Combining Entity Matching Techniques for Detecting Extremist Behavior on Discussion Boards

Johan Dahlin
Division of Automatic Control, Linköpings University
SE 581 83 Linköping, Sweden
Email: johan.dahlin@isy.liu.se

Fredrik Johansson, Lisa Kaati,
Christian Mårtenson, Pontus Svenson
FOI
SE 164 90 Stockholm, Sweden
Email: firstname.lastname@foi.se

***Abstract*—Many extremist groups and terrorists use the Web for various purposes such as exchanging and reinforcing their beliefs, making monitoring and analysis of discussion boards an important task for intelligence analysts in order to detect individuals that might pose a threat towards society. In this work we focus on how to automatically analyze discussion boards in an effective manner. More specifically, we propose a method for fusing several alias (entity) matching techniques, that can be used to identify authors with multiple aliases. This is one part of a larger system, where the aim is to provide the analyst with a list of potential extremist worth investigating further.**

I. Introduction

One of the many tasks intelligence analysts are facing is to search for and identify violent extremists in order to prevent them from causing harms to the society. Traditional methods such as infiltration of the extremist networks are often used, but it is also well known that extremist groups and terrorists use the Internet for various purposes, and that some radical web pages on the Internet are facilitators for violent extremism. Moreover, the ability to stay connected to extremist ideology can promote radicalization. For many extremist groups the Internet is an important initiator for their ideology and serves as a natural meeting place to share knowledge and thoughts [1], which increases the risk of committing violent actions.

For the above reasons, the focus of our research is to develop tools and techniques for supporting analysts with the detection of weak signals of possible violent extremism on Internet. A weak signal is something that by itself does not necessarily mean anything, but typically becomes stronger when combined with other signals. As an example, information that someone has bought a gun does not indicate anything special if no other information is available, but if it is combined with information saying that the same individual has bought large quantities of ammonium nitrate and made radical postings in extremist discussion boards, this individual may need closer attention by the police or intelligence service. Other examples of weak signals can be changes in public attitudes regarding some political question or an increasing trend in using certain explosives in terror attacks.

One of the major problems with analyzing information from the Internet is that it is vast, making it impossible for analysts to manually find, read, and analyze all relevant information.We have therefore recently presented a framework for using web harvesting and natural language processing techniques in order to semi-automatically detect web sites of interest, and to analyze their content in order to be able to rank the radicality level of users writing on extremist forums (based on the discussions they take part in and the content of their postings). The ranking of the authors is intended to be used as a guidance and prioritization tool for analysts in their work when analyzing discussion boards to detect signs of violent extremism.

One important part of this process is to do entity matching, i.e., to discover if an author is using several aliases, and to merge the aliases if that is needed. This can, e.g., be the case if an individual is a member of several discussion boards, or if he or she uses several aliases on the same forum. There are several potential reasons for an individual to use multiple aliases on a single discussion board. It could be the case that the first alias becomes banned from the discussion board, or that the author simply forgot the password. It could also be the case that an alias has lost the others trust in the discussions, or that the author has developed bad personal relationships with certain members of the discussion board. Another potential reason is that the author creates multiple aliases that writes messages that supports his or her own arguments.

No matter what the reason is for having multiple aliases, the fact that many people use several aliases is troublesome since it may make it harder to fuse several weak signals generated by an individual. Therefore, in this paper, we are presenting various entity matching techniques, based on 1) field matching of alias names, 2) text analysis, 3) graph (network) analysis, and 4) spatio-temporal matching, as well as a method for combining these methods in order to match aliases better.

The rest of this paper is outlined as follows. In Section II we present our framework for detection of violent extremists, of which the alias matching presented is one of the key components. Then we focus on entity (alias) matching and present several approaches for such matching in Section III. In Section IV, we present a novel method for combining all of the previously presented entity matching algorithms into a single classifier, judging which aliases should be treated as belonging to the same individual. Integrity aspects are discussed in Section V, and conclusions and future work are presented in Section VI.

II. System overview

The work presented in this paper is part of a framework containing processes and tools for supporting an analyst to detect signs of violent extremism on the web [2]. The main process, depicted in Figure 1, starts with identifying and modeling the type of phenomenon that is of interest to detect. The modeling is based on a problem breakdown approach, where an initial hypothesis (e.g. ”Actor X is a potential lone wolf terrorist”) is broken down into sub-hypotheses (e.g. that Actor X has the ”Intent”, ”Capability” and ”Opportunity” to commit an act of terror). The sub-hypotheses are further decomposed until they are detailed enough so that concrete actions can be taken to determine their value (e.g. ”Actor X has expressed intent to commit violent actions through radical postings in Forum A”). These sub-hypotheses are called indicators and correspond to the weak signals we want to collect and fuse in order to detect and prevent potentially illicit acts.

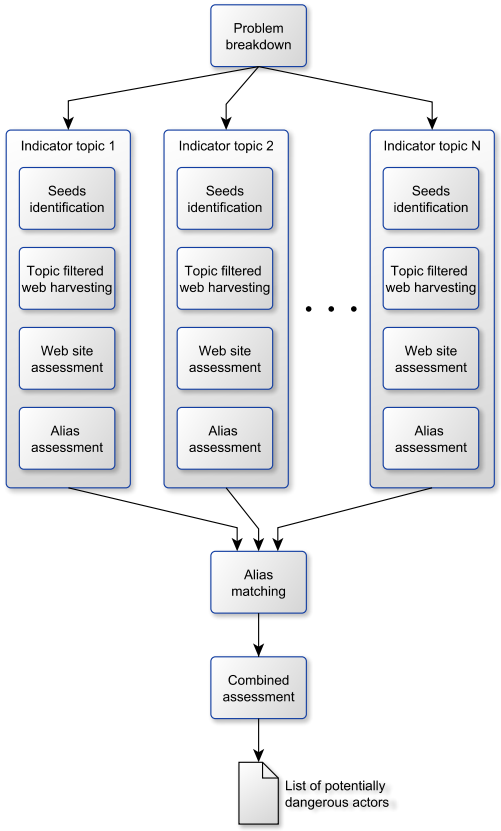

Fig. 1. The overall process using a problem breakdown approach and indicator topics.

Weak signals in general can consist of almost anything, but in this paper we focus on detecting weak signals on the web and in particular on discussion boards. To collect the information necessary to assess the indicators we suggest to group indicators by topic (e.g. all indicators dealing with right-wing extremism in one group, and all indicators concerning bomb manufacturing in another) and then perform topic filtered web harvesting to sort out relevant sites and forums. This basically means using a web crawler, starting from a number of seeds consisting of known sites of interest, and only following links which lead to other topic relevant sites, determined through automatic content analysis.

Once a list of interesting web sites or forums have been created, a deeper analysis of postings on these is performed using natural language processing and text mining. The goal is to automatically extract aliases and to estimate the indicator values for each one of them in order to output a ranked list of aliases. As noted in the introduction, the same author can use multiple aliases. Consequently, there is a risk that weak signals stemming from the same actor will end up in separate models and not be assessed together. As this might cause relevant actors to be overlooked, matching and fusing aliases of the same author is an important step to make the process of detecting weak signals robust.

III. Matching Multiple Aliases

In the digital society it is increasingly common for users to interact on discussion boards using a number of different aliases. There are several potential reasons for using multiple aliases, as already has been mentioned. To get accurate information about the authors that are active on extremism discussion boards, we need to detect if some of them use multiple aliases, and if that is the case, merge these as well as the messages from the different aliases. For finding users that make use of multiple aliases, entity matching techniques come in handy. Most entity matching techniques focus on matching of alias names that can be useful for cases where the user is not deliberately choosing to hide that he or she is using several aliases. Examples of this situation is when a user has similar (or identical) alias names on several web sites of interest, or if a user has forgot the old password and therefore creates a new alias with a similar name. This kind of entity matching techniques are however not working when a user deliberately is choosing aliases with dissimilar names. For such situations, various stylometric techniques in which the authors’ writing style is analyzed can be more fruitful. Also graph-based techniques can be of value for entity matching purposes. In the following, we describe some of the techniques that can be used and combined to detect authors that uses different aliases. It is worth noticing that entity or alias matching is closely related to the problem of alias disambiguation [3], in which one tries to find out if a certain alias name refers to one and the same, or multiple entities. Hence, much of the methods and insights presented here can be used for alias disambiguation as well.

A. Matching of alias names (field matching)

The first step towards creating a method for matching similar aliases is to quantify the similarity or difference between them. Many well-known measures already exist for quantifying similarities in integers, floats and sets. These

include the usual norm measures (Manhattan and Euclidean distances), set measures (the Jaccard and cosine measures), and others. Aliases, however, consist of text strings, which are non-trivial to compare using standard methods. It is possible to use set measures to compare the similarity between strings, by treating each character as a member in a set comprised by the text string. However, these type of methods heavily penalize misspellings of words and are therefore not suitable for our problem.

To counter the problems of using traditional measures for the studies of sets, some other methods have been developed during the last few decades. These methods are special cases of the so-called *field matching techniques*, which are methods used to match individual fields in a data set to each other. Field matching methods can be divided into character-based, token-based, and phonetic similarity metrics. We continue by reviewing the most promising methods in the two former metric types. We disregard phonetic methods in this paper, as these depend heavily on the language considered, i.e. methods developed for e.g. American English does not necessarily perform well for the Nordic languages. Therefore this method is considered impractical in comparing names in a more globalized world in which cultures and people (thereby also names) tend to increasingly mix.

The first type of similarity measures are comparing the characters used in two different strings. We present the two most commonly used metrics in this subsection: edit distances and the Jaro-Winkler metric.

1) Edit distance: Perhaps the simplest of character-based methods is the *Levenshtein edit distance* presented in [4]. This metric is related to the Hamming distance used in information theory. This measure compares two strings by the number of edits needed to transform one string into the other. Denote the two strings s_1 and s_2, then the edit distance is the minimum amount of operations needed to transform s_1 into s_2 or vice versa. The allowed *edit operations* are:

- *insertion* of a character into a string,
- *removal* of a character in a string,
- *replacement* of a character with a different character.

To calculate the edit distance, a dynamic programming problem is usually solved with a complexity of $\mathcal{O}(|s_1||s_2|)$. However, there exist more elaborate versions of the algorithm which limit the complexity to $\mathcal{O}(k|s_1|)$ or similarly $\mathcal{O}(k|s_2|)$ for some k if the edit distance between s_1 and s_2 is less than k.

Other refinements to the edit distance includes gaps, which allows for a smaller number of operations in strings with left-out words. For example, using gaps a single operation is needed to transform *Sven Anders Svensson* into *Sven Svensson*. This is done by just removing the additional word (the gap) using a single operation, i.e. removing or adding Anders to transform the strings. The cost of this operation can vary depending on the location of the gap, usually higher costs are given to gaps in the beginning and end of strings than in the middle.

2) Jaro-Winkler metric: Another common similarity measure is known as the *Jaro-Winkler metric* presented in [5], which is used for comparing short strings such as names. We begin by discussing the simpler *Jaro metric* due to [6], which is calculated as follows:

1) Find the length of each string, $n_1 = |s_1|$ and $n_2 = |s_2|$.
2) Find the number of common characters c shared between the two strings. A common character fulfills the following:

$$s_1[i] = s_2[j], \qquad |i - j| \leq \frac{1}{2}\min\{n_1, n_2\}. \tag{1}$$

3) Find the number of possible transpositions, t, which is the number of common characters for which $s_1[i] \neq s_2[i]$ where $i = 1, 2, \ldots, c$.
4) The Jaro metric, $\mathsf{J}(s_1, s_2)$, is given by

$$\mathsf{J}(s_1, s_2) = \frac{1}{2}\left(1 + \left[\frac{n_1 + n_2}{n_1 n_2}\right] c - \frac{t}{2c}\right). \tag{2}$$

The complexity of this algorithm is $\mathcal{O}(n_1 n_2)$ and is due to the calculation of the number of common characters. A common extension of the Jaro metric is the Jaro-Winkler metric due to [5], which gives a higher weight to prefix matches by the following

$$\mathsf{JW}(s_1, s_2) = \mathsf{J}(s_1, s_2) + p\max\{l, 4\}\left(1 - \mathsf{J}(s_1, s_2)\right), \tag{3}$$

where $p \in [0, 0.25]$ is a factor[1] controlling how the score is increased for having common prefixes, l is the length of the longest common prefix of s_1 and s_2. As previously stated, this method is well suited for comparing names of all types and does not have the drawback of the phonetic family of measures, which is strongly language dependent.

3) Comparison: The different character-based metrics are best understood by some comparative examples. In *Table I*, some simple examples of misspellings and different formatted strings are compared using the edit distance, Jaro, and Jaro-Winkler metrics. The three metrics find a large similarity between two pairs of strings "Sven Svensson/Sven Svenson" and "Svensson, Sven/Svensson, S". The metrics does not however find any larger similarity between the other three examples in the table.

The more advanced metric, Jaro-Winkler finds larger similarity between the three first comparisons in the table. This is as expected as Jaro-Winkler is specially design to weight the first letters in a name higher than the ending of names, i.e. this metric handles initial vs. full first name well. The Jaro and Jaro-Winkler metrics are also designed for matching names and does not perform well with other types of strings, such as organizational names. For these types of text strings, token-based metrics are needed, which are not further discussed in this paper since it is not directly related to alias matching.

[1]A common choice for this factor is $p = 0.1$ and this value is used in the following example.

String 1	String 2	L	J	JW
Sven Svensson	Sven Svenson	1	0.974	0.985
Svensson, Sven	S Svensson	8	0.790	0.811
Svensson, Sven	Svensson, S	3	0.923	0.957
Division of Information Systems	Information Systems Division	21	0.710	0.710
Division of Information Systems	Information Systems	12	0.713	0.713

TABLE I
SMALL COMPARISON BETWEEN SOME TEXT STRINGS USING THE LEVENSTHEIN (EDIT DISTANCE), JARO, AND JARO-WINKLER METRICS.

B. Text-based (stylometric) matching methods

While it is easy to choose alias names that are dissimilar if one would like to avoid that others detect that one is using several aliases, it is harder to avoid using the writing style one is used to. Hence, it can be useful to study the specific text characteristics used by various aliases in order to match those using similar writing styles. This kind of statistical analysis of writing style is known as stylometry [7]. According to [8], relatively much research has been devoted to the use of stylometric analysis techniques for online author identification (i.e. the situation where one would like to determine the author of a text from a set of possible authors), while considerably less emphasis has been given the case of similarity detection, in which no possible authors are known a priori and the task instead is to compare anonymous texts against other anonymous texts and to assess the degree of similarity in an unsupervised fashion.

Examples of stylistic features (characteristics) that can be used for author identification and similarity detection are: choice of words, language, syntactic features, syntactical patterns, choice of subject, and different combinations of these [9]. The short texts that are typical for the online case makes writing style analysis extra difficult, but it is noted in [10] that Internet-specific characteristics such as smileys can give better results when recognizing authors from short texts.

In [11], multiple aliases that belong to an author are identified using content analysis on text posted in public fora such as bulletin boards, weblogs, and web pages. In their work, the users vocabulary is considered and the feature set representing the corpus of a particular alias they take into consideration are: choice of words, misspellings (words that are not present in a large dictionary), punctuation, emotion symbols and function words (frequently used words with no content whose occurrence frequency does not change much from domain to domain such as: and, but, this, very and which). Aliases from discussion boards are in that work clustered into equivalence classes, where the aliases in a class have a high probability of belonging to the same individual. The clustering is based on a characterization of the content of the messages written by an alias. There are of course various unsupervised methods that can be used for the clustering, but the basic foundation is the same for most of them.

Another approach to alias matching is to use text analysis methods to identify texts by different aliases that discuss the same topics and uses the same type of words. This could indicate that the texts are written by the same entity and that therefore these two aliases are in fact the same.

Text mining is usually applied to find documents with similar context or for finding semantic expressions (grouping words with similar meaning). The most commonly used method is often referred to as *Latent Semantic Analysis* (LSA) and follows a series of three steps: (i) the creation of a weighted Term-Document (TD) matrix, (ii) calculating a truncated singular value decomposition, (iii) calculating similarity and clustering similar documents. These different steps are now discussed in detail.

The grouping of similar documents is done using the partitional clustering algorithm called k-means with the *cosine similarity* between documents

$$\mathsf{sim}(\mathbf{D}_i, \mathbf{D}_j) = \frac{|\mathbf{x} \cap \mathbf{y}|}{\sqrt{|\mathbf{x}||\mathbf{y}|}}, \tag{4}$$

where $\mathbf{D}_i$ and $\mathbf{D}_j$ are some vectors describing the content of two (different) documents i and j. These vectors are found by a truncated *Singular Value Decomposition* (SVD) of the weighted TD-matrix. The TD-matrix, $\mathbf{X} = [X_{ij}]$, is found by calculating the frequency of words occurring in each document, the rows denote the different terms used in all documents in the collection of documents (the corpus) and the columns denote the different documents. Each element in the TD-matrix, X_{ij}, describes the frequency with which the word i occurs in the document j.

The TD-matrix is weighted using the well-known TF-IDF measure. The aim with this measure is to give uncommon words more weight than more commonly found words[2]. The weighted TD-matrix is decomposed using the following relation

$$\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top, \tag{5}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices with *singular vectors* and Σ is a diagonal matrix with the *singular values*. To truncate this decomposition, an appropriate number of singular values needs to be determined. This is usually done by some rule-of-thumb, e.g. the Kaiser criteria (stating that all values larger than unity should be included). Another method usually applied in Principal Component Analysis (PCA) is to the find the *elbow point* in the *scree plot*. The latter is a plot of the

[2]It is worth noting that *stop words* often are removed before constructing the term-document matrix. These stop words include common prepositions and other functional words, e.g. the, is, and, which, and that, which carries no specific meaning.

decreasingly sorted singular values and the former is found as the point after which the scree line tends to level out, see the graph in *Figure 2*.

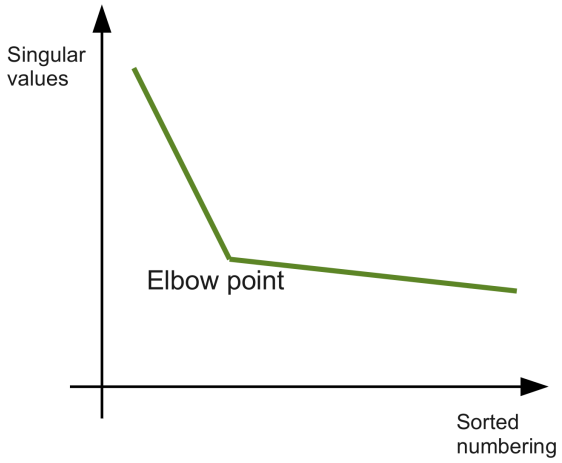

Fig. 2. A scree plot with the singular values sorted in decreasing magnitude. The elbow point is found as the point after which the line tends to level out, as indicated by the label in the graph.

The truncated expansion is found by choosing some appropriate number of singular values, k, as the following

$$\hat{\mathbf{X}}_k = \sum_{i=1}^{k} \mathbf{U}_{ik} \Sigma_{ii} \mathbf{V}_{ki}^{\top}, \tag{6}$$

where the columns of $\mathbf{X}$ are the new *coordinates* describing each document. These column vectors are used in the cosine measure as the vectors describing documents. The advantage of this method is that the truncated SVD reduces the number of words (by creating groups of words with similar meaning as new basis vectors) and by reducing the amount of noise (infrequently occurring words). This makes it easier to find documents (abstracts) with similar content.

The cosine similarity is computed for each pair of documents and placed in a similarity matrix, $\mathbf{S} = [S_{ij}]$, where the element $S_{ij} = \mathsf{sim}(\hat{\mathbf{X}}_{k,i}, \hat{\mathbf{X}}_{k,j})$ denotes the cosine similarity between columns i and j in the truncated SVD computed above. This matrix describes the similarity between documents and is used together with the k-means algorithm to cluster similar documents into groups. The k-means algorithm is used to generate k clusters of documents and the result is a label for each document c_i, where $i = 1, 2, \ldots, N$ and N is the number of documents, describing to which cluster each document belong.

C. Graph-based entity matching

Another approach to alias matching is to use graph-based methods. An example of this is to use social network analysis (SNA) [12], [13] to analyze the relationship between different data entries. If two aliases post to the same forums, on the same topics, and regularly comment on the same type of posts, it is likely that they are in fact the same. It is also possible to use abstraction techniques such as simulation [14] to determine the likelihood with which two aliases are the same. In [3] the social network in which aliases reside is studied. The social network is constructed from email data mined from the Internet.

An approach for identifying similar entities in networked data (such as link or author networks) is to use the network structure itself. The idea for this originates from the fields of complex networks and sociology, where *vertex similarity* and *structural equivalence* have been studied. The main underlying thought is that nodes are similar if they share a large fraction of neighbors, e.g. share many friends in a social network or share many coauthors in citation networks. An example of this is shown in *Figure 3*, where two nodes corresponding to authors in a citation network are shown with their neighbors. As these nodes share a large fraction of neighbors and have similar names, it is possible that they correspond to the same real-world person.

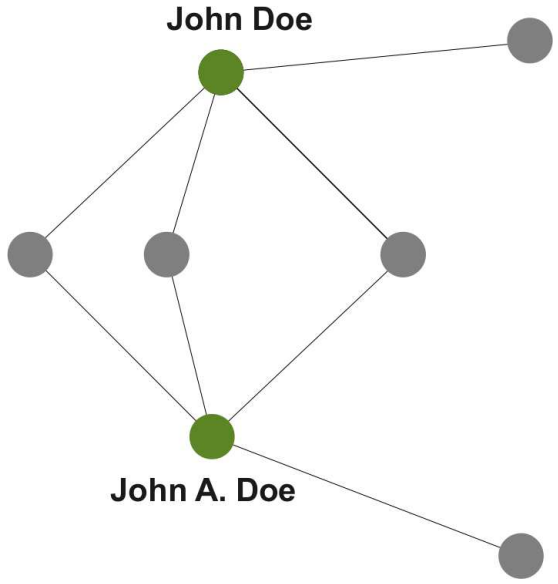

Fig. 3. Two nodes and their neighbors (grey nodes) corresponding to a sub-network of some data on aliases and forum posts.

We follow [15] by discussing some common simple metrics for determining vertex similarity. The simplest possible measure of the similarity of two nodes in the graph-based setting is the number of shared neighbors found as

$$\sigma(v_i, v_j) = |\Gamma_i \cap \Gamma_j|, \tag{7}$$

where Γ_i denotes the set of neighbors of node v_i. The drawback of this measure is that nodes with a high degree (many neighbors) will have a larger similarity than nodes with smaller degree. To obtain a comparable measure, it is necessary to normalize the similarity by the node degrees. Some common similarity measures, $\sigma(v_i, v_j)$, are

$$\textbf{Jaccard}: \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i \cup \Gamma_j|}, \quad \textbf{Cosine}: \frac{|\Gamma_i \cap \Gamma_j|}{\sqrt{|\Gamma_i||\Gamma_j|}}, \tag{8}$$
$$\textbf{Vertex}: \frac{|\Gamma_i \cap \Gamma_j|}{|\Gamma_i||\Gamma_j|}, \textbf{Dice}: \frac{2|\Gamma_i \cap \Gamma_j|}{|\Gamma_i| + |\Gamma_j|},$$

$$\textbf{Inv. log} - \textbf{weighted} : \sum_{x \in \Gamma_{ij}} \frac{1}{\log |\Gamma_x|},$$

where $x \in \Gamma_{ij}$ is common neighbors to nodes i and j, i.e. $\Gamma_{ij} = \Gamma_i \cap \Gamma_j$ in the inverse log-weighted similarity that is due to [16]. Similar nodes in graphs will have high values of similarity, which may indicate that two nodes are not distinct. By using some predetermined limit value, l, classification is achieved using some linkage function such that

$$f(x) = \begin{cases} -1 & \textbf{if} x = \sigma(v_i, v_j) \leq l \\ 1 & \textbf{if} x = \sigma(v_i, v_j) > l \end{cases}, \tag{9}$$

i.e. the two nodes match if $f(x) = 1$ and unmatch if $f(x) = -1$ for some similarity x.

D. Spatio-Temporal Entity Matching

Yet another component for detecting the use of multiple aliases is to consider the points in time when different aliases post messages to the forums. If two aliases post messages during the same hours of the day, correlation of posting times can be used as a factor to increase the likelihood that the author behind the aliases is actually the same. This is obviously not a factor that can be used as hard evidence of two aliases having the same author, but can in combination with other methods such as text-based approaches be used as an extra indicator. To the best of our knowledge, no previous attempts exist where one takes the time of posting into consideration when doing alias matching.

IV. Fusion

In order to determine whether two aliases really are the same, as many as possible of the methods described earlier must be used. Exactly which methods that can be used in each specific case will depend on what data is available. It is highly unlikely in any realistic application that a single method will provide a conclusive answer by itself. Assume that N different methods were used to compute the likelihood that aliases x and y are the same and denote the result of applying method i as $C_i(x, y)$. We must now fuse these N classification scores to determine whether we should in fact consider x and y to be identical.

Figure 4 shows a conceptual overview of how fusion of the different methods can be done (only a selection of the presented methods are shown for illustration purposes, but obviously all of them can be combined). In the figure, the box labeled "Classification voting" represents the fusion rule. There are a wide variety of different fusion rules that could be used. The simplest is to simply take a majority vote of the classifiers used, and to consider x and y equal if a majority of the C_i say so. This very simplistic method has for some fusion applications actually been shown to outperform other, more sophisticated approaches [17]. Simple variations of this include weighting the classifiers (with, e.g., their confidence).

A more advanced and potentially better approach is to fuse using Dempster's rule of combination [18]. Dempster-Shafer theory allows us to model uncertainty about the classification

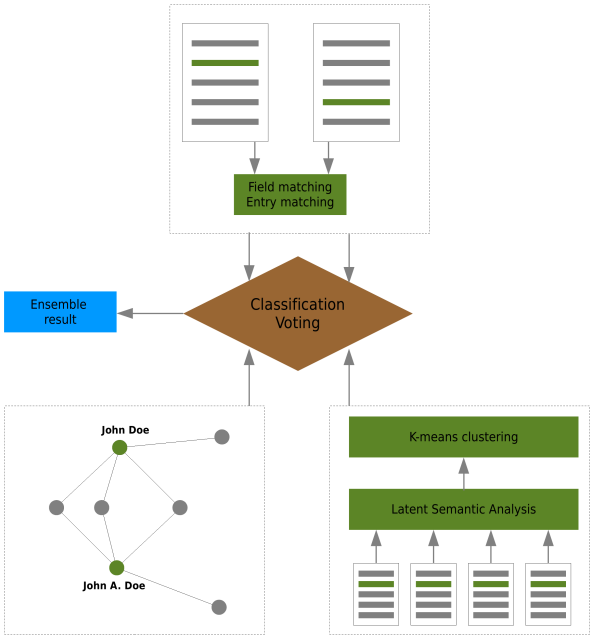

Fig. 4. Fusion of the alias matching approaches

in a better way than if we just used standard probability. In Dempster-Shafer theory, the basic object of study is the mass function (or basic belief assignment), m , which is a function from the subsets 2^{Θ} of some set Θ to the interval $[0, 1]$ such that

$$m(\emptyset) = 0$$

and

$$\sum_{x \subseteq \Theta} m(x) = 1.$$

The mass function looks superficially like a probability function, but is interpreted differently. $m(t)$ for any subset $t \subseteq \Theta$ is the belief we have that the true answer is in t but not in any proper subset of t. It can be seen as a random set representing an oracle that allows us to query it regarding the true answer. In Dempster-Shafer theory, we also define the belief and plausibility of a subset $t \subseteq \Theta$ as

$$B(t) = \sum_{u \subseteq t} m(u) \tag{10}$$

and

$$P(t) = \sum_{u \cap t \neq \emptyset} m(u).$$

$B(t)$ can be interpreted as all the belief that the true answer is contained in t (including its subsets), while $P(t)$ represents the plausibility of t, or equivalently the belief that can, if we combine with new evidence, later support t.

Note the subtle but important difference between m and B: the former represents the evidence that points to t and cannot be further specified for the subsets of t, while the latter tells us the total belief the true answer is any of the elements of t.

Dempster-Shafer theory is most useful when we have information that indicates the true answer, but where we cannot assume that the absence of evidence for an answer A indicates that A is not the answer. For the alias matching case, this is a perfect match: each individual classifier C_i gives us some indication whether or not x and y are the same, but can't really be used to say anything about whether they are not the same.

We model this using N mass functions m_i specified on a domain $\Theta = \{\mathbf{Same}, \neg\mathbf{Same}\}$ and assume that each classifier gives us a certainty $m_i(\mathbf{Same})$ and that $m_i(\neg\mathbf{Same}) = 0$ (i.e., there is never any evidence that two aliases are not the same[3]).

Fusion of mass functions is then done using Dempster's rule of combination:

$$m_{i,j}(t) = \frac{1}{1-K} \sum_{u \cap v = t} m_i(u) m_j(v), \tag{11}$$

where

$$K = \sum_{u \cap v = \emptyset} m_i(u) m_j(v)$$

is a normalization constant. Fusion of more than two mass functions is done iteratively; it is also possible to compute a similar formula for the direct combination of an arbitrary number of mass functions. After fusing all N mass functions, we can then compute the total belief that x and y represent the same individual by computing $B(\mathbf{Same})$ using equation (10).

Consider an example where we wish to determine the belief that two aliases represent the same person given the following evidences:

$$C_1 : m_1(\mathbf{Same},) = 0.4, m_1(\Theta) = 0.6 \tag{12}$$

$$C_2 : m_2(\mathbf{Same},) = 0.6, m_2(\Theta) = 0.4 \tag{13}$$

$$C_3 : m_3(\mathbf{Same},) = 0.7, m_3(\Theta) = 0.3. \tag{14}$$

Applying Dempster's rule results in

$$m_{1,2,3}(\mathbf{Same},) = 0.928, m_3(\Theta) = 0.072., \tag{15}$$

from which we can conclude that it is very likely that the two aliases represent the same person.

V. Discussion

We have here presented a number of alias matching techniques that can be used for merging aliases, but how can one identify the physical person that authored the message? This question is outside the scope of our research, but the police or intelligence services can in some cases get information about the IP addresses that has been used when making the postings. Such an IP address may however not necessarily be of interest, since people can use dynamic IP numbers, computers at Internet cafes, etc. Other ways to identify the physical person is to use other information that is revealed in the messages. Example of such information could be expressed relationships, expressed information about location or other personal information that can be deduced to a physical person.

A dilemma that arises when searching for violent extremism is that surveillance can not always be based on suspicion of crime. Potentially integrity-violating methods such as wiretapping of phones may generally be used only when the police have reasons to believe that a crime has been committed or is being planned and permission for the interception must be obtained from a court. Such a procedure requires that the police know the identity of the suspect. This is not possible when identifying violent extremists with unknown identity, where it is instead necessary to watch, e.g., a specific forum and everybody who are active in it. The problem is similar to camera-surveillance of public places. The cameras are meant to be used to detect criminal behavior, but have the drawback of monitoring also innocents. One partial solution to this can be to only monitor web sites that are known homes for people with violent extremism connections, but this limits the possibility to find extremists before they strike against society. Moreover, there is still a risk that people who are not planning to commit any crimes.

The needs of the law enforcement and intelligence communities and the right to privacy must be balanced. It should however be noted that analysts are checking extremist forums already today. It is always a human analyst that should check the reasons for why a user has been classified as expressing violent extremism, and if actions should be taken to bind an author to a physical person, and to collect more information using other means. The initial ranking of authors should only be used to direct the analysts in their work and it is of great importance to use a mixed initiative system with a human-in-the-loop as a central component. Using automatic natural language processing may in fact reduce integrity problems, since much legal discussions can be filtered out, leaving a smaller set of radical texts to be checked by analysts.

Lastly, a promising new research area is integrity-preserving data mining, which aims to develop algorithms whose design take account of both ethical and privacy aspects. The algorithms developed thus ensure that privacy and integrity is protected when analyzing data from, e.g., the Internet.

VI. Conclusion and Future Work

We have presented various character-based, graph-based, and text mining-based methods for entity matching, as well as a method for combining the outputs of the various entity matching techniques. The presented methods have been applied on the domain of web intelligence, or more specifically, for detecting and combining messages posted by an individual using multiple aliases. This is one important part of a larger system, where the overall goal of the larger system is to support the work of intelligence analysts by filtering out individuals that based on the content of their online behavior can be suspected to commit severe crimes related to violent extremism.

We see several possibilities for future work in this area. The concepts presented here need to be evaluated thoroughly,

[3]The extension to cases where we do get such evidence is trivial.

both as standalone tools and integrated into our framework for detection of lone-wolf terrorists [2]. There is also room for improving the algorithms, particularly in the area of integrity and privacy preservation. An interesting research question is what can be done if we purposefully limit the amount of data that is collected, to conform to privacy and integrity guidelines.

Although the focus here has been on the intelligence domain only, the presented algorithms and the fusion of them are general enough to be applicable also for other domains, such as for merging authors in researchers' co-citation networks.

ACKNOWLEDGMENT

This research was financially supported by Vinnova through the Vinnmer-programme, and by the Swedish Armed Forces Research and Development Programme.

REFERENCES

[1] E. Pressman, "Risk assessment decisions for violent political extremism 2009-02," in *Report, Public Safety Canada*, 2009.

[2] J. Brynielsson, A. Horndahl, F. Johansson, L. Kaati, C. Mrtenson, and P. Svenson, "Analysis of weak signals for detecting lone wolf terrorists," in *Submitted to EISIC*, 2012.

[3] R. Hölzer, B. Malin, and L. Sweeney, "Email alias detection using social network analysis," in *Proceedings of the 3rd international workshop on Link discovery*, ser. LinkKDD '05. ACM, 2005, pp. 52–57.

[4] V. I. Levenshtein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, vol. 10, pp. 707+, Feb. 1966.

[5] W. E. Winkler and Y. Thibaudeau, "An Application Of The Fellegi-Sunter Model Of Record Linkage To The 1990 U.S. Decennial Census," in *U.S. Decennial Census. Technical report, US Bureau of the Census*, 1987.

[6] M. A. Jaro, *UNIMATCH: A Record Linkage System*. Bureau of the Census, Washington, 1978.

[7] R. Zheng, L. J, Z. Huang, and H. Chen, "A framework for authorship analysis of online messages: Writing-style features and techniques," *J. Amer. Soc. Inf. Sci. Technol.*, vol. 57, pp. 378–393, 2006.

[8] A. Abbasi and H. Chen, "Writeprints: A stylometric approach to identity- level identification and similarity dectection in cyberspace," *ACM Transactions on Information Systems*, vol. 26, 2008.

[9] S. Kim, H. Kim, T. Weninger, and J. Han, "Authorship classification: a syntactic tree mining approach," in *Proceedings of the ACM SIGKDD Workshop on Useful Patterns*, 2010.

[10] M. Fissette, "Author identification in short texts," in *Bachelor theses, Raboud Universiteit Nijmegen*, 2010.

[11] J. Novak, P. Raghavan, and A. Tomkins, "Anti-aliasing on the web," in *Proceedings of the 13th international conference on World Wide Web*, ser. WWW '04. ACM, 2004, pp. 30–39.

[12] J. Scott, *Social Network Analysis: A Handbook*, 2nd ed. London: Sage Publications, 2000.

[13] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[14] J. Brynielsson, L. Kaati, and P. Svenson, "Social positions and simulation relations," *Journal of Social Network Analysis and Mining*, vol. 2, no. 1, pp. 39–52, 2012.

[15] E. A. Leicht, P. Holme, and M. E. J. Newman, "Vertex similarity in networks," *Physical Review E*, vol. 73, no. 2, pp. 026 120+, Feb. 2006. [Online]. Available: http://dx.doi.org/10.1103/PhysRevE.73.026120

[16] L. Adamic and E. Adar, "Friends and neighbors on the Web," *Social Networks*, vol. 25, no. 3, pp. 211–230, 2003.

[17] H. Bostrm, R. Johansson, and A. Karlsson, "On evidential combination rules for ensemble classifiers," in *Proceedings of the 11th International Conference on Information Fusion*, 2008.

[18] G. Shafer, *A mathematical theory of evidence*. Princeton university press Princeton, NJ, 1976.