Finding and Rating Personal Names on Drives for Forensic Needs

Rowe, Neil C.

Finding and Rating Personal Names on Drives for Forensic Needs

Neil C. Rowe(✉)

Computer Science, U.S. Naval Postgraduate School, Monterey, CA 93940, USA
ncrowe@nps.edu

Abstract. Personal names found on drives provide forensically valuable information about users of systems. This work reports on the design and engineering of tools to mine them from disk images, bootstrapping on output of the Bulk Extractor tool. However, most potential names found are either uninteresting sales and help contacts or are not being used as names, so we developed methods to rate name-candidate value by an analysis of the clues that they and their context provide. We used an empirically based approach with statistics from a large corpus from which we extracted 303 million email addresses and 74 million phone numbers, and then found 302 million personal names. We tested three machine-learning approaches and Naïve Bayes performed the best. Cross-modal clues from nearby email addresses improved performance still further. This approach eliminated from consideration 71.3% of the addresses found in our corpus with an estimated 67.4% F-score, a potential 3.5 times reduction in the name workload of most forensic investigations.

Keywords: Digital forensics · Personal names · Extraction · Email addresses
Phone numbers · Rating · Filtering · Bulk Extractor · Naïve Bayes
Cross-modality

1 Introduction

When we scan raw drive images we can often find information about people who have used the drives and their contacts, and this information is often important in criminal and intelligence investigations using digital forensics. We call these "personal artifacts" and others have called them "identities" [8].

Our previous work [14] developed a methodology for finding interesting email addresses on drives using Bayesian methods and graphing their social networks. Personal names could provide more direct information than email addresses about users of a drive and their contacts. Candidates can be found using lists of known names. They can be combined with the email data and other information to build a more complete picture of users. However, we only are interested in "useful" names, names relevant to most criminal or intelligence investigations. We define "useful" to exclude those not being used as names, those that are business and organizational contacts, those associated with software and projects, those in fiction, and those that occur on many drives. (These criteria would need modification for an investigation involving an associated organization or an important common document.) We estimate that useful names are only 30% of the names found on drives. We shall test the hypothesis that a set of easily calculable local clues can reliably rate usefulness of name candidates with precision well beyond that of name-dictionary lookup alone, so that most candidates irrelevant to most investigations can be excluded.

2 Previous Work

Finding personal data is often important in forensics. Web sites provide much useful information about people, but only their public faces, and email servers provide much semi-public data [9]. Registries, cookie stores, and key chains on drives can provide rich sources of personal data including names [11], but they often lack the deleted and concealed information that may be critical in criminal and intelligence investigations. Thus this work focuses on more thorough search of a drive image for personal names.

Tools for "named entity recognition" in text [1] can find locations and organizations as well as personal names. Most learn sequences of N consecutive words in text that include named entities [12]. This has been applied to forensic data [15]. Capitalization, preceding articles, and absence from a standard dictionary are clues. But these methods do not work well for forensic data since our previous work estimated that only 21.9% of the email addresses in our corpus occurred within files. Secondly, only a small fraction of the artifacts within files occur within documents or document fragments for which linguistic sequence models would be helpful; for instance, articles like "the" are rare in most forensic data. Instead, other clues from the local context are needed to find artifacts. Thirdly, most forensic references to people are business and vendor contacts, not people generally worth investigating, a weakness in the otherwise interesting work of [8]. For these reasons, linguistic methods for named-entity recognition do not work well for forensic tasks.

An alternative is to make a list of names and scan drives for them using a keyword search tool. We explored this with the well-known Bulk Extractor tool [2] and its –F argument giving a file of the names. But full scans are time-consuming. An experiment extracting all delimited names from a single 8.92 gigabyte drive image in EWF (E01) format took around ten days, given the additional requirement of breaking the names into 927 runs to satisfy Bulk Extractor's limit of 300 per run. Furthermore, the percentage yield of useful names was low. Because names had to be delimited, the output did not include run-on personal-name pairs in email addresses, so it found only 21% of the name occurrences found by the methods to be described. It did find a few new name candidates beyond those of our methods since it searched the entire drive, but 98.7% of these candidates in a random sample were being used as non-names (e.g. "mark" and "good"). Another criticism of broad scanning is that finding an isolated personal name is less useful than finding it near other personal information, since context is important in an investigation; [14] showed that email addresses more than 22 bytes distant on a drive were statistically uncorrelated, and names are probably similar. It is also difficult to confirm the validity of names without other nearby personal artifacts, making it hard to train and test on them.

This work will thus pursue an approach of examining context in which useful personal names are more likely to occur in routine Bulk Extractor output, rating the candidates using machine-learning methods, and selecting the best ones. The ultimate goal of extraction of personal artifacts in forensics will be to construct graphs modelling human connections. This can provide a context for the artifacts as well as aid in name disambiguation [3], and permits cross-drive analysis to relate drives [4].

3 Test Setup

This work used the Real Data Corpus [6], a collection of currently 3361 drives from 33 countries that is publicly available subject to constraints. These drives were purchased as used equipment and represent a range of business, government, and home users. We supplemented this with images of twelve substantial computers of seven members of our research team.

The first step was to run the Bulk Extractor tool [2] to get all email addresses (including cookies), phone numbers, bank-card numbers, and Web links (URLs), along with their offsets on the drive and their 16 preceding and 16 following characters. Such data extraction is often routine in investigations, so we bootstrapped on generally available data. Such extraction can exploit regular expressions effectively and can be significantly faster than name-set lookup. Bulk Extractor can find data in deleted and unallocated storage as well as within many kinds of compressed files [5]. It found 2442 of the drives had email addresses, 1601 had phone numbers, and 10 had bank-card numbers. In total we obtained 303,221,117 email addresses of which 17,484,640 were distinct, and 21,184,361 phone numbers of which 1,739,054 were distinct. As discussed, this research assumed that most useful personal names are near email addresses, phone numbers, and personal-identification numbers. Thus we wrote a tool to extract names from the Bulk Extractor "context" output using a hashed dictionary of possible names. We segmented words at spaces, line terminators, punctuation marks, digits, lower-to-upper case changes, and by additional criteria described in Sect. 4.1.

Our hashed name dictionary had 277,888 personal names obtained from a variety of sources. The U.S. Census Bureau (www.census.gov) has published 95,025 distinct surnames which occurred at least five times in their data 1880-2015 and 88,799 last names in 1990. For international coverage, tekeli.li/onomastikon provided more names. We supplemented this with data from email user names in our corpus split at punctuation marks, looking for those differing in a single character from known names; this found variant names not in existing lists, but had to be manually checked to remove a few errors. We also mined our corpus for the formats like "John Smith <jsmith@hotmail.edu>" and "'John Smith' 555-123-4567" that strongly suggest names in the first two words. Most dictionary names were Ascii, as non-Ascii user names were not permitted by email protocols for a long time and rarely appear in our corpus. We did not distinguish surnames and family names since many are used for both purposes. Note this is a "whitelisting" approach to defining names; a blacklisting approach storing non-names is unworkable because the number of such strings is unbounded and they have too much variety to define with regular expressions.

We also created a list of 809,216 words from all our natural-language dictionaries [13], currently covering 23 languages and 19 transliterations of those, together with counts of words in the file names of our corpus to get a rough estimate of usage rates.

We created another list of 54,918 generic names like "contact" and "sales" from manual inspection of our corpus and translation of those words into all our dictionary languages, to serve as definite non-names for our analysis.

Bulk Extractor provides offsets (byte addresses) on the drive for the artifacts it finds. Offsets for nearby personal names can be computed from these; nearby artifacts are often related. Bulk Extractor gives at least two offsets for compressed files, one of the start of the compressed file and one within the file if it were decompressed; a few files were multiply compressed and had up to six offsets. It is important not to add these numbers since the sums could overlap into the area of the next file on the drive when compression reduced the size of a file. We separate the container address space by taking -10 times the offset of the container and adding to it the sum of the offsets of the artifact within the decompressed container. Since few compressions exceed a factor of 10, this maps offsets of compressed files to disjoint ranges of negative numbers.

4 Analysis of Personal-Name Candidates

Overall, 95.6% of the personal-name candidates our methods extracted were found in Bulk Extractor email records, 1.0% in phone-number records, 0.0% in ccn (bank-card) records, and 3.4% in URL records when we included them. URL paths can refer to personal Web pages, but a random sample of 1088 candidate personal-name candidates found only 71 or 6.5% useful personal names since it found many names of celebrities, fictional characters, and words that are predominantly non-names. It appeared that the "url" plugin provides too many false alarms to be useful. Output of other Bulk Extractor plugins was also unhelpful.

We obtained in total 302,242,805 personal names from 2222 drives with at least one name in our corpus, of which 5,921,992 were distinct (though names like "John" could refer to many people). The number of files on these drives was 61,365,153, so only a few files had personal names. Interestingly, several hundred drives had no recoverable files but many names, apparently due to imperfect disk wiping.

4.1 Splitting Strings to Find Names

Personal names are often run together in email addresses, e.g. "johnjsmith". We can segment these by systematically examining splits of unrecognized words. Usually one should prefer splits that maximize the size of the largest piece since this increases the reliability of the names found. For instance, there are 10,785 personal names of length 4 in our name wordlist (fraction 0.024 of all possible 4-character Ascii words), 39,548 of length 5 (fraction 0.0033), 62,114 of length 6 (fraction 0.00020), 58,119 of length 7 (fraction 0.0000072), and 37,461 of length 8 (fraction 0.00000018). That suggested the following algorithm for splitting to find names:

1. Check if the string is a known word (personal name, generic name, or dictionary word); if so, return it and stop. For instance, "thompson", "help", and "porcupine". However, there must be an exception for hexadecimal strings using digits and the characters "abcdef" for which there can be false alarms for names like "ed" and

"bee". So we exclude words preceded by a digit, but not names followed by a digit since these can be numberings of identical names like "joe37".

2. Check whether the string minus its first, last, first two, last two, first three, or last three characters (in that order) is a known personal name; if so, return the split and stop. For instance, "jrthompson" and "thompsonk".
3. Split the string into two pieces as evenly as possible, and then consider successively uneven splits. Check whether both pieces can be recognized as personal names or dictionary words, and stop splitting if you do. For instance, "johnthompson" and "bigtable" can be split into "john thompson" and "big table", but only the first is a personal name.

Unicode encodings raise special problems. Bulk Extractor often represents these with a "\x" and number, and these can be easily handled. But sometimes it encodes characters in languages like Arabic, Cyrillic, and Hebrew as two characters with the first character the higher-order bits, appearing usually as a control character. We try to detect such two-character patterns and correct them, though this causes difficulties for subsequent offset-difference calculations. More complex encodings of names and addresses used with phishing obfuscation [10] need additional decoding techniques.

4.2 Combining Adjacent Personal Names

Once names are extracted, it is important to recognize multiword names that together identify an individual since these are more specific and useful than the individual names, e.g. "Bobbi Jo Riley". We do this with a second pass through the data, which for our corpus reduced the number of name candidates from 556 million to 302 million. After study of sample data, we determined that names could only be combined when separated by 0 to 4 characters for the cases shown in Table 1. These cases can be applied more than once to the same words, so we could first append "Bobbi" and "Jo", then "Bobbi Jo" and "Riley".

Table 1. Cases for appending names.

Intervening characters	Example	Extracted name
None	johnsmith	John Smith
Space, period, hyphen, or underscore	john_smith	John Smith
Period and space after single-letter name	j. smith	J Smith
Comma and space	smith, john	John Smith
Space, letter, space; or underscore, letter, underscore	john a smith	John A Smith
Space, letter, period, space; or underscore, letter, period, underscore	john a. smith	John A Smith

A constraint applied was that appended names cannot be a subset of one another ignoring case. For instance, for the input "John Smith smithjohn@yahoo.com" we can extract “John”, “Smith”, “smith”, and “john”, and we can combine the first two. But we cannot then combine “John Smith” and “smith” because the latter is a substring of the former ignoring case, though we can combine “smith” with “john”. Another constraint that eliminates many spurious combinations is that the character cases must be consistent between the words appended. For instance, “smith” and “SID” cannot be combined because one is lower-case and one is upper-case; that was important for our corpus because “SID” occurs frequently indicating an identification number. We permit only lower-lower, upper-upper, capitalized-capitalized, and capitalized-lower combinations, with exceptions for a few name prefixes such as “mc”, “la”, and “des” that are inconsistently capitalized.

Overlapping windows found by Bulk Extractor enable finding additional names split across two contexts, as with one context ending with “Rich” and another context starting with “chard”. One can also eliminate duplicate data for the same location found from overlapping Bulk Extractor context strings.

As an example, Table 2 shows example Bulk Extractor output in which we can recognize name candidates “John” at offset 1000008, “Smith” at 1000013, “j” at 1000021, “smith” at 1000022, “Bob” at 1000057, “Jones” at 1000061, and “em” at 1000070. Looking at adjacencies we should recognize three strong candidates for two-word names: “John Smith” at offset 1000008, “J Smith” at 1000021, and “Bob Jones” as 1000056. The first two-word combinations match which makes them both even more likely in context. However, possible nickname “Em” is unlikely to be a personal name here because its common-word occurrence is high, it is not capitalized, and it appears in isolation.

Table 2. Example Bulk Extractor artifacts.

Artifact offset	Artifact	Context
100000021	Address jsmith@hotmail2.com	ylor”\x0A“John Smith” <jsmith@hotmail2.com>, 555-623-1886\x0A”Bo
100000043	Phone number 555-623-1886	smith@hotmail2.com>, 555-623-1886 \x0A”Bob Jones”, <em>Ne
6834950233	Possible bank card number 5911468437490705	222382355433193\x0A5911468437490705 \x0A101333182109778
3834394303	URL (web link) faculty.ucdi.edu/terms.pdf	Terms of use at http://faculty.ucdi.edu/terms.pdf

4.3 Rating Personal-Name Candidates

Personal names matching a names dictionary are not guaranteed to be useful in an investigation. Many names are also natural-language words, and others can label software, projects, vendors, and organizations. So it is important to estimate the probability of a name being useful. We tested the following clues for rating a name:

- Its length. Short names like “ed” are more likely to appear accidentally as in code strings and thus should be low-rated.

- Its capitalization type (lower case, upper case, initial capital letter, or mixed case). The convention to capitalize the initial letter of names provides a clue to them, but is not followed much in the digital world. Again, there must be exceptions for common name prefixes like “Mc”, “De”, “St”, “Van”, and “O” which are often not separated from a capitalized subsequent name.
- Whether the name has conventional delimiters like quotation marks on one or both sides. Table 3 lists the matched pairs of delimiters on names seen at significant rates in our corpus, based on study of random samples.

Table 3. Matched pairs of name delimiters sought.

Front delimiter	Rear delimiter	Front delimiter	Rear delimiter
“	“	<	>
(	)	[	]
<	@	(	@
[	@	>	<
>	@	:	<
:	@	;	<
;	@	‘	‘

- Whether the name is followed by a digit. This often occurs with email addresses, e.g. “joe682”.
- Whether the name is a single word or multiple words created by the methods of Sect. 4.2.
- Whether the name frequently occurs as a non-name, like “main” and “bill”. We got candidates from intersecting the list of known personal names with words that were frequent in a histogram of words used in the file names of our corpus, then manually adding some common non-names missed.
- The count of the word in all the words of the file paths in our corpus.
- The number of drives on which a name occurs. Names occurring on many drives are more likely to be within software and thus be business or vendor contacts. However, a correction must be made for the length of the name, since short names like “John” are more likely to refer to many people and will appear on more drives. Figure 1 plots the natural logarithm of the number of drives against the natural logarithm of the name length for our corpus. We approximated this by two linear segments split at 10.0 characters (the antilog of 2.3 on the graph), which fit formulas in the antilog domain of $59.7 * length^{-1.56}$ (left side) and $3.56 * length^{-0.33}$ (right side). We then divided the observed number of drives for a name by this correction factor. For instance, “john” alone occurred on 1182 drives in our corpus, and “john smith” on 557 drives, for correction factors of 6.87 and 1.64 and normalized values of 172 and 339 respectively, so “john smith” is twice as significant as “john”.
- The average number of occurrences of the name per drive. High counts tend to be local names and likely more interesting.

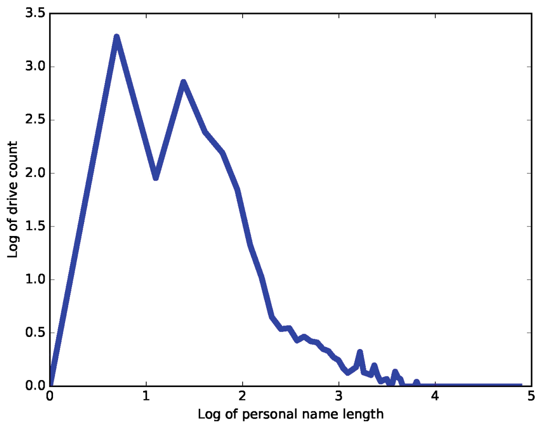

Fig. 1. Natural logarithm of number of drive appearances versus natural logarithm of name length for our corpus.

- Whether there is a domain name in the context window around the personal name that is .org, .gov, .mil, .biz, or a .com, where the subdomain before the .com is not a known mail or messaging server name. This clue could be made more restrictive in investigations involving organizations. This clue is helpful because usually people do not mix business and personal email.

4.4 Experimental Results with a Bayesian Model

We trained and tested the name clues on a training set which was a random sample of 5639 name candidates found by Bulk Extractor on our corpus. For this sample, we manually identified 1127 as useful personal names and 4522 as not, defining "useful" as in Sect. 1. Some names required Internet research to tag properly.

Our previous work developed clues for filtering email addresses as to interestingness using Bayesian methods, and we can use a similar approach for names. Probabilities are needed because few indicators are guaranteed. This work followed a Naive Bayes odds formulation:

$$o(U \mid E_1 \& E_2 \& \ldots \& E_N) = o(U|E_1)o(U|E_2)\ldots o(U|E_N)o(U)^{1-N}$$

We used previously a correction factor of $\lambda = 1$ to handle odds with zero and maximum counts:

$$o(U|E) = (n(U|E) + \lambda o(E))/(n(\sim U|E) + \lambda)$$

We calculated odds for each of the clues from the training/test set by a 100-fold cross-validation, choosing 100 times a random 80% for training and the remaining 20% for testing. Table 4 shows the computed mean odds and associated standard deviations for the clues in the 100 runs. We used maximum F-score as the criterion for setting partitioning thresholds on the four numeric parameters. So when numeric thresholds are given for the clues, they represent the values at which the maximum F-score was obtained for our training set with that parameter alone. We also tested having more than two subranges for each numeric clue, but none of these improved performance significantly. F-score weights recall and precision equally; if this is not desired, a weighted metric could be substituted.

Table 4. Odds on clues for personal names.

Clue	Odds on training set	Standard deviation on training set
Length ≤ 5 characters	0.168	0.006
Length >5 characters	0.272	0.006
All lower case	0.319	0.006
All upper case	0.150	0.015
Capitalized only	0.172	0.006
Mixed case	0.134	0.012
Delimited both sides	0.361	0.009
Delimited on one side	0.301	0.013
No delimiters	0.158	0.004
Followed by a digit	1.243	0.077
No following digit	0.214	0.004
Single word	0.236	0.005
Multiple words	0.249	0.007
Ambiguous word	0.055	0.004
Not ambiguous word	0.294	0.006
≤ 9 occurrences in corpus file names	0.451	0.011
>9 occurrences in corpus file names	0.162	0.004
Normalized number of drives ≤ 153	0.421	0.009
Normalized number of drives >153	0.112	0.004
≤ 399 occurrences per drive	0.189	0.004
>399 occurrences per drive	0.664	0.025
Organizational domain name nearby	0.009	0.001
No organizational domain name nearby	0.760	0.015
Prior to any clues	0.241	0.004

The average best F-score in cross-validation on our training set was 0.6681 at an average threshold of 0.2586 (with recall 80.7% and precision 57.0%). At this threshold, we eliminate from consideration 71.3% of the 302 million personal-name candidates found in our full corpus, and we set that threshold for our subsequent experiments. We also could obtain 90% recall at 48.9% precision and 99% recall at 30.1% precision on the training set, so even investigations needing high recall can benefit from these methods.

Such rates of data reduction do depend on the corpus, as over half the drives in our corpus appear to be business-related. Running time on the full corpus was around 120 h on a five-year-old Linux machine, or about 3.2 min per drive, not counting the time for Bulk Extractor.

All the clues except multiple words appear to be significant alone, either positively or negative. However, it is also important to test for redundancy by removing each clue and seeing if performance is hurt. We found that the capitalization clue was the most redundant since removing it helped performance the most, improving F-score by 0.94% on the full training set, and 0.6744 on 100-fold cross-validation. After capitalization was removed, no other clues were found helpful to remove. So we removed it alone from subsequent testing.

4.5 Results with Alternative Conceptual Models

We also tested a linear model for of the form $t = w_0 + w_1x_1 + w_2x_2 + \ldots + w_{11}x_{11}$ where the w_i values are relative likelihoods. We fit this formula to be 1 for tagged personal names and 0 otherwise. This required converting all clues to probabilities, for which we used the logistic function $1/[1 + \exp(-c * (x - k))]$ with two parameters k and c set by experiments. We obtained a best F-score of 0.6435 and a best threshold of 0.3807 with ten-fold cross-validation, similar to what we got with Naïve Bayes. The best weights were 0.1678 on length in characters (with best k = 6 and c = 6), 0.0245 for capitalization, −0.0380 for dictionary count (with best k = 5000 and c = 5000), −0.2489 for adjusted number of drives (with best k = 2000 and c = 2000), −0.0728 for rate per drive (with best k = 10 and c = 10), −0.0234 for number of words, 0.1383 for lack of explicit non-name usage, 0.1691 for having a following digit, 0.3823 for lack of having a nearby uninteresting site name, 0.0285 for number of delimiters, with $w_0 = -0.1695$.

We also tested a case-based reasoning model with the numeric clues, using the training set as the case library. We took the majority vote of all cases within a multiplier of the distance to the closest case. With ten-fold cross-validation, we got an average maximum F-score of 0.6383 with an average best multiplier of 1.97, but it took considerable time. We also tested a set-covering method and got an F-score of 0.60 from training alone, so we did not pursue it further. Thus Bayesian methods were the best, but it appears that the choice among the first three conceptual models does not affect performance much.

5 Cross-Modal Clues

Important clues not yet mentioned for personal names are the ratings on a nearby recognized artifact of a different type such as email addresses and phone numbers. For instance, "John Smith" is a common personal name, but if we find it just before "jsmith@officesolutions.com" we should decrease its rating because the address sounds like a vendor contact, and people usually separate their business mail and personal mail. Similarly, if we see the common computer term "Main" is preceded by interesting address "bjmain@gmail.com", we should increase its rating since Gmail is primarily a personal-mail site. These can be termed cross-modal clues. Our previous work [14] rated email addresses on our corpus, so those ratings can be exploited.

5.1 Rating Phone Numbers

Other useful cross-modal clues are nearby phone numbers, and their restricted syntax makes them easy to identify. Bulk Extractor finds phone numbers, but it misidentified some numeric patterns like IP numbers as phone numbers, and erred about 5% of the time in identifying the scope of numbers, most often in missing digits in international numbers. Code was written to ignore the former and correct the latter by inspecting the adjacent characters. For example, “123-4567” preceded by “joe 34-” is modified to “+34-123-4567” and “12345-” followed by “6789 tom smith” is modified to “12345-6789”. Since the country of origin for each drive was known, its code was compared to the front of each phone number and the missing hyphen inserted if it matched. Some remaining 889,158 candidates proffered by Bulk Extractor were excluded because of inappropriate numbers of digits and invalid country codes. The code also regularized the format of numbers to enable recognition of different ways of writing the same number. U.S. numbers were converted to the form of ###-###-#### and international numbers to +##-######## and similar variants. Some U.S. numbers were missing area codes, and “?” was used for the missing digits.

The main challenge was in identifying the forensically interesting phone numbers, those that were personal and not of businesses or organizations, since the numbers themselves provide few clues. We evaluated the following clues for a Bayesian model:

- Whether the area code indicated a business or informational purpose as publicly announced (e.g., 800 numbers for businesses in the United States).
- Whether the number appeared to be artificial (e.g. 123-4567).
- Whether the number was in the United States.
- The number of drives on which the phone number occurred.
- Whether the number occurred on only one drive and at least four times, which suggests a localized number.
- Whether the number was preceded by “phone” or something equivalent.
- Whether the number was preceded by “fax” or something equivalent.
- Whether the number was followed by “fax” or something equivalent.
- Whether the number was preceded by “cell” or something mobile-related.
- Whether the last character preceding the number was a digit (usually an indicator of a scope error).
- Whether any of the words in the preceding 16 characters could be names.

Table 5 shows the calculated odds for each of the clues using 100 runs on random partitions of a training set of 4105 tagged random selections from our corpus, 3507 uninteresting and 446 interesting. Each of the 100 runs chose 50% of the training set for training and 50% for testing (an even split because we had little training data with positive examples). We then averaged the resulting odds over the runs. Either the clue or its absence was statistically significant, so all clues are justified to be included the model. For these tests the average best F-score with the model using all clues was 0.403 with an average best threshold of 0.0214, so most phone numbers are uninteresting and thus negative clues to nearby personal names, but there are not many. The output is also useful for rating phone numbers.

Table 5. Odds of interesting phone numbers based on particular clues.

Clue	Odds on training set	Standard deviation of odds
US	0.204	0.009
Non-US	0.121	0.045
Informational area code	0.008	0.004
Not informational area code	0.231	0.010
Artificial	0.018	0.004
Not artificial	0.204	0.009
Occurred on only one drive in corpus	0.246	0.016
Occurred on 2–4 drives in corpus	0.405	0.029
Occurred on 5 or more drives in corpus	0.012	0.004
Whether it occurred on only one drive and at least 4 times	0.378	0.061
Whether it occurred on multiple drives or less than 4 times	0.194	0.009
Personal name preceding	0.393	0.045
No personal name preceding	0.186	0.009
Preceded by “phone” or similar words	0.203	0.009
Preceded by “fax’ or similar words	0.138	0.022
Followed by “fax” or similar words	0.203	0.009
Preceded by mobile-related words	0.630	0.242
No useful preceding or following words	0.209	0.010
Preceded by a digit after all possible corrections	0.136	0.011
No preceding digit after all possible corrections	0.238	0.012
Prior to any clues	0.203	0.009

5.2 Combining Cross-Modal Clues

We explored three cross-modal clues to personal names: the rating on nearby email addresses with words in common, the rating on closely nearby email addresses, and the rating on closely nearby phone numbers. Preliminary experiments showed that personal name ratings only correlated over the entire corpus with email ratings within a gap of 10 or less bytes or if they had at least half their words in common; personal name ratings only correlated with phone numbers within 20 or less bytes. So we used those results to define “closely nearby”. There were 708 instances of email addresses with common words within 50 bytes, 690 instances of email addresses within 10 bytes, and 21 instances of phone numbers within 20 bytes.

Since the ratings were widely varying probabilities, for these cross-modal candidates we fit a linear rather than Bayesian model of the form $t = w_0 + w_n r_n + w_{ew} r_{ew} +$

$w_{eo}r_{eo} + w_{po}r_{po}$. Here t was 1 for valid personal names and 0 otherwise, r_n is the rating on the personal name, r_{ew} is the rating on the nearby email address sharing words, r_{eo} is the rating on the closely nearby email address, and r_{po} is the rating on closely nearby phone number. The w values were the weights on the corresponding ratings, and c_{ew}, c_{eo}, and c_{po} were default constants for r_{ew}, r_{eo}, and r_{po} when there was no nearby cross-modal clue. Evidence from more than one candidate could be used for a single personal name. We computed the least-squares fit of the linear model for the four weights and three constants applied to the training set. This gave a model of $t = -0.328 + 0.590r_n + 0.157r_{ew} + 0.005r_{eo} + 0.006r_{po}$ with $c_{ew} = 0.300$, $c_{eo} = -0.121$, and $c_{po} = 0.476$. Using these values we achieved a best F-score of 0.7990 at a threshold of 0.2889, a 19% improvement over rating without cross-modal clues. Using this model we could now achieve 90% recall with 69.5% precision and 100% recall with 66.5% precision, albeit in testing only on the subset of the training set that had evidence for at least one of the cross-modal clues. We also tested clues from the ratings on other nearby personal names, but found their inclusion hurt performance, reducing F-score to 0.7576.

6 Identifying the Principals Associated with a Drive

A secondary use of name extraction from a drive is quick identification of the main people associated with a drive, something important for instance when drives are obtained in raids apart from their owners. The most common names on a drive are not necessarily those of the owner and associates since names of vendor contacts and common words that can be used as names occur frequently. User-directory names (e.g. in the "Users" directory in Windows) can be misleading because they can be aliases, they only show people who log in, and do not give frequencies of use.

A better criterion for the owner and associates that we found is the highest-count personal names with a rating above a threshold, where the rating is computed by the methods of Sect. 5. We applied this this to 12 drives we obtained from co-workers, the only drives for which we could confirm the owner. For 8 of those 11, the owner name was the top-rated name over a 0.2 rating, for one it was second, for one it was fourth, and for one it was twentieth (for apparently a drive used by many people). So the rating threshold criterion appears to be reliable. For instance for an author's old drive, the first name rated above 0.2 was the author's first initial and last name, though it was the fifth most common name on the drive, and the second rated above 0.2 was the author's wife's name, even though it was the tenth most common name on the drive.

7 Conclusions

Personal names are among the most valuable artifacts an investigator can find on a drive as they can indicate important personal relationships not otherwise made public. This paper has shown that 71.3% of name candidates near email addresses and phone numbers can be eliminated from consideration from a representative corpus with an estimated average F-score of 67.4%. With cross-modal clues, F-score can be improved to 79.9%. Since our assumptions and methods apply to nearly any criminal or intelligence application of forensics, our methods permit a 3.5 times reduction in the workload of such investigators looking for personal names on drives who need no longer examine everything that matches a dictionary of names. At the same time, our ability to bootstrap on existing output of Bulk Extractor means our methods require only an additional few minutes per drive, far better than the days needed to do keyword search for names on a typical drive image (see Sect. 2). The work could be extended by developing a more specialized and efficient Bulk Extractor plugin; exploiting street addresses, IP addresses, names of associated organizations, and file names as additional cross-modal clues; and testing differences in strategy for different types of drives.

Acknowledgements. This work was supported in part by the U.S. Navy under the Naval Research Program and is covered by an IRB protocol. The views expressed are those of the author and do not represent the U.S. Government. Daniel Gomez started the implementation, and Janina Green provided images of project-team drives.

References

1. Bikel, D., Miller, S., Schwartz, R., Weischedel, R.: Nymble: a high-performance learning name-finder. In: 5th Conference on Applied Natural Language Processing, Washington DC, US, March, pp. 194–201 (1997)
2. Bulk Extractor 1.5: Digital Corpora: Bulk Extractor [Software] (2013). http://digitalcorpora.org/downloads/bulk_extractor. Accessed 6 Feb 2015
3. Fan, X., Wang, J., Pu, X., Zhou, L., Bing, L.: On graph-based name disambiguation. ACM J. Data Inf. Qual. **2**(2), Article No. 10 (2011)
4. Garfinkel, S.: Forensic feature extraction and cross-drive analysis. Digit. Invest. **3S** (September), S71–S81 (2006)
5. Garfinkel, S.: The prevalence of encoded digital trace evidence in the nonfile space of computer media. J. Forensic Sci. **59**(5), 1386–1393 (2014)
6. Garfinkel, S., Farrell, P., Roussev, V., Dinolt, G.: Bringing science to digital forensics with standardized forensic corpora. Digit. Invest. **6**(August), S2–S11 (2009)
7. Gross, B., Churchill, E.: Addressing constraints: multiple usernames, task spillage, and notions of identity. In: Conference on Human Factors in Computing Systems, San Jose, CA, US, April–May, pp. 2393–2398 (2007)
8. Henseler, H., Hofste, J., van Keulen, M.: Digital-forensics based pattern recognition for discovering identities in electronic evidence. In: European Conference on Intelligence and Security Informatics, August (2013)
9. Lee, S., Shishibori, M., Ando, K.: E-mail clustering based on profile and multi-attribute values. In: Sixth International Conference on Language Processing and Web Information Technology, Luoyang, China, August, pp. 3–8 (2007)
10. McCalley, H., Wardman, B., Warner, G.: Analysis of back-doored phishing kits. In: Peterson, G., Shenoi, S. (eds.) DigitalForensics 2011. IAICT, vol. 361, pp. 155–168. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-24212-0_12
11. Paglierani, J., Mabey, M., Ahn, G.-J.: Towards comprehensive and collaborative forensics on email evidence. In: 9th IEEE Conference on Collaborative Computing: Networking, Applications, and Worksharing, pp. 11–20 (2013)

12. Petkova, D., Croft, W.: Proximity-based document representation for named entity retrieval. In: 16th ACM Conference on Information and Knowledge Management, Lisbon, PT, November, pp. 731–740 (2007)
13. Rowe, N., Schwamm, R., Garfinkel, S.: Language translation for file paths. Digital Invest. **10S**(August), S78–S86 (2016)
14. Rowe, N., Schwamm, R., McCarrin, M., Gera, R.: Making sense of email addresses on drives. J. Digit. Forensics Secur. Law **11**(2), 153–173 (2016)
15. Yang, M., Chow, K.-P.: An information extraction framework for digital forensic investigations. In: Peterson, G., Shenoi, S. (eds.) DigitalForensics 2015. IAICT, vol. 462, pp. 61–76. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24123-4_4