MODELING OBSERVATION IMPORTANCE FOR ALIAS DETECTION

Patrick Pantel

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292
pantel@isi.edu

ABSTRACT

Alias problems are commonly encountered in the intelligence community, social network analysis, databases, biology, and marketing. Inspired by recently developed technologies in natural language processing, we propose an information theoretic approach for automatically detecting aliases. The algorithm discovers the most informative connections (e.g. emails, phone calls, and transactions) between entities, greatly reducing the search space, and then compares them to identify entities exhibiting similar behaviors. We test our model by applying it to the task of discovering duplicate facilities in heterogeneous environmental databases. Given our system's top-5 guesses for each facility, we extracted with 92% accuracy 89% of the true aliases.

INTRODUCTION

Alias detection is the problem of uncovering duplicate or near-duplicate entities in a dataset. Problem domains can be as simple as datasets containing accidentally replicated data, or as complex as populations containing criminals or terrorists wielding multiple identities. Teasing out duplicate or near duplicate entities in the later case is a serious and challenging problem.

Alias problems are commonly encountered in the intelligence community when performing background checks or, in general, when tracking individuals from a broad population. Often, simple orthographic cues indicate an alias, as in *Osama bin Laden* and *Usama bin Laden*, for example. Other times, semantic variations can be detected, as in for example *Richard Fendlebaum* and *Dick Fendlebaum*.

Malicious individuals, however, can easily fool such verifications by assigning completely different labels to their identities. But, their *behaviors* are likely to be similar since they are much harder to fake or separate across identities. Behaviors can be observed from various sources of information such as communications (emails, phone calls, chats), transaction material (financial transactions, travel logs, shipments), social links, etc.

For large populations, the total number of such observations can become enormous, with only a small portion of *important* observations overlapping for aliases. Consider the following scenario of a population of Southern California residents and two particular residents *John Doe*

and *Alex Forrest*. If you were told that last year both *John* and *Alex* called the Hollywood area about 21 times a month, then would this increase your confidence that *John* and *Alex* are the same person? Yes, certainly. Now, suppose I also told you that *John* and *Alex* both called Kabul about 21 times a month. Intuitively, this observation yields much more evidence that *John* and *Alex* are similar or aliases since not many Southern California residents call Kabul with such frequency. Measuring the relative importance of such observations and leveraging them to detect aliases is the topic of this paper. We will outline an information theoretic framework that models the importance of observations by capturing the intuition of the above example.

The remainder of this paper is organized as follows. In the next section, we review previous approaches to alias detection. We then present five important problems that can be cast as an alias detection problem. Subsequently, we outline our information theoretic model and present experimental results on the task of database integration. Finally, we conclude with a discussion and future work.

RELATED WORK

Most previous solutions to the alias problem search for morphologic, phonetic, or semantic variations of the labels associated with the entities. One of the earliest approaches, patented in 1918 by Margaret O'Dell and Robert Russell, was a rule-based system that matched labels which are roughly phonetically alike. This algorithm, later refined as the Soundex matching algorithm [8], removes vowels and represents labels with six phonetic classifications of human speech sounds (bilabial, labiodental, dental, alveolar, velar, and glottal).

Recently, researchers have begun looking at combining orthographic (and phonetic) features with semantic features. In addition to string edit distance features, [2] and [6] began looking at the behavioral observations that we introduced in the previous section. They asserted connections between entities for each interrelation present in a link data set, ignoring the actual relation types. Adding these semantic cues outperformed previous methods like Soundex. Unlike these approaches, our technique makes use of the link labels (e.g. relation types such as *email*, *financial transaction*, *travel to*, etc.) Also, our method automatically determines the importance of each link.

A related problem in the natural language processing community is automatic spelling correction. The most widely used systems are based on Shannon's noisy channel model [3][7]. The systems assume that the word that was meant to be written was altered by some corruption model (the noisy channel). A decoder is then trained on tagged examples to reconstruct the original (intended) word given the surface error.

Another related but different problem is when multiple entities are referred to by the same label [10]. For example, the name Michael Jackson refers not only to the singer, but also to the bank president, the talk show host, and the author of several books about beer. This problem is important in natural language applications, such as question answering, which must answer questions such as "Where is the Taj Mahal?" and must select between candidate answers such as Agra, India or Atlantic City, NJ. Most approaches to name resolution have used clustering techniques over coreference chains [1], multiple syntactic and semantic features [9], and over referents by first applying a maximum entropy model that estimates the probability that two labels refer to the same entity [5].

DID YOU KNOW? FIVE IMPORTANT ALIASING PROBLEMS

Solving the aliasing problem is important for many different purposes in various domains such as the intelligence community, social network analysis, databases, biology, and marketing. In some cases, it can be used to flag malicious intents while in others it can be used to clean data and to link knowledge. In this section, we describe five problems that exhibit an aliasing problem at the core: identity thefts, identifying and monitoring terrorist cells, data integration, social network analysis, and author identification. For each of these problems, we describe below the entities of the population as well as some of the behavioral observations that may be available for modeling in our system.

Identity Theft

Identity theft is the fastest growing financial crime in the U.S. According to a Federal Trade Commission survey, it is estimated that 27 million Americans have been victims of identity theft in the past five years. In one of its incarnations, thieves acquire social security numbers and other personal information in order to fraudulently acquire credit cards, bank accounts and cell phone accounts. It can be years before unsuspecting victims discover that their credit is ruined, they owe large sums of money to creditors, or worse that they are prosecuted for financial frauds.

Supposing authorities are tracking a known identity thief, we expect that our system will be capable of discovering the identities that were stolen by modeling their usage behaviors. The population consists of identities (e.g. all social security numbers) and examples of observed behaviors may include communications, financial transactions, travel information, etc.

Another type of identity theft is when a criminal fraudulently acquires identities, but then sells them instead of using them himself or herself. Here, we expect that our method would not work since the identities are used in different ways by different people.

Terrorist Cells

In 2004, the FBI estimated that al-Qaida sleeper cells were believed to be operating in 40 states, awaiting orders and funding for new attacks on American soil. It is also approximated that between 2,000 and 5,000 terrorist operatives currently operate in the U.S.

Generally, it is believed that the social links as well as the similar religious and cultural background of terrorists yield cells of very similar people. Treating each terrorist cell as an identity, one can perform group detection to identify its members using aliasing models. Here, the population consists of, say, people living in America and known terrorist cells. The observed behaviors may include communications, financial transactions, travel information, etc. We expect that certain observations will be rare and some will be erroneous, but our model is quite tolerant to sparsity and noise.

Data Integration

In many large organizations and especially government, data is split in many different ways and is collected at different times by different people. The resulting massive data heterogeneity means that staff cannot effectively locate, share, or compare data across sources, let alone achieve computational data interoperability.

What is needed is a method to integrate the data in two comparable but heterogeneous data sources. We can partially address this problem by finding matching records across sources. The population consists of all possible records in two data sources, and the observable behaviors are the data fields that are contained in each record. For example, given records of contact names along with contact information such as phone number, address, zip code, etc., we can use this contact information as the observations for our model.

Social Network Analysis

User modeling and recommendation systems are easily hampered by duplicate users in social networks. Duplicate identities, whether accidentally created or not, are prominent. The population here consists of all identities in the network and the observed behaviors include personal attributes such as contact information, colleges attended, board memberships, etc., as well as communication links.

Author Identification

Author identification is the task of attributing authorship to anonymous text [13]. The problem exists in many forms like identifying plagiarism, accrediting famous historical writings to their true authors, and even ownership or copyright legal battles. In the context of alias detection, techniques for author identification can be used to model a source's communication style by looking at the very words and expressions she uses. In an electronic world where one's population consists solely of citizens of the Internet, or Netizens, then one can simply observe the language use and style to find aliases.

INFORMATION THEORETIC MODEL

Many alias detection problems consist of large numbers of entities and observations. However, as illustrated in the introduction, certain observations are much more important than others when matching entities (e.g. calling Kabul vs. Hollywood). In this section, we outline an information theoretic model that measures this importance.

Before we formally describe the model, we appeal to the reader's intuition. Recall our phone call scenario from the introduction where we were asked if Southern California residents *John Doe* and *Alex Forrest* are the same person given their monthly phone call records. Figure 1 a) lists *John*'s most frequent phone calls along with the call frequencies. It is not surprising that a Californian would call L.A., Culver City, Anaheim, and even D.C. If *Alex* had similar calling patterns to these four cities, it would certainly increase our confidence that him and *John* are the same person, but obviously our confidence would increase much more if *Alex* called the more *surprising* cities Kabul and Mosul.

Looking only at the frequencies of the calls in Figure 1 a), however, one would put more importance on matching calls to L.A. than to Kabul. The goal of our framework is to have a better measurement than frequency for the importance of each call and to re-rank them in order of information content. Figure 1 b) illustrates the frequencies of *John* calling *D.C.*, *John* calling any city, and anyone calling *D.C*, whereas c) illustrates the same for *Kabul*. Notice that although D.C. is much more frequent than Kabul, many more people in the population call D.C. than Kabul. Our model leverages this idea by adding importance for a city to which *John* calls

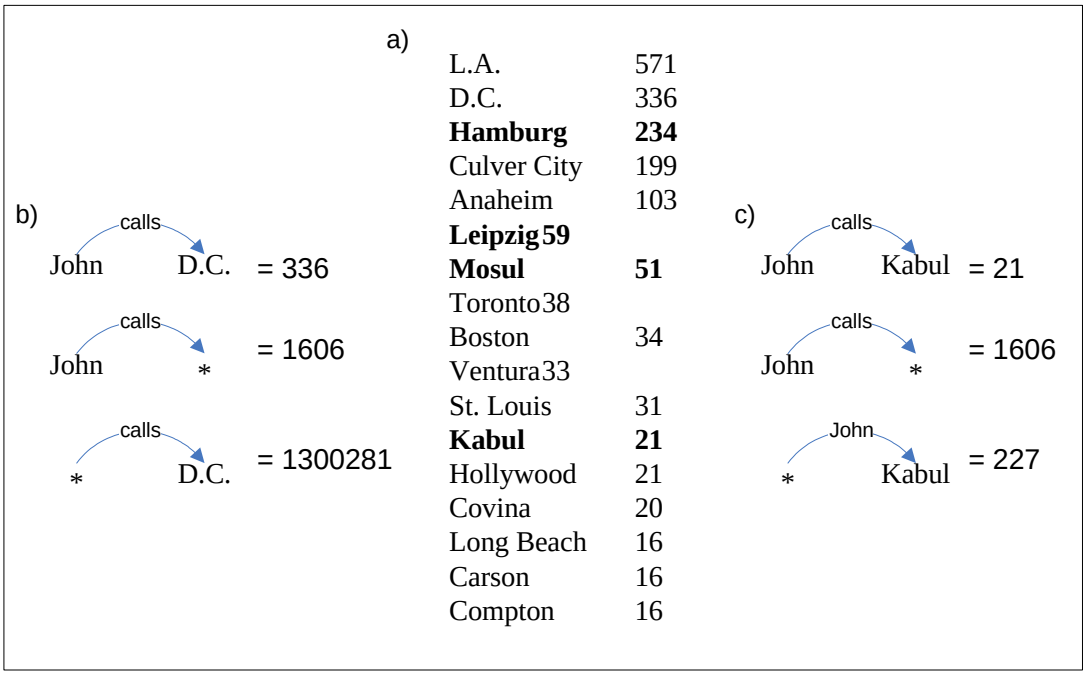

Figure 1. Identifying important observations in our fictitious scenario of phone calls placed by Southern California residents. a) Frequency of phone calls placed monthly by *John Doe*. b) Frequency of calls placed by *John* and others (*) to D.C. and other cities (*). c) Frequency of calls placed by *John* and others (*) to Kabul and other cities (*).

frequently and by deducting importance if many people in the general population call the same city. After applying our model, the cities in Figure 1 a) are re-ranked as follows:

Kabul	**7.88**
Mosul	**7.05**
Leipzig	**5.78**
Hamburg	**5.58**
Culver City	5.48
D.C.	5.33
L.A.	4.77
Anaheim	4.46
Ventura	4.38
Toronto	4.36
Boston	4.31
Covina	2.91
Compton	2.86
St. Louis	2.40
Long Beach	2.03
Carson	1.62
Hollywood	1.43

The four cities that were bold in Figure 1 b) are now at the top of this list, and consequently more importance is now put on matched calls to Kabul than matched calls to Hollywood. We now formally introduce the model.

Pointwise Mutual Information

Pointwise mutual information is commonly used to measure the association strength between two events [4]. It essentially measures the amount of information one event gives about another. For example, knowing that a Southern Californian calls L.A. is not informative since most residents call L.A. Conversely, if he or she calls Kabul, then this is an informative observation. The pointwise mutual information between two events *x* and *y* is given by:

$$mi(x,y) = \log \frac{P(x,y)}{P(x)P(y)}$$

Pointwise mutual information is high when *x* and *y* occur together more often than by chance. It compares two models (using KL-divergence) for predicting the co-occurrence of *x* and *y*: one is the MLE (maximum-likelihood estimation) of the joint probability of *x* and *y* and the other is some baseline model. In the above formula, the baseline model assumes that *x* and *y* are independent. Note that in information theory, mutual information refers to the mutual information between two random variables rather than between two events as used in this paper. The mutual information between two random variables *X* and *Y* is given by:

$$MI(X,Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}$$

The mutual information between two random variables is the weighted average (expectation) of the pointwise mutual information between all possible combinations of events of the two variables.

For each entity in our population *e*, we first construct a frequency count vector $C(e) = (c_{e1}, c_{e2}, \ldots, c_{em})$, where *m* is the total number of features (observations) and c_{ef} is the frequency count of feature *f* occurring for entity *e*. Here, c_{ef} is the number of times we observed feature *f* for entity *e*. For example, in Figure 1 b), one feature for *e* = *John Doe* is *f* = *Kabul* with frequency 21.

We then construct a mutual information vector $MI(e) = (mi_{e1}, mi_{e2}, \ldots, mi_{em})$ for each entity *e*, where mi_{ef} is the pointwise mutual information between *e* and feature *f*, which is defined as:

$$mi_{ef} = \log \frac{\frac{c_{ef}}{N}}{\frac{\sum_{i=1}^{n} c_{if}}{N} \times \frac{\sum_{j=1}^{m} c_{ej}}{N}}$$

where *n* is the number of entities and $N = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}$ is the total frequency count of all features of all entities.

In our example from Figure 1 b), assuming that $N = 1.32 \times 10^{12}$, the mutual information for *e* = *John Doe* and feature *f* = *D.C.* is:

$$mi_{ef} = \log \frac{\frac{336}{1.32 \times 10^{12}}}{\frac{1{,}300{,}281}{1.32 \times 10^{12}} \times \frac{1606}{1.32 \times 10^{12}}} = 5.33$$

and for *f* = *Kabul*:

$$mi_{ef} = \log \frac{\frac{21}{1.32 \times 10^{12}}}{\frac{227}{1.32 \times 10^{12}} \times \frac{1606}{1.32 \times 10^{12}}} = 7.88$$

A well-known problem is that mutual information is biased towards infrequent entities/features. We therefore multiply mi_{ef} with the following discounting factor [11]:

$$\frac{c_{ef}}{c_{ef}+1} \times \frac{\min\left(\sum_{i=1}^{n} c_{ei}, \sum_{j=1}^{m} c_{jf}\right)}{\min\left(\sum_{i=1}^{n} c_{ei}, \sum_{j=1}^{m} c_{jf}\right)+1}$$

Similarity Model

Now that we have a method of ranking observations according to their relative importance, we still need a comparison metric for determining the likelihood that two entities are aliases. The requirement is that the metric handles large feature dimensions and that it not be too sensitive to 0-valued features. That is, the absence of a matching observation is not as strong an indicator of dissimilarity as the presence of one is an indicator of similarity. Some measures, like the Euclidean distance, do not make this distinction. Many models could apply here; we chose the cosine coefficient model [12]. The similarity between each pair of entities e_i and e_j, using the cosine coefficient of their mutual information vectors, is given by:

$$sim(e_i, e_j) = \frac{\sum_f mi_{e_i f} \times mi_{e_j f}}{\sqrt{\sum_f mi_{e_i f}^{\ 2} \times \sum_f mi_{e_j f}^{\ 2}}}$$

This measures the cosine of the angle between two mutual information vectors. A similarity of 0 indicates orthogonal vectors whereas a similarity of 1 indicates identical vectors. For two very similar elements, their vectors will be very close and the cosine of their angle will approach 1.

Detecting Aliases

We now have all the pieces of the puzzle to detect aliases from a given population and a set of observations. Figure 2 illustrates our system architecture for alias detection. The various observations (e.g. travel logs, communications, social links, etc.) are first processed through our mutual information model to generate a ranked composite view of the important observations. Then, our similarity model is used to detect and rank candidate aliases for each entity in our population.

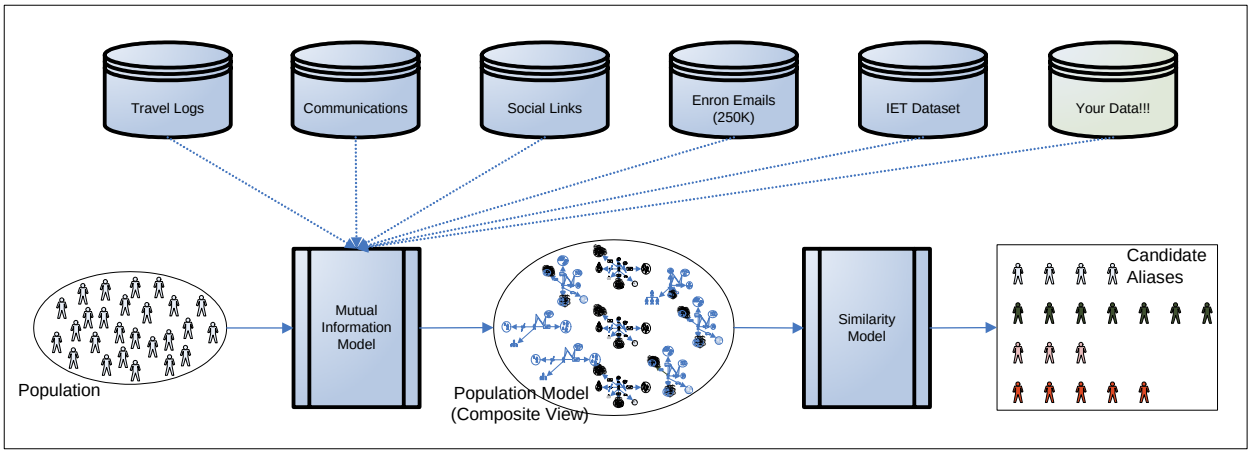

Figure 2. System architecture. First, the mutual information model is applied to the observations of a population. Then, the similarity model yields candidate aliases for each entity in the population.

EXPERIMENTAL RESULTS

We evaluate our model on one of the five problems described earlier in this paper: database integration. Below, we describe our experimental setup and present our results.

Experimental Setup

We acquired two heterogeneous environmental databases containing overlapping emissions inventories for various facilities (e.g. gas stations, companies, universities, etc.) The first database was provided by the Santa Barbara County Air Pollution Control District (SBCAPCD), which contains a complete archive of the emissions inventory it conducted for 2001. The second database was provided by the California Air Resources Board (CARB) in the form of the statewide emissions inventories for 2001.

For each database, we extracted all records containing a facility identifier and constructed the mutual information vectors described in the previous section using each record cell as an observation (feature). Our task then is to identify the facilities in SBCAPCD that map to the CARB database, without making use of any facility labels.

We experimented with two representation models:

- *BOW*: Bag-of-words. This representation initially counts the frequency of each cell value for each facility before computing the mutual information vectors;
- *SOW*: Set-of-words. This representation initially either sets the frequency to 0 for unobserved cell values of a facility or to 1 for observed values before computing the mutual information vectors.

For example, suppose that facility *fac* occurred in multiple records with the cell value “large” eight times, the value “10” twice, and never with the value “xyz”. Then, under the *BOW* model, the observations for *fac* are {large:8, 10:2, xyz:0}; under the *SOW* model, the observations are {large:1, 10:1, xyz:0}. These frequency vectors are then converted into mutual information vectors as described in the previous section.

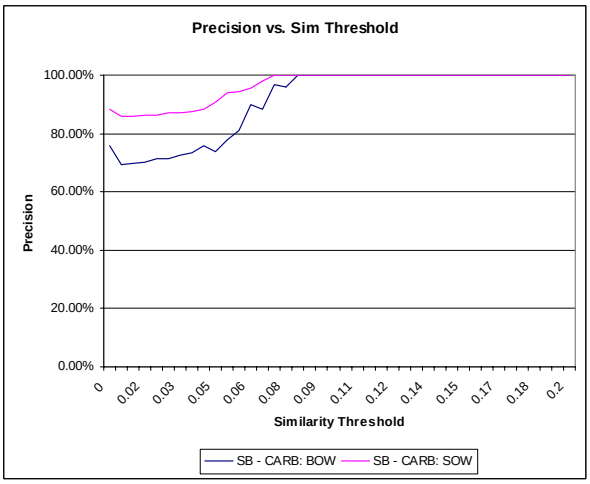

Figure 3. System precision on varying cosine similarity thresholds.

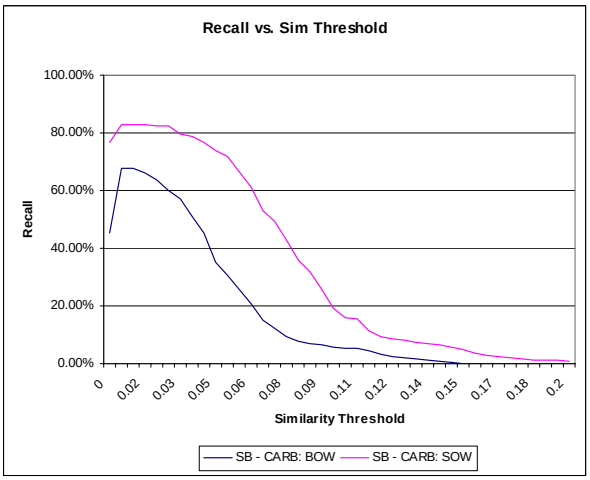

Figure 4. System recall on varying cosine similarity thresholds.

After applying the cosine coefficient similarity metric, we obtain a ranked list of the most similar CARB facilities for each SBCAPCD facility. To identify potential aliases, a similarity threshold θ must be defined as a cutoff in this list. In our experiments, we tested different values of θ.

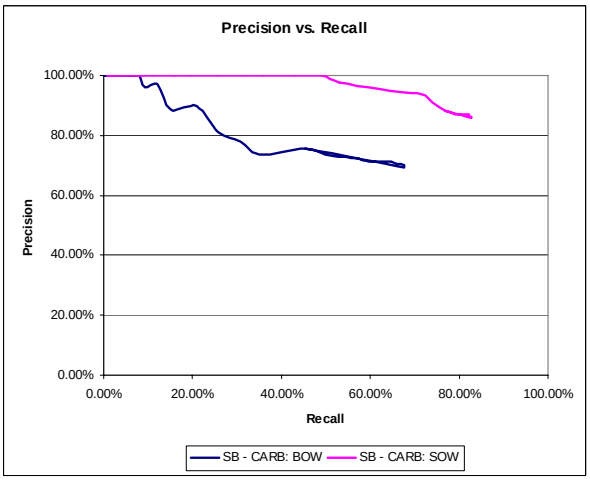

Figure 5. Precision vs. recall tradeoff.

Precision and Recall

To evaluate our system, we first manually mapped each facility in SBCAPCD to their corresponding CARB facilities. We then measured the system's precision and recall on the task of automatically detecting the matching facilities across databases. The precision of the system is the percentage of correct detections:

$$Precision = \frac{C}{T_A}$$

where C is the number of correctly detected aliases and T_A is the total number of proposed aliases by the system. The recall is the percentage of manually extracted aliases, T_M, which were retrieved by the system:

$$Recall = \frac{C}{T_M}$$

Precision and recall measure the tradeoff between identifying aliases correctly and retrieving all of them.

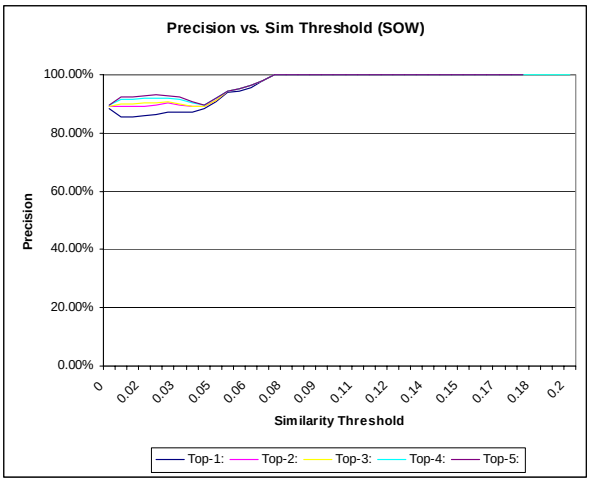

Figure 6. System precision in the Top-K returned aliases for the Set-of-words model on varying cosine similarity thresholds.

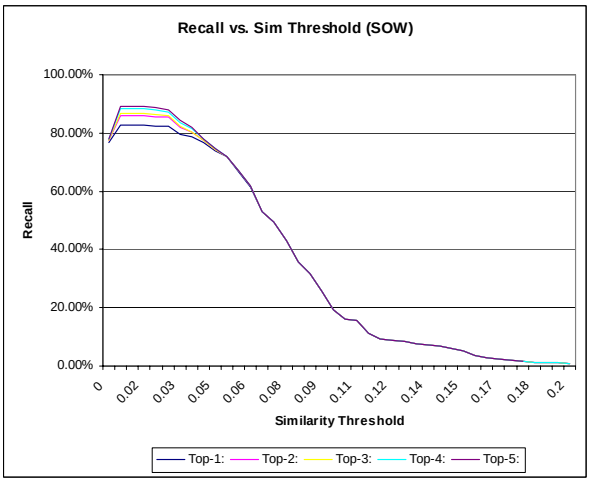

Figure 7. System recall in the Top-K returned aliases for the Set-of-words model on varying cosine similarity thresholds.

Figures 3 and 4 illustrate the precision and recall of our system with varying similarity thresholds. The *SOW* model consistently outperforms the *BOW* model. We suspect that this is due to the many observations that coincidently share the same value (for example, many data cells contain the value 1 to represent a measurement, a unit, and even a Boolean). The *SOW* model is less sensitive to these chance mappings since it simply records whether the value was present or not. Increasing the similarity threshold θ increases the precision of the system but decreases the recall. This indicates that the similarity model correctly makes fewer errors when it assigns higher confidence to an alias. The tradeoff between precision and recall is illustrated in Figure 5.

Allowing the system to return its top-*K* best guesses for each facility could potentially significantly increase the recall of the system while still greatly reducing the amount of time an analyst has to spend to verify the system (i.e. she would look at only *K* guesses for each facility instead of each possible mapping). We experimented with varying values of *K* and measured the precision and recall of the *SOW* model. The results are shown in Figures 6 and 7. At low similarity thresholds, where the system has less confidence on matches, both the precision and recall increase when returning five guesses instead of just one. However, with larger thresholds, the system either correctly identifies an alias in the first position, or completely misses it altogether.

CONCLUSIONS AND FUTURE WORK

Aliasing is a common problem encountered in various domains from the intelligence community to social network analysis, databases, biology, and marketing. Instead of detecting aliases by looking for morphological, phonetic, or semantic cues in entity labels, we focus our attention on behavioral cues exhibited by the entities (e.g. communications, financial transactions, social links, etc.) For large populations, the total number of such observations can become enormous, with only a small portion of the *important* observations overlapping for aliases. In this paper, we proposed an information theoretic model for measuring this importance and leveraging it to

detect aliases. We applied our model to the task of detecting duplicate facilities in two heterogeneous environmental databases. Below is a summary of the evaluation results:

- with 100% accuracy, our system was able to extract 50% of the matching facilities;
- with 90% accuracy, our system was able to extract 75% of the matching facilities;
- for a given facility and the top-5 mappings returned by our system, with 92% accuracy, our system was able to extract 89% of the matching facilities.

At a minimum, our model can dramatically reduce the time a human needs to find matching facilities (looking at only five possible aliases for each facility would recall 89% of the possible aliases). However, the power of the model is critically dependent on gathering the right observations that aliases might share, which in itself is a very interesting avenue of future work. Given the right types of observations, our model has the potential to solve several serious and urgent problems such as terrorist detection, identity thefts, and data integration.

REFERENCES

[1] Bagga, A. and Baldwin, B. 1998. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of COLING/ACL-98*. Montreal, Canada.

[2] Baroni, M.; Matiasek, J.; and Trost, H. 2002. Unsupervised discovery of morphologically related words based on orthographic and semantic similarity. In *Proceedings of the Workshop on Morphological and Phonological Learning of ACL/SIGPHON-2002*. pp. 48-57. Philadelphia, PA.

[3] Brill, E. and Moore, R. C. 2000. An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of ACL-00*. pp. 286-293. Hong Kong.

[4] Church, K. and Hanks, P. 1989. Word association norms, mutual information, and lexicography. In *Proceedings of ACL-89*. pp. 76–83. Vancouver, Canada.

[5] Fleischman, M. B. and Hovy, E. H. 2004. Multi-Document Person Name Resolution. In *Proceedings of the ACL Workshop on Reference Resolution*. Barcelona, Spain.

[6] Hsiung, P. 2004. *Alias Detection in Link Data Sets*. Technical report CMU-RI-TR-04-22, Carnegie Mellon University.

[7] Kernighan, M.D.; Church, K.; and Gale, W. 1990. A pelling correction program based on a noisy channel model. In *Proceedings of COLING-90*. pp. 205-211. Helsinki, Finland.

[8] Knuth, D. 1973. The Art of Computer Programming – Volume 3: Sorting and Searching. Addison-Wesley Publishing Company.

[9] Mann, G. and Yarowsky, D. 2003. Unsupervised Personal Name Disambiguation. In *Proceedings of CoNLL-2003*. Edmonton, Canada.

[10] Martinich, A.P. 2000. *The Philosophy of Language*. Oxford University Press. Oxford, UK.

[11] Pantel, P. and Lin, D. 2002. Discovering word senses from text. In *Proceedings of SIGKDD-02*. pp. 613–619. Edmonton, Canada.

[12] Salton, G. and McGill, M.J. 1983. Introduction to Modern Information Retrieval. McGraw Hill.

[13] Stamatatos, E.; Fakotakis, N.; and Kokkinakis, G. 2001. Computer-based Authorship Attribution without Lexical Measures. *Computers and the Humanities*, Volume 35, Issue 2, May 2001. pp. 193-214.