresholds. Section 3 describes the proposed methodology, followed by the results of preliminary testing in Section 4. Section 5 discusses insights related to the proposed methodology, especially in the context of the original CTFI problem space. Section 6 highlights limitations of the suggested approach and opportunities for future work. Finally, Section 7 presents our conclusions.

2 BACKGROUND

2.1 Threat Finance Intelligence

The U.S. Government has enacted several authorities which serve as powerful tools in its fight to disrupt and dismantle threat finance networks. These authorities allow the government to prosecute individuals for terrorist financing and facilitation (18 U.S.C §1956; 18 U.S.C. §§2339A-D; International Emergency Economic Powers Act, 2011), seize the assets of terrorists (Money Laundering Control Act, 2011), or freeze the assets of any person associated with individuals or organizations identified as Specially Designated Global Terrorists (SDGT) (Executive Order No. 13224, 2001). As of December 31, 2017, there were 1,030 individuals, businesses, and organizations on the SDGT list (U.S. Department of the Treasury, 2017, p. 6). Even when financial transactions occur outside U.S. jurisdiction, authorities exist. For example, *U.S. vs. Bank of Nova Scotia* (1982) gives the United States the ability to obtain foreign bank information if the bank has a U.S. branch, and the Patriot Act (2001) enables the U.S. Government to issue administrative subpoenas to foreign banks with correspondent assets within the United States.

Investigations that successfully employ the above authorities invariably rely on efforts to follow the money and connect financial transactions to individuals or groups designated in the SDGT or the U.S. State Department's list of Foreign Terrorist Organizations. In order to charge a suspect with a substantive terrorist financing charge, the investigation team needs to prove that an intermediary in a transaction is a member, agent, or established conduit of a terrorist organization (Taxay et al. 2014, p. 13). Therefore, the key to a successful investigation and prosecution leveraging these authorities is the ability to link financial transactions that occur within U.S. jurisdiction (any transaction that uses the U.S. dollar or that flows through a U.S. banking institution) to designated entities or organizations.

2.2 Record Linkage and Identify Resolution

Entity Resolution (ER) is "the process of determining whether two references to real-world objects are referring to the same, or to different, objects" (Talburt 2011). When entity resolution is performed by

matching across different types of records or databases, it is often referred to as record linkage. The need to link records across databases and information systems emerged in the 1950s with the advent and proliferation of the modern computer (Newcombe et al. 1959). Rooted in the fields of vital statistics and census-related studies (Marshall 1947), its principal objective remains largely unchanged: to assemble accurate, expanded depictions of individuals from relevant yet disparate data sets. Record linkage is accomplished by matching entity references (records) based on record attributes (e.g., name, address, phone number, etc.) (Felligi and Sunter 1969).

Identity resolution (IR) extends entity resolution (record linkage) to include resolution against known and unique identities. In CTFI, the record linkage problem involves determining which of the entities described in the various financial transaction records are the same entities described differently. The identity resolution problem involves determining whether any of those entities are likely to be persons/businesses/organizations on the SGDT list, other sanction lists, or entities of interest that are documented in various intelligence databases. Due to the nature of the threat finance operations, it is critical to be able to conduct ER and IR in concert, and to be able to link records in a largely automated fashion, because criminal and terror networks often engage in elaborate money-laundering schemes that involve the movement of money through elaborate networks of both criminal and legitimate businesses. One must also follow the transactions through the global financial system by linking records across many different banking institutions and countries, all with different record-keeping conventions, to meet both standards required for a case: jurisdiction and documentation of the illicit money flow to a designated entity.

Illuminating threat financial networks involves leveraging all of the various type of ER matching methods: alias matching for common transliterations (e.g., Muhammed and Mohammed), name matching across many known aliases (i.e., against all possible combinations of address, phone number, and name employed by known criminals/terrorists), deterministic matching for attributes such as birth dates (with aliasing considered in cases such as 4/27/87 vs. April 27, 1987), and fuzzy string matching of free text fields such as names and addresses. By far the most difficult of the matching requirements is the need to match (often transliterated) names and addresses entered as a free-text field. This is especially common in records of remittance payments. Because of its exceptional balance of speed (for processing bulk stores of financial transactions) and performance, Huddleston et al. (2016) employed the well-known J-W algorithm for fuzzy string matching, establishing minimum thresholds for similarity in compared text fields for the assertion of a likely match and the resulting generation of a "link" in a threat network flagged for analyst review. Given its central role in the present study, a detailed description of the J-W algorithm is presented in Section 2.3.

2.3 J-W Algorithm

The J-W Algorithm traces its roots back to Jaro's UNIMATCH, a record-linkage system developed under the purview of the U.S. Census Bureau in the 1970s (Jaro 1978). Unlike its narrowly-focused predecessors, UNIMATCH was conceived as a general tool for matching records across a wide variety of fields, including "names having an uncertainty in spelling" (Jaro 1978, p. 8). To this end, UNIMATCH employed a simple formula to calculate the similarity of two strings based on their respective lengths, number of common characters, and number of character transpositions (Jaro 1978, p. 87). Using contemporary notation, Jaro's formula is given by:

$$sim_j = \begin{cases} 0 & \text{if } m = 0 \\ \frac{1}{3}\left(\frac{m}{|s|} + \frac{m}{|t|} + \frac{m-\tau}{m}\right) & \text{otherwise,} \end{cases}$$

where $|s|$ and $|t|$ are the lengths of the strings being compared (i.e., s and t), m is the number of matching characters (i.e., characters that are identical and no further apart than $\lfloor \max(|s|,|t|)/2 \rfloor - 1$), and τ is the number of character transpositions (i.e., half the number of matching characters in a different sequence order). If no characters match between strings s and t, $m = 0$, and their Jaro similarity (sim_j) is 0. On the

other hand, if every character matches and their character sequences are the same, $|s| = |t| = m$ and $\tau = 0$; thus, $sim_j = 1$.

Based on evidence that "the probability of keypunch errors increased as the character position in a string moved from the left to the right" (Winkler 2006), Winkler modified Jaro's similarity to reward consecutive matching characters at the beginning of compared strings (Winkler 1990). Specifically, if sim_j for strings s and t exceeds a predetermined value (b_t, the boost threshold), then sim_j is augmented by the product of the length of s and t's common prefix ($l \leq 4$), a scaling factor (p), and the unrealized similarity ($1 - sim_j$). The formula for this adjustment, known as the Jaro-Winkler similarity (sim_{jw}), is given by:

$$sim_{jw} = \begin{cases} sim_j & \text{if } sim_j < b_t \\ sim_j + (l \cdot p \cdot (1 - sim_j)) & \text{otherwise,} \end{cases}$$

where sim_j is as defined earlier (Winkler 1990), and p and b_t are typically set at 0.1 and 0.7, respectively.

To illustrate the impact of Winkler's adjustment, consider the calculation of sim_j and sim_{jw} for the string pairs $\{s =$ "*alison*", $t_1 =$ "*alisa*"$\}$ and $\{s =$ "*alison*", $t_2 =$ "*mason*"$\}$. For both string pairs, $|t_1| = |t_2| = 5$, so identical characters are considered a match if they are within $\lfloor \max(6,5)/2 \rfloor - 1 = 2$ positions of each other. Looking at "*alison*" and "*alisa*", the first four character positions match in the same sequence order; thus, $m = 4$, $\tau = 0$, and $sim_j = \frac{1}{3}\left(\frac{4}{6} + \frac{4}{5} + \frac{4}{4}\right) = 0.8222$. Similarly, for "*alison*" and "*mason*", the last three character positions match exactly, while the "*a*" in "*alison*" is within two character positions of the "*a*" in "*mason*". Once again, the sequences of the four matching characters (i.e., "*ason*" and "*ason*") are the same and $sim_j = 0.8222$. However, when common prefixes are considered, the situation changes. Specifically, "*alison*" and "*alisa*" share "*alis*"'; thus, $l = 4$ and $sim_{jw} = 0.8222 + (4 \cdot 0.1 \cdot (1 - 0.8222)) = 0.8933$. On the other hand, "*alison*" and "*mason*" have no common prefix, and $sim_{jw} = sim_j = 0.8222$. In sum, although the Jaro similarities suggest the distances from "*alison*" to "*alisa*" and "*alison*" to "*mason*" are equal, the J-W similarities suggest "*alison*" is closer to "*alisa*."

Using the R statistical software package `stringdist`, the J-W similarity of the string pair $\{s,t\}$ can be calculated in $O(|s||t|)$ time (Van der Loo 2014, p. 120). When used to deduplicate n strings, this requires the calculation of $\frac{n(n-1)}{2}$ similarities, which is $O(n^2)$. Given sim_{jw}'s quadratic time complexity and relationship to string lengths, Dreßler and Ngonga Ngomo (2017) derived a length-based filter "based on the insight that large length differences are a guarantee for poor [J-W] similarity" (p. 187). As seen below, their filter provides an upper bound for sim_{jw} ($\theta(s,t)$) using only $|s|$ and $|t|$ (where $|s| \leq |t|$), eliminating the need to calculate sim_{jw} for string pairs where $\theta(s,t)$ is less than θ (Dreßler and Ngonga Ngomo 2017).

$$sim_{jw} \leq \frac{2}{3} + \frac{|s|}{3|t|} + l \cdot p \cdot \left(\frac{1}{3} - \frac{|s|}{3|t|}\right) = \theta(s,t).$$

For instance, if two strings have the same length, $|s| = |t|$, and $\theta(s,t) = \frac{2}{3} + \frac{1}{3} + 4 \cdot 0.1 \cdot (\frac{1}{3} - \frac{1}{3}) = 1$. Intuitively, this makes sense, as equally sized strings can be identical, and $sim_{jw} = 1$ for identical strings. No computational savings are possible. Alternatively, if one string is twice as long as the other (i.e., $|t| = 2|s|$), $\theta(s,t) = \frac{2}{3} + \frac{1}{6} + 4 \cdot 0.1 \cdot (\frac{1}{3} - \frac{1}{6}) = 0.9$. If we assume $\theta = 0.9$, then string pairs with $|t| > 2|s|$ can safely be ignored as potential matches, as sim_{jw} must be less than $\theta(s,t) = \theta$.

The above discussion highlights the critical role θ plays in entity deduplication, not only in identifying potential matches but also in discarding likely non-matches. Accordingly, θ's assigned value deserves careful consideration, and Table 1 provides a sample of θs reported in papers employing the J-W algorithm. As Table 1 shows, θ typically lies in the interval $[0.75, 0.95]$; however, the method for assigning its value is rarely documented. Moreover, using a static value for θ, especially in the presence of shorter strings with typos, is potentially problematic. As Li et al. (2014) note, the J-W Algorithm is "likely to fail for record pairs containing short names with typographical errors, because, faced with the same errors occurring in strings with different lengths, the JWSS [i.e., sim_{jw}] for two 'short strings' is often more degraded than the JWSS for two 'long strings'" (p. 378).

Table 1: θ's observed in the literature.

θ	Assignment Method	Source
0.75-0.95	Not specified	(Marukatat 2009)
0.80	Not specified	(Grannis et al. 2004)
0.80	Not specified	(Szekely et al. 2013)
0.80-0.90	Not specified	(Bjørkelund et al. 2012)
0.85	Hand-tuned	(Crim et al. 2005)
0.90	Not specified	(Cohen et al. 2003)
0.90	Not specified	(Gali et al. 2016)
0.90	Not specified	(Nunes et al. 2012)
0.95	Optimized for precision	(Spitters et al. 2010)

Ultimately, the choice of θ should be fit for the task and data at hand. In the context of identifying terror financiers, several things stand out. First, there is a limited supply of analysts to manually review probable matches - *precision matters*. Second, missing potential matches has operational, potentially deadly consequences - *recall matters*. Finally, names are often short, and misspelled names are common (Tenore 2012) - *typical values for θ may not apply*. Section 3 addresses these considerations directly in a dynamic, simulative way.

3 METHODOLOGY

Defining a proper θ is somewhat trivial given data containing a representative sample of known name misspellings. The process involves calculating the J-W similarity (sim_{jw}) values between names and typos (name-typo pairs) and names and other names (name-name pairs). Then, theoretical precision and recall can be estimated at every hypothetical θ by counting the number of typo-name and name-name matches on either side of θ. Unfortunately, such name misspelling data is not readily available, necessitating the development of a mechanism to create realistic synthetic data. The discussion that follows pays particular attention to this typo-generating methodology as the results rely heavily on this function.

3.1 Simulation Methodology

As shown in Figure 1, the main input to the simulation is a list of names. This study used 5,454 first names and 86,161 last names from the 1990 United States Census (United States Census Bureau. 1990). Each name was passed through a typo-generating algorithm 100 times to create a set of realistic misspellings of the name (step 1). These typos were then compared to the original names using sim_{jw} (step 2). The process resulted in a large (9,161,500 row) data set containing 100 realistic typos paired with every name (typo-name pairs) from the census data.

After generating the typo data, the next step was to create a similar data set by comparing each real name with 100 other real names (step 3). This process creates another 9,161,500 row data set. As with the typo-name data, sim_{jw} was used to compare the each name-name pair (step 4). Because sim_{jw} is by definition sensitive to the string lengths of the strings being compared, it was important to compare names to other names with the same string length distribution as the generated typo data set. In a sense, this represents the worst-case scenario when attempting to discriminate between name-typo and name-name comparisons because string length will not artificially lower the name-name sim_{jw}.

For example, suppose we generated the following three typos for the name "Ian": "Iaan", "Ien", and "In". When generating the corresponding name-name data, we would use names with the same lengths as the typos, such as: "Matt", "Sam", and "Ed". If, for example, we used a longer name like "Benjamin" to compare to "Ian", it would clearly yield a lower sim_{jw} simply because of the name length disparity. By selecting names for the name-name data as described above, we ensure that sim_{jw} differences for typo-name and name-name pairs are not conflated with sim_{jw} differences caused by string length disparity alone.

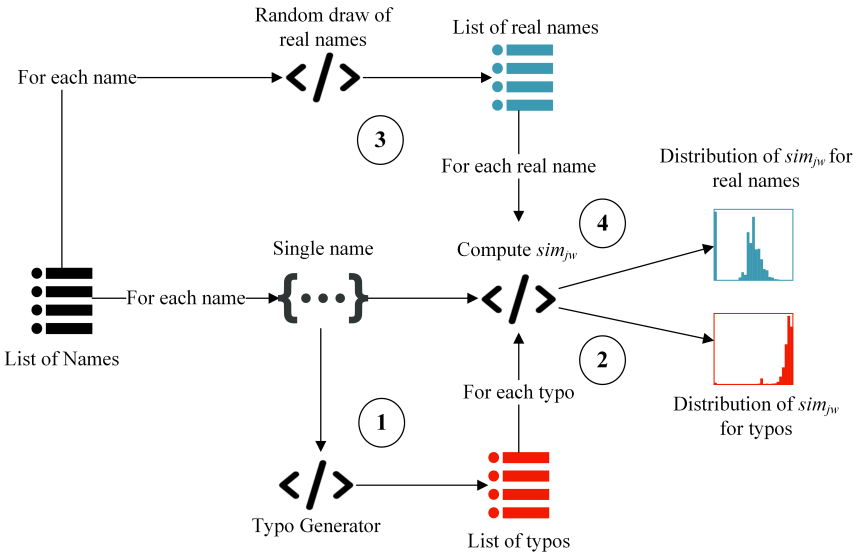

Figure 1: Simulation methodology.

3.2 Typo Generation Methodology

The validity of the synthetic data generated in Figure 1 depends on the ability of the typo generator (step 1) to create realistic variation in strings. In previous studies, researchers used a range of methods to introduce errors such as additions, deletions, substitutions, and transpositions (Peng et al. 2012; Li et al. 2014). Some studies also leveraged generative concepts such as keyboard proximity and drawing error types/locations from named distributions to introduce more realistic errors (Gray et al. 1994).

While these generative methodologies may produce realistic typos in some circumstances, the authors of this paper favored an approach that used empirical distributions observed in real-world typos to create synthetic data. Rather than attempting to recreate the inherently nebulous human process of making errors, our proposed methodology allows the algorithm to learn from a corpus of actual typos. In addition to producing realistic typos, this method also allows a researcher to tune the typo generator for specific domains by providing a custom typo corpus. Further research is required to determine whether this method generates more realistic typos than those commonly employed in other research, but preliminary testing confirms that typos generated using the process described in Figure 2 are corrected by word-processing software with similar accuracy to the typo corpus used in this study.

The inputs to the typo-generating algorithm are a corpus of typos and a string to modify (in this case, a name). The corpus used in this study is a combination of typos from Wikipedia's List of Common Misspellings (Wikipedia. 2019) and another curated list of common misspellings (Dumbtionary. 2007). Together, these sources create a corpus of 8,893 unique typos. Using a combination of Python's `difflib` (Python Software Foundation. 2019) and a custom algorithm, we extracted the operation(s) required to go from the original word to the typo for each word-typo pair. The empirical distributions shown in the margins of Figure 2 were derived from this data.

Because string length has a significant effect on sim_{jw}, the typo generating algorithm enforces that the final string distance between the original word and the generated typo is drawn from a data-derived empirical distribution (step 1). If the final length is larger than the original string length, the algorithm draws the location of the error(s) and the specific character(s) to add (step 2a). If the final length is smaller than the original length, the algorithm draws the location(s) of the character(s) to remove (step 2b).

After assessing additions or subtractions, the algorithm draws the number of transpositions and substitutions (step 3), including the locations and characters required to make these modifications as needed

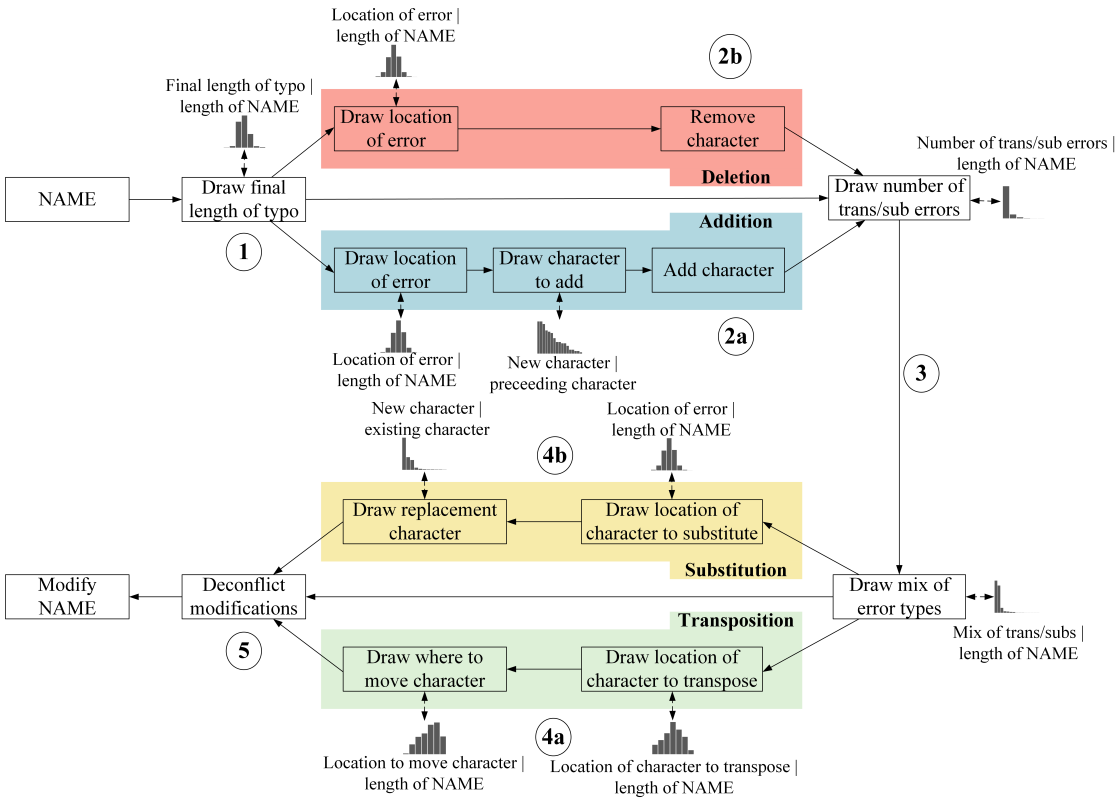

Figure 2: Typo generator methodology.

(steps 4a and 4b). The final step (step 5) assures that there are no conflicts (e.g., the same character was transposed then deleted) before returning the final synthetic typo.

We wrote the simulation in a combination of R, Python, and C++ and ran it in parallel on 30 cores. The typo generator algorithm introduces variability in runtime because the final step (deconflicting changes) may be impossible for a given set of proposed modifications, requiring the algorithm to re-run until a legitimate solution is found. However, to give a general idea of the computation time, one example run of the full simulation took 125 minutes.

4 RESULTS

Using the data generated in the simulation described in Section 3, we set hypothetical θ values between 0 and 1 and calculated the proportion of name-name pairs above the threshold (i.e., how many false positives were selected) and name-typo pairs below the threshold (i.e., how many true positives were missed). The results are shown in Figure 3. For visual simplicity, the results are aggregated by original string length (i.e., the length of the census names for which typos were generated).

Denoted by the dashed vertical lines in Figure 3, the points where the red and blue lines cross are logical values for θ as they balance the minimization of false positives and true negatives. One could certainly argue that precision or recall should be favored at the expense of the other for a specific application, and these curves would describe the effect of shifting the threshold in either direction. Additionally, it is clear that the recommended θ is not the same for all string lengths.

Figure 4 shows the recommendations for θ at every combination of string lengths from three to nine. These suggested θ values exhibit a few interesting characteristics. First, same-to-same string length

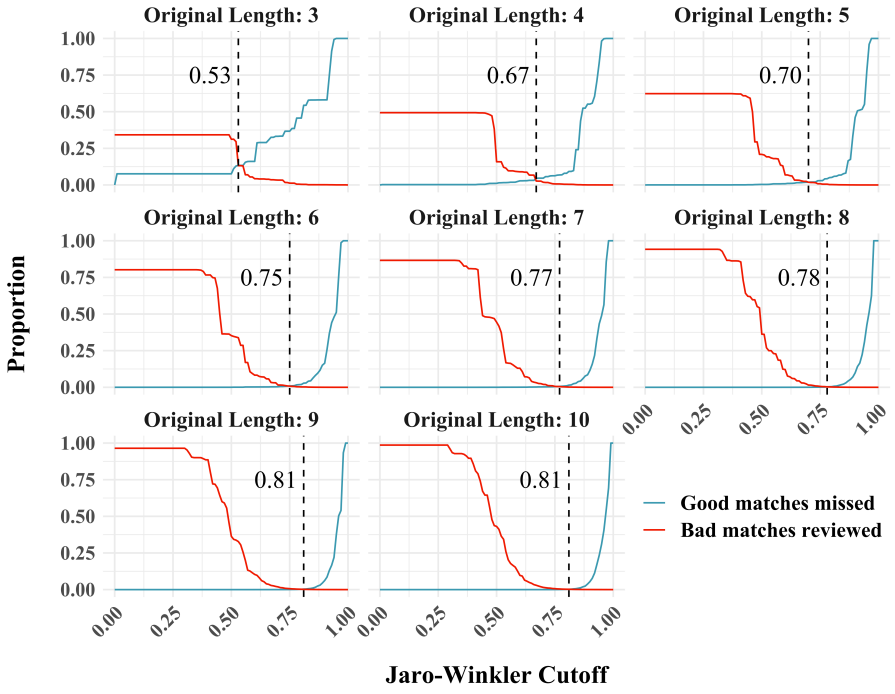

Figure 3: Simulation results for all name-name and name-typo pairs.

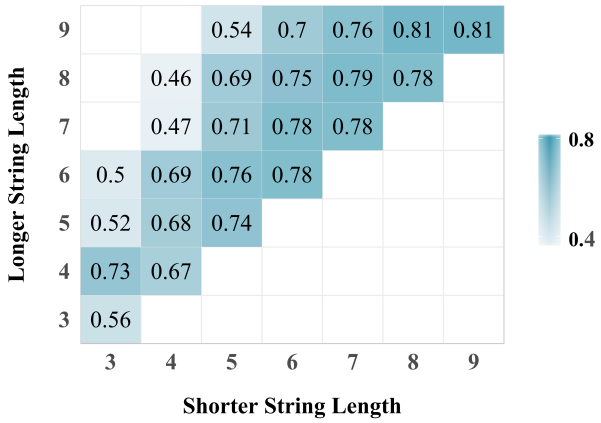

Figure 4: Suggested θ by string length.

comparisons (e.g., 3-to-3, 4-to-4) require lower θ values for smaller strings. This is consistent with the observation that sim_{jw} tends to be higher when comparing larger strings. Second, θ decreases as the difference between the two strings increases. This is also expected behavior because the maximum possible sim_{jw} decreases as the string length distance between strings increases.

5 DISCUSSION

The results presented in Section 4 clearly suggest that θ should be dependent on the string lengths of the names being compared. In an effort to quantify the benefit of this new approach, we compared the suggested approach (using a dynamic θ based on string length) to traditional approaches that use the static values of θ seen in Table 1. Figure 5 shows the results of this analysis.

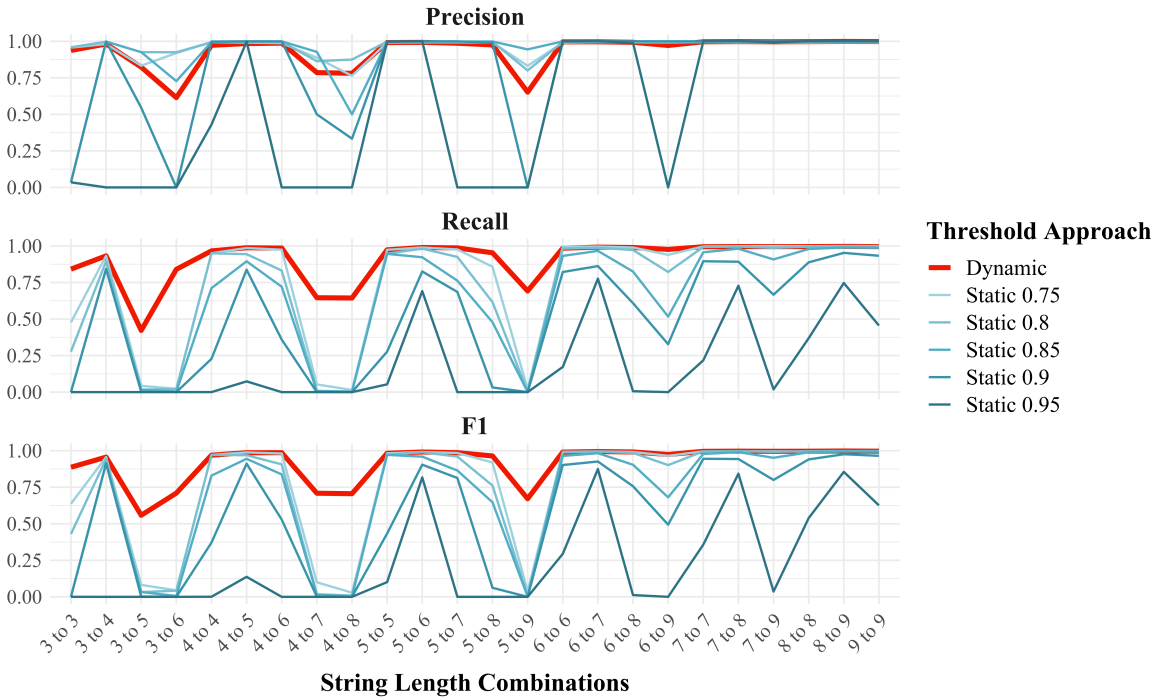

Figure 5: Precision, recall, and F1 for various string length combinations.

When considering precision, there are several string length combinations for which the static approaches outperform the dynamic approach. In all of these cases, however, the recall of the static approaches is dramatically low. In the counter-terrorism field, low recall is unacceptable as it opens up the possibility of losing track of important entities with potentially catastrophic consequences. On the other hand, the dynamic approach maximizes recall with limited detriment to precision. This is evident in the fact that the dynamic approach has a greater or equal F1 (the harmonic mean of precision and recall) for every string length combination, over all static thresholds. Because of the CTFI domain's insistence on high recall and performance on F1, the dynamic approach is clearly superior.

Even in domains that require high precision at the expense of recall, these results are informative. It is commonly accepted that precision and recall are traded-off such that a higher θ results in higher precision and lower recall. In practice, we can see that setting θ too high (e.g., 0.9) results in significant detrimental effects on precision for some string length combinations. This occurs because some string length combinations are mathematically (or practically) limited to a sim_{jw} score that is below the threshold. For example, setting θ to 0.9 will never return a potential match when comparing a string of length 5 to a string of length 9, so precision and recall are both driven to zero.

The benefits of the dynamic approach are most clearly visible when comparing strings of disparate lengths. While it is true that name-typo matches are more likely to be within plus or minus a single string length of the original name, there are many cases where this is not true. Clearly, using a static approach limits one's ability to detect typos in these circumstances.

It is important to note that using dynamic θ values will not substantially add to the computational complexity of the record linkage process. This is particularly important in CTFI as data sets are often large and timeliness is critical. Furthermore, record linkage calculations in this domain often occur on closed-off networks where access to large-scale cluster computing is rare. Implementing the dynamic threshold method would be possible without any significant changes to the hardware or software on which these calculations are currently executed.

6 LIMITATIONS AND FUTURE WORK

While the authors are confident in the results presented in this paper, there are several aspects of the methodology that would benefit from further research. Perhaps most significant among these limitations surrounds the concept of typo generation. It is common in other research to use very basic and simplified methods to introduce errors in strings. Some of these studies acknowledge that sim_{jw} may be sensitive to the type of errors in a string (Li et al. 2014). This study, as others before it, highlighted sim_{jw}'s sensitivity to string length. Because of these sensitivities, it is important to use data that mirrors errors as they occur in real-world applications.

The typo generator described in Figure 2 is designed to generate realistic typos, and it appears to do exactly that based on our testing. The validity of the data generated by the algorithm is, however, entirely dependent on the typo corpus that is used to generate the empirical distributions. The algorithm seems to faithfully mimic the data that we were able to find (in as much as Microsoft Word's spell checking software corrected synthetically generated typos correctly at a statistically similar rate to real typos), but it is likely that name-based typos are systematically different from word-based typos. Ultimately, a name-based typo corpus should be developed and used in future studies. Furthermore, it is likely that typos for names from different regions of the world are systematically misspelled in different ways. With a sufficiently large data set containing typos of region-specific names, one could construct a region-specific typo generator to provide even more reliable results.

In addition to being subject to different types of misspellings, names from different regions are also likely to behave differently with regard to sim_{jw}. For example, names from areas of the world where last names share common prefixes or suffixes would likely be more difficult to discriminate using sim_{jw}. As a result, the suggested values of θ could be tuned to specific regions, or even further refined by studying the sim_{jw} scores that come from comparing names of different regions to each other. It is also likely that region-specific θs will differ for first and last names. This is supported by preliminary research showing small variations in suggested θ when using only first or last names.

Continuing with the idea of the importance of prefixes and suffixes, we recommend further research on the boost threshold (b_t) described in the original Jaro-Winkler paper (Winkler 1990). The sim_{jw} metric applied in this paper used 0.1 for p and 3 for l, but it is possible that there are other optimal values for these parameters – either universally or under specific circumstances.

Based on preliminary research, using a modified sim_{jw} approach that adds a boost threshold to the end of a string may also provide better ways to find typos. Applying this modified sim_{jw} metric to a concatenation of the first and last name where the last name is reversed (e.g., John Smith becomes JohnhtimS) showed promising results, but more research into this idea is required. This is a particularly interesting concept because it would greatly reduce the number of computations required to process a data set.

It will also be important to observe the performance of this algorithm on real-world data sets to confirm the validity of the dynamic θ approach. In practice, it is unrealistic to expect to find a large corpus of tagged record linkage data, especially in the CTFI domain, as generating such a data set would require a prohibitively long amount of analyst time. Accordingly, the best way to test this methodology is by applying it to data that has already been processed by the currently accepted techniques. Evaluating our methodology directly against the currently accepted practices is the only way to truly test its utility.

7 CONCLUSION

This study demonstrates that using dynamic values of θ can improve recall while generally maintaining precision in record linkage methodologies. While this paper recommends significant follow-on research that will likely further refine the conclusions of this study, the dynamic thresholds recommended in this paper should benefit current record linkage efforts, especially in the CTFI domain. Given that the dynamic θ approach does not complicate record linkage computations, there is little reason to continue using static θ values.

ACKNOWLEDGMENTS

The views expressed herein are those of the authors and do not reflect the official policy or position of the United States Military Academy, the Department of the Army, or the Department of Defense.

REFERENCES

18 U.S.C §1956. 2011. "Laundering of Monetary Instruments". https://www.law.cornell.edu/uscode/text/18/1956, accessed 5th September 2019.

18 U.S.C. §2339A. 2011. "Providing Material Support to Terrorists". https://www.law.cornell.edu/uscode/text/18/2339A, accessed 5th September 2019.

18 U.S.C. §2339B. 2011. "Providing Material Support or Resources to Designated Foreign Terrorist Organizations". https://www.law.cornell.edu/uscode/text/18/2339B, accessed 5th September 2019.

18 U.S.C. §2339C. 2011. "Prohibitions Against the Financing of Terrorism". https://www.law.cornell.edu/uscode/text/18/2339C, accessed 5th September 2019.

18 U.S.C. §2339D. 2010. "Receiving Military-Type Training from a Foreign Terrorist Organization". https://www.law.cornell.edu/uscode/text/18/2339D, accessed 5th September 2019.

American Security Project. 2012. "Threat Finance and Financial Intelligence". https://www.americansecurityproject.org/asymmetric-operations/threat-finance-and-financial-intelligence/, accessed 19th April 2019.

Bahney, B., H. J. Shatz, C. Ganier, R. McPherson, and B. Sude. 2010. *An Economic Analysis of the Financial Records of Al-Qa'ida in Iraq*. Santa Monica, CA: RAND Corporation.

Bjørkelund, E., T. H. Burnett, and K. Nørvåg. 2012. "A Study of Opinion Mining and Visualization of Hotel Reviews". In *Proceedings of the 14th International Conference on Information Integration and Web-based Applications & Services*, edited by D. Taniar, E. Pardede, M. Steinbauer, and I. Khalil, 229–238. New York, New York.

Cohen, W., P. Ravikumar, and S. Fienberg. 2003. "A Comparison of String Metrics for Matching Names and Records". In *Proceedings of the KDD-03 Workshop on Data Cleaning and Object Consolidation*, Volume 3, 73–78, August 24th-27th, Washington, DC: American Association for Artificial Intelligence.

Crim, J., R. McDonald, and F. Pereira. 2005. "Automatically Annotating Documents with Normalized Gene Lists". *BMC Bioinformatics* 6(1):1–7.

Cunningham, M. 2017. "Bankrupting Terrorism: Alumnus Hits Extremists Where It Hurts". *Carnegie Mellon Today*. Carnegie Mellon University, Pittsburgh, Pennsylvania. https://www.cmu.edu/cmtoday/publicpolicy_technology/bankrupting-terrorism-alumnus-hits-extremists-where-it-hurts/index.html, accessed 23rd July 2019.

Dreßler, K., and A.-C. Ngonga Ngomo. 2017. "On the Efficient Execution of Bounded Jaro-Winkler Distances". *Semantic Web* 8(2):185–196.

Dumbtionary. 2007. "Dumbtionary: A Dictionary of Misspelled Words". http://www.dumbtionary.com/, accessed 21st March 2019.

Exec. Order No. 13,224, 3 C.F.R. 49079. 2001. "Blocking Property and Prohibiting Transactions With Persons Who Commit, Threaten To Commit, or Support Terrorism". https://www.treasury.gov/resource-center/sanctions/Documents/13224.pdf, accessed 5th September 2019.

Felligi, I. P., and A. B. Sunter. 1969. "A Theory for Record Linkage". *Journal of the American Statistical Association* 64(328):1183–1210.

Gali, N., R. Mariescu-Istodor, and P. Fränti. 2016. "Similarity Measures for Title Matching". In *Proceedings of the 23rd International Conference on Pattern Recognition (ICPR),* December 5th-8th, Cancun, Mexico, 1548–1553.

Goerger, S. R., O. Brooks, and D. Ahner. 2016. "MORS Prizes and Awards". *Phalanx* 49(3):22–25.

Grannis, S. J., J. M. Overhage, and C. J. McDonald. 2004. "Real World Performance of Approximate String Comparators for use in Patient Matching". In *Medinfo 2004 - Proceedings of the 11th World Congress on Medical Informatics*, edited by M. Fieschi, E. Coiera, and Y. J. Li, 43–47. Washington, DC: IOS Press.

Gray, J., P. Sundaresan, S. Englert, K. Baclawski, and P. J. Weinberger. 1994. "Quickly Generating Billion-Record Synthetic Databases". In *Proceedings of the 1994 ACM SIGMOD International Conference on Management of Data*, 243–252. New York, New York: Association for Computing Machinery.

Huddleston, S., I. Kloo, R. Greenway, M. Haskell, and A. Usher. 2016. "Data Science for Threat Finance Intelligence". In *Proceedings of the 84th Military Operations Research Society Symposium*. June 20th-23rd, Quantico, Virginia, 16446.

International Emergency Economic Powers Act, 50 U.S.C. §1701-1708. 2011. "International Emergency Economic Powers". https://www.law.cornell.edu/uscode/text/50/chapter-35, accessed 5th September 2019.

Jaro, M. 1978. *UNIMATCH: A Record Linkage System - Users Manual*. Washington, D.C.: U.S. Bureau of the Census.

Li, X., A. Guttmann, S. Cipière, L. Maigne, J. Demongeot, J.-Y. Boire, and L. Ouchchane. 2014. "Implementation of an Extended Fellegi-Sunter Probabilistic Record Linkage Method Using the Jaro-Winkler String Comparator". In *Proceedings*

of IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), 375–379. Piscataway, New Jersey: IEEE.

Marshall, J. 1947. "Canada's National Vital Statistics Index". *Population Studies* 1(2):204–211.

Marukatat, R. 2009. "Dipper: A Data Integration with Privacy Protection Environment". In *Proceedings of the International MultiConference of Engineers and Computer Scientists*. March 18th-20th, Kowloon, Hong Kong, 750–754.

Money Laundering Control Act, 18 U.S.C §981. 2011. "Civil Forfeiture". https://www.law.cornell.edu/uscode/text/18/981, accessed 5th September 2019.

Newcombe, H. B., J. M. Kennedy, S. Axford, and A. P. James. 1959. "Automatic Linkage of Vital Records". *Science* 130(3381):954–959.

Nunes, A., P. Calado, and B. Martins. 2012. "Resolving User Identities Over Social Networks Through Supervised Learning and Rich Similarity Features". In *Proceedings of the 27th Annual ACM symposium on Applied Computing*, 728–729. New York, New York: ACM.

Peng, T., L. Li, and J. Kennedy. 2012. "A Comparison of Techniques for Name Matching". *GSTF Journal on Computing (JoC)* 2(1):55–61.

Python Software Foundation. 2019. "The Python Standard Library". https://docs.python.org/3/library/, accessed 23rd July 2019.

Spitters, M., R. Bonnema, M. Rotaru, and J. Zavrel. 2010. "Bootstrapping Information Extraction Mappings by Similarity-Based Reuse of Taxonomies". In *Proceedings of the CEUR Workshop*, Volume 673.

Szekely, P., C. A. Knoblock, F. Yang, X. Zhu, E. E. Fink, R. Allen, and G. Goodlander. 2013. "Connecting the Smithsonian American Art Museum to the Linked Data Cloud". In *Proceedings of the 10th Extended Semantic Web Conference*, edited by P. Cimiano, O. Corcho, V. Presutti, L. Hollink, and S. Rudolph, 593–607. Berlin, Germany: Springer.

Talburt, J. 2011. *Entity Resolution and Information Quality*. New York, New York: Morgan Kaufmann.

Taxay, M., L. Schneider, and K. Didow. 2014. "What to Charge in Terrorist Financing or Facilitation Case". *United States Attorneys' Bulletin* 62(5):9–15.

Tenore, M. J. 2012. "Why Misspelled Names are so Common & What Journalists are Doing to Prevent Them". https://www.poynter.org/reporting-editing/2012/why-misspelled-names-are-so-common-what-journalists-are-doing-to-avoid-them/, accessed 19th April 2019.

United States Census Bureau. 1990. "Frequently Occurring Surnames from Census 1990 – Names Files". https://www.census.gov/topics/population/genealogy/data/1990_census/1990_census_namefiles.html, accessed 21st March 2019.

U.S. Department of the Treasury. 2017. *Terrorist Assets Report Calendar Year 2017 - Twenty-sixth Annual Report to the Congress on Assets in the United States Relating to Terrorist Countries and International Terrorism Program Designees*. https://www.treasury.gov/resource-center/sanctions/Programs/Documents/tar2017.pdf, accessed 23rd July 2019.

U.S. v. The Bank of Nova Scotia, 691 F.2d 1384. 1982. https://openjurist.org/691/f2d/1384/grand-jury-proceedings-united-states-v-bank-of-nova-scotia, accessed 5th September 2019.

USA PATRIOT Act, Pub. L. 107-56 (H.R. 3162), 115 Stat. 272, 107th Cong., 1st Sess. 2001. "Uniting and Strengthening America by Providing Appropriate Tools Required to Intercept and Obstruct Terrorism Act". https://www.gpo.gov/fdsys/pkg/PLAW-107publ56/html/PLAW-107publ56.htm, accessed 5th September 2019.

Van der Loo, M. P. 2014. "The Stringdist Package for Approximate String Matching". *The R Journal* 6(1):111–122.

Wikipedia. 2019. "Lists of Common Misspellings". https://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings, accessed 21st March 2019.

Winkler, W. E. 1990. "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage". In *Proceedings of the Section on Survey Research Methods*, 354–359. American Statistical Association.

Winkler, W. E. 2006. "Overview of Record Linkage and Current Research Directions". Technical Report No. 2, Statistical Research Division, U.S. Census Bureau, Washington, D.C.

AUTHOR BIOGRAPHIES

IAN KLOO is a data scientist and instructor in the United States Military Academys (USMA's) Department of Systems Engineering (DSE). He earned a Masters in Policy Analytics from Carnegie Mellon University in 2014 and a BBA from The College of William and Mary in 2011. His email address is ian.kloo@westpoint.edu.

MATTHEW DABKOWSKI is an Academy Professor in USMA's DSE, currently serving as the Director of the Systems Engineering Program. He holds a BS in Operations Research from USMA, an MS in Systems Engineering from the University of Arizona (UA), and a PhD in Systems and Industrial Engineering from the UA. His email address is matthew.dabkowski@westpoint.edu.

SAM HUDDLESTON is a senior data scientist at Biocore LLC where he leads teams that support the health and safety of athletes. He holds a BS in Mechanical Engineering from USMA, an MS in Systems Engineering from the University of Virginia (UVA), and a PhD in Systems and Information Engineering from UVA. His email address is shuddleston707@gmail.com.