Bradley Malin. Unsupervised name disambiguation via social network similarity. In Proceedings of the *Workshop on Link Analysis, Counterterrorism, and Security*, in conjunction with the *SIAM International Conference on Data Mining*. Newport Beach, CA. 2005; 93-102.

Unsupervised Name Disambiguation via Social Network Similarity*

Bradley Malin†

Abstract

Though names reference actual entities it is nontrivial to resolve which entity a particular name observation represents. Even when names are devoid of typographical error, the resolution process is confounded by both ambiguity, where the same name correctly references multiple entities, and by variation, when an entity is correctly referenced by multiple names. Thus, before link analysis for surveillance or intelligence-gathering purposes can proceed, it is necessary to ensure vertices and edges of the network are correct. In this paper, we concentrate on ambiguity and investigate unsupervised methods which simultaneously learn 1) the number of entities represented by a particular name and 2) which observations correspond to the same entity. The disambiguation methods leverage the fact that an entity's name can be listed in multiple sources, each with a number of related entity's names, which permits the construction of name-based relational networks. The methods studied in this paper differ based on the type of network similarity exploited for disambiguation. The first method relies upon exact name similarity and employs hierarchical clustering of sources, where each source is considered a local network. In contrast, the second method employs a less strict similarity requirement by using random walks between ambiguous observations on a global social network constructed from all sources, or a community similarity. While both methods provide better than simple baseline results on a subset of the Internet Movie Database, findings suggest methods which measure similarity based on community, rather than exact, similarity provide more robust disambiguation capability.

Keywords: Disambiguation, Social Networks, Random Walks, Multi-class Clustering

1 Introduction

Technological advances have sustained a continuing increase in our abilities to gather, store, and model information at the entity-specific level. With respect to entity-specific, or social, networks, the types of relationships which are learnable are vast and can provide detailed knowledge ranging from individual preferences to organizational structures. Yet, before knowledge regarding an entity or relationships between entities can be extracted from relational systems we must first attend to a more fundamental feature of data: correctness. Specifically, we must be able to decide when two pieces of data correspond to the same entity or not. Failure to ensure correctness can result in the inability to make inferences or the learning of false knowledge. The ability to decide when two or more pieces of data refer to the same entity is crucial not only for correct network construction and analysis, but to a wide range of critical processes, including data fusion, cleaning, profiling, speech recognition, and machine translation.

For surveillance and counterterrorism analysis, the relational data of interest is often made up of names, such that a vertex refers to a particular name and an edge specifies the relationship between two names. However, even when names are devoid of typographical errors, there are additional confounders to data correctness. First, there can exist name variation, where multiple names correctly reference the same entity. Second, there can exist name ambiguity, such that the same name correctly references multiple entities. While both problems must be accounted for, this paper concentrates on the basic aspects, and how to resolve, ambiguity. The basic question we ask is, how do you resolve which particular entity is referred to, or disambiguate, various observations of the same name?

Disambiguation is by no means a trivial feat, and the manner by which an individual makes the decision is often contingent on the available contextual clues as well as prior, or background, information. For example, when a reader encounters the name "George Bush", the reader must decide if the name represents "*George H.W. Bush*" - the 41st President of the United States of America, or "*George W. Bush*" - the 43rd president, or some other individual of lesser notoriety. How does one determine whom the name corresponds to? When the name is situated in a traditional communique, such as a news story, we tend to rely on linguistic and biographical cues. If the name is situated in the following sentence, "*George Bush was President of the*

*Partially supported by the Data Privacy Laboratory at Carnegie Mellon University and NSF IGERT 9972762 in CASOS.

†Institute for Software Research International, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA 15213-3890, malin@cs.cmu.edu

United States of America in 1989.", then, with basic knowledge of American history, it is clear the story refers to the elder "*George H.W. Bush*".

Though spoken conversations and written communications between entities are structured by known grammars there is no requirement for text-based documents to provide traditional semantic cues. One such counter scenario, which explicitly concerns social networks, occurs when documents are merely rosters that consist of nothing but names. [30] To relate information corresponding to the same entity in this type of environment, disambiguation methods must be able to leverage list-only information. Models employed in natural language processing [32], such as those available in the sentence regarding the American President, are not designed to account for this new breed of semantics.

There has been some headway made in the design of less structure dependent disambiguation methods. [6, 7, 21] However, these methods are often tailored to assumptions and characteristics of the environments where the references reside. For example, some methods leverage the covariates of references (i.e. the observation of two references in the same source) or require that social groups function as cliques. [6, 7] This model expects environments in which strong correlations exist between pairs or sets of entities, such that they often co-occur in information sources. While closely knit groups of entities provide an ideal scenario, it is not clear if such social settings manifest in the real world. In contrast, it is feasible, and intuitive, to leverage less directly observed relationships. This is precisely the route explored in this paper. We consider networks of the references in question, such that one can leverage "community" structures among entities. By studying communities of entities, we exploit relationships between entities which have minimal, or no, observed interactions. This is extremely powerful, since it allows for disambiguation when covariates are weak or the social network of entities is less centralized.

In this research paper, we investigate the degree to which disambiguation methods can be automated using relational information only. More specifically, given only a set of observations of names from information sources, such as webpages, can we construct an automated system to determine how many entities correspond to each particular name? Furthermore, can we determine which particular name observation corresponds to which underlying entity? The methods discussed in this paper are evaluated on a real world dataset derived from the Internet Movie Database (IMDB). Experimental findings from this research suggest that community similarity, which leverage indirect relationships, is more reliable for disambiguation than similarity methods which rely on direct relationships. In addition, we demonstrate that simple methods, such as those based on random walks can be applied towards estimating community similarity.

The remainder of this paper is organized as follows. In the following section, related research in linkage and disambiguation, including recent developments within the data mining community, is reviewed. In Section 3, the disambiguation methods which are applied in this research are formally introduced and defined. In Section 4, the IMDB dataset is summarized and the results of disambiguation experiments with this dataset are presented. Then, in Section 5, we consider some of the limitations of this research, discuss some of potential extensions, and consider some applications of social network-based disambiguation. Finally, in Section 6, the contributions of this research are summarized.

2 Background and Related Research

There exist a number of approaches that have been applied to disambiguation. In this section, we briefly review previous disambiguation research and where the work presented in this paper differs.

In general, disambiguation methods can be taxonomized on two features: 1) information type and 2) supervision. Information type specifies to whom data corresponds and there are two main types often used for disambiguation: a) personal and b) relational. Personal information corresponds to static biographical (e.g. *George H.W. Bush* was the 41^{st} President) and grammatical (e.g. *fall* used as a noun vs. as a verb) information. To leverage this information, disambiguation methods usually use sets of rules for discerning one meaning from another. In contrast, relational information specifies the interactions of multiple values or terms (e.g. *George H.W. Bush* collocates with *Ronald Reagan* whereas *George W. Bush* collocates with *Dick Cheney*).

The second taxonomizing feature is the supervision of the disambiguation process. In a supervised learning systems, each of the a disambiguation method is trained on labeled sample data (e.g. first sample corresponds to first meaning, second sample corresponds to second meaning, etc.). In an unsupervised learning system, methods are not trained, but instead attempt to disambiguate based on observed patterns in the data.

2.1 Personal Disambiguation. Word sense disambiguation methods initially gained momentum in natural language processing. Early computational methods tagged sentences with parts of speech and disambiguated words/phrases based on part of speech. [8, 19] With the incorporation of a database-backed model, IBM's "Nominator" system [33], uses phrase context

(e.g. punctuation, geographic position in sentence, and capitalization) in parallel with prior knowledge (e.g. known type of entity for names) for disambiguation. Names encountered by the system are matched to names whose context and knowledge have been previously specified. An alternative supervised method is to perform disambiguation using parallel corpora, such as in the cross-lingual context. [28]

Bagga and Baldwin [3] introduced an unsupervised disambiguation model based on sentence comparison for when prior knowledge is unknown. Sentence are parsed into vector-space summaries of words or concepts. Summary pairs are compared and similarity scores above a certain threshold are predicted as the same entity. Mann and Yarowsky [25] extend summaries to parse and structure biographical data, such as birth day, birth year, occupation, and place of birth. Once each name is associated with a simply biography, the name observations are clustered based on similarity of their biographies.

The recently developed "Author-ity" system, is an unsupervised system developed for database queries. Input is provided to this system as an author's name, in the form of last name and first initial. The system returns a list of scientific articles, authored by the name of interest, ranked in decreasing certainty of whether or not an article was authored by the same person. [31] Articles are ranked by performing a pairwise similarity of title, journal name, coauthor names, medical subject headings, language, affiliation, and prevalence of name in the database.

A drawback of personal information dependent methods is their lack of accountability for unstructured information. These methods require rules, grammars, and or multiple attributes for comparison.

2.2 Relational Disambiguation. An alternative approach for natural language disambiguation is based on a probabilistic model of word usage. Lesk [24] extended rule based models to account for the relationship of an ambiguous word with its surrounding words. He demonstrated that for an ambiguous word, overlap in the dictionary definitions' of surrounding text words can be used to disambiguate. Gale et. al. [14] demonstrated that the dictionary definitions are unnecessary provided a representative sample of word covariation was available. In their research, a Naïve Bayes classifier was trained for each ambiguous word and its surrounding words. Given a new word observation for disambiguation, the word was labeled with the definition of the max score classifier. Additional statistical models for using word and concept covariates have been studied. [9, 15, 16, 27, 34] A classifier based on covariance (i.e. the probability that a word occurs with another word) is trained for each meaning of the ambiguous word. For each new ambiguous word occurrence, a sense prediction is made based on which classifier the word, and its surrounding words, are most similar to.

Networks provide a way to construct robust patterns from minimally structured information. Certain word disambiguation methods have employed semantic [11, 18, **?**] networks from corpora for more robust similarity measures. Similarly, other models have considered belief propagation networks and Bayesian models for disambiguation. [12] In this research, we consider the degree to which social networks can be used for disambiguation. Recent research has considered a specific case of social networks for unsupervised social disambiguation network [6, 7], in which both ambiguity and variation problems are tackled simultaneously using an iterative approach akin to expectation-maximization. In the maximization step, two references are predicted as the same entity if they are within a certain "distance" of each other. The distance predictions are achieved in the expectation step, and are calculated as a weighed average of 1) the distance between the observed set of references and 2) the groups which the predicted entity for the observed references is expected to be a part of. In the first measure, a measurement between the attributes of the references is incorporate as used in record linkage research (e.g. *John* vs. *Jon*). The second measure corresponds to the distance between two sets of groups, where a group is a clique of entities representative of the document in which the reference resides in, as predicted from the previous iteration.

A shortcoming of this model is a design tailored to an expectation of how citation networks are organized. The proposed model has not been evaluated on actual collaboration networks, but rather synthetic data in which clique structures are guaranteed to exist. As a result, their approach skews predictions towards groups which are not only equivalent, but function as cliques. This bias can have serious difficulty in a lesser connected environment, or decentralized, environments such as the Internet Movie Database studied in this paper. Clique detection requires what we informally term *exact similarity*, such that relationships between entities must be directly observed (e.g. Alice and Bob are related if they collocate in the same source). Furthermore, this model is not necessarily representative of the space of social networks. It is unclear if this model generalizes to other types of social networks [2, 26], such as small-world [22], hierarchical [29], or cellular [10].

As applied in this research, we incorporate community similarity to relax the direct observation requirement and permit relationships to be established be-

tween entities indirectly. For instance, Alice and Bob may never be observed together, but both Alice and Bob collocate Charlie, Dan, and Fran. Though community similarity measures do not necessarily all types of networks, the goal of this research is to demonstrate their capability in comparison to exact similarity in a controlled environment. We suspect that in a less centralized system, such as the IMDB, similarity measures based on community provide more robust metrics. In following section, we introduce two methods: one dependent on exact similarity and an alternative method which is dependent on community similarity.

3 Disambiguation Models and Methods

In this section, we formalize aspects of disambiguation in a more formal manner. In order to do so, we borrow from set theory and introduce a basic set of terminology, definitions, and notations.

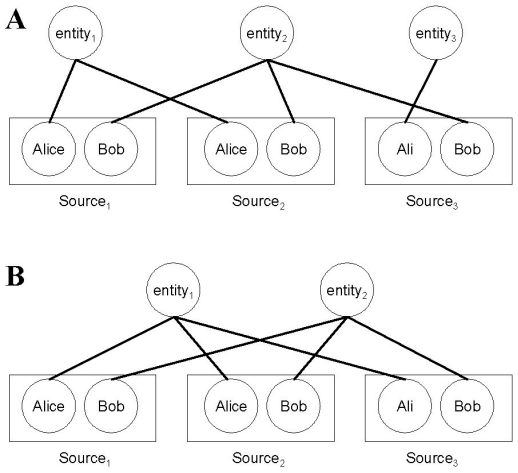

Figure 1: **A**) An example of an ambiguous name *Alice* for $entity_1$ and $entity_3$. **B**) An example of name variation of *Alice* and *Ali* for $entity_1$.

An *entity* is defined as an element from a population of objects. However, entities are not necessarily observed, and thus we consider a set of entities are considered as a set of unobserved, or latent, variables $H = \{h_1, h_2, \ldots, h_k\}$. Rather, there exist a set of objects which are used to reference entities. For this research, we consider these referencing objects to take the form of names. These names manifest in a set of information sources $S = \{s_1, s_2, \ldots, s_m\}$, such that each source s_i consists of a set of extracted names N_i. For example, one can consider a single webpage as a single source. The set of distinct names observed in S is represented by $E = \{e_1, e_2, \ldots, e_n\} = N_1 \cup N_2 \cdots \cup N_m$.

While the same name can be ambiguous to multiple entities, each occurrence of a name references a single entity only. A name which refers to k different entities is called k-ambiguous. This is the scenario depicted in Figure 1.A, where the name *Alice* correctly represents $entity_1$ in $source_1$ and $entity_3$ in $source_3$. Similarly, an entity may be correctly represented by k different names. An entity which is referred to by k different names is called k-variant. In Figure 1.B, $entity_1$ and $entity_2$ are 2- and 1-variant, respectively. For this study, investigation is restricted to 1-variant entities and k-ambiguous names.

In this paper there are two techniques evaluated for name disambiguation, the first leverages directly observed relationships, whereas the second incorporates unobserved, though meaningful, relations. The first technique is a version of hierarchical clustering on sources with ambiguous names only. The second constructs social networks from all sources, regardless of the existence of the ambiguous name of interest. The following sections explain these methods in detail.

3.1 Hierarchical Clustering.

For the first method, each source is represented as a Boolean vector $s_i = [e_{i1}, \ldots, e_{in}]$, where $e_{ij} = 1$ if name e_j is in source s_i and 0 otherwise. Hierarchical clustering is performed using an average linkage criterion calculated as follows. [13] Each source to be clustered is initialized as a singleton cluster. Then, similarity between two clusters c_i, c_j, denoted $csim(c_i, c_j)$, is measured as:

$$csim(c_i, c_j) = \frac{\sum_{s \in c_i, t \in c_j} ssim(s, t)}{|c_i||c_j|}$$

where the similarity between two sources s_i, s_j, denoted $ssim(s_i, s_j)$, can measured using any distance or similarity function. The similarity function of choice for this research is one minus the cosine distance of the vectors of the two source vector representations. More specifically, cosine similarity between two sources is calculated as:

$$ssim(s_i, s_j) = \frac{\sqrt{\sum_{x=1}^{n} e_{ix} e_{jx}}}{\sqrt{\sum_{y=1}^{n} e_{ix}} \sqrt{\sum_{z=1}^{n} e_{iz}}}$$

The most similar clusters are then merged into a new cluster. This process proceeds until either a prespecified stopping criteria is satisfied or all sources reside in one common cluster.

3.2 Random Walks and Network Cuts.

An alternative method considered in this research is the analysis of social networks constructed via names with high certainty. Mainly, we are interested in the partitions of networks as prescribed by random walks from nodes of ambiguous names. One principle difference between the random walk method described in this section and the

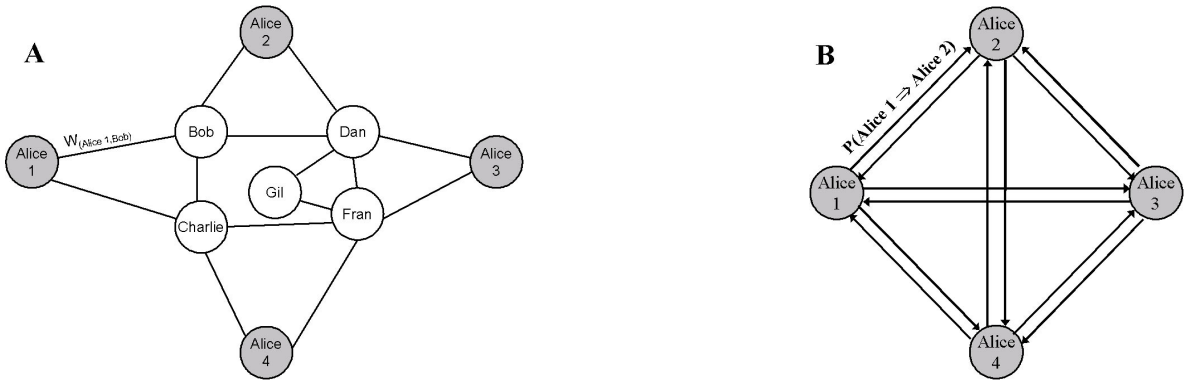

Figure 2: **A**) Social network with a four ambiguous name observations. Nodes connected to ambiguous nodes correspond to original sources. **B**) network with non-ambiguous names removed. The directed edges correspond to the probability of walking from one node to another.

hierarchical clustering of the previous section is the walk is permitted to proceed over nodes (names) which occur in sources devoid of ambiguous names. By doing so, we exploit weak ties, which taken in combination, can permit the discovery of community structures in the graph.

From the set of sources S, a social network is constructed in the following manner. Every distinct name in S is set as a node in the network. An edge exists between two nodes if the names collocate in a source at least one time. The weight of the edge between two nodes i, j is related to the inverse of the number of names observed in a source. This weight is calculated as

$$w_{ij} = \frac{\sum_{s \in S} \theta ijk}{|s|},$$

where θ_{ijk} is an indicator variable with value 1 if names for nodes i and j collocate in source s and 0 otherwise. The reasoning behind this weighting schema is the belief that the lesser number of entities observed in a source, the greater the probability the entities have a strong social interaction. For instance, a website which depicts a list of all students, faculty, and staff of a university conveys less specific information than the class roster for a machine learning graduate course.

In order to test disambiguation in a controlled environment, we make the following adjustment to the networks. For each ambiguous name, we construct a separate network. Basically, the social network is constructed in same manner, except each observation of the ambiguous name of interest is set as its own node in the network. An example network is depicted in image Figure 2.A for the name *Alice*. In this network, *Gil* is indirectly connected to *Alice* through her acquaintances (*Dan* and *Fran*.

Given the social network, we proceed with random walks over the graph. Each walk begins at a node with an ambiguous name observation. The probability a step is taken from node a to node b is the normalized weight of the edge with respect to all edges originating from node a. This probability is calculated as $P(a \rightarrow b|a) = w_{ab} / \sum_j w_{aj}$. Note the probability $P(a \rightarrow a|a) = 0$.

The random walk proceeds from until either 1) an ambiguous name node is encountered or 2) a maximum number of steps are taken. In our studies, we limit the maximum number of steps to 50. After a certain number of random, we approximate the posterior probability of reaching b given the walk originated at a and the observed network, which is represented as $P(a \Rightarrow b)$. As depicted in Figure 2.B, the posterior probabilities remove the necessity for all network nodes except for the ambiguous names. The similarity between nodes a and b is set to the average of the probability of reaching a given b as a start node and vice versa, or $[P(a \Rightarrow b) + P(b \Rightarrow a)]$ / 2. This similarity score is then used in a single linkage clustering process, such that edges are removed if their similarity is below a threshold value. Each resulting components of the graph corresponds to a particular latent variable, or entity. The set of names for each component correspond to the names for a particular entity.

More complex schemes for measuring similarity are proposed in the discussion, but were not evaluated in this study.

3.3 F scores for Multi-class Accuracy. Given a clustering of names, we measuring the accuracy of the predictions through the F-score. This metric was initially introduced by the information retrieval community for testing the accuracy of clusters with greater than two predefined classes, such as the topics of webpages (e.g. baseball vs. football vs. tennis vs. etc..). [23] As applied to disambiguation, the F-score is mea-

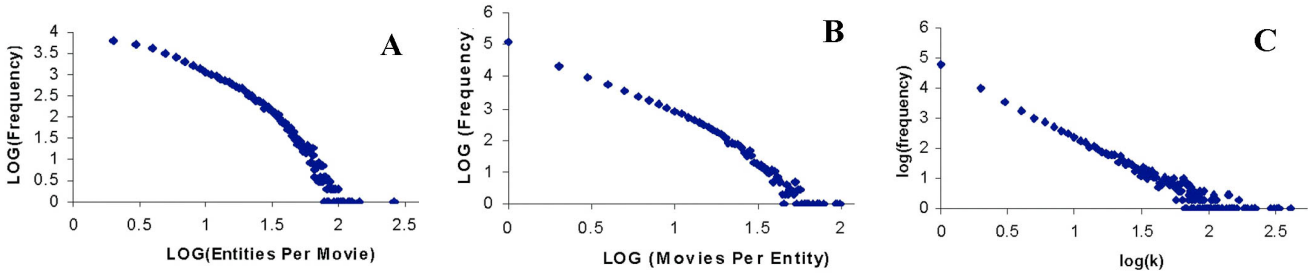

Figure 3: Summary statistics of entity, source, and name distributions in the IMDB. **A**) Log-log plot of movies per entity, **B**) log-log plot of entities per movie, and **C**) log-log plot of frequency of ambiguous name size.

sured as follows. Let $H_e = \{h_1, h_2, \ldots, h_m\}$ be the set of entities referenced by a specific name. Let $S_e = \{s_{e1}, s_{e2}, \ldots, s_{em}\}$ be a set of sets of sources, such that s_{ei} corresponds to the set of sources that entity h_i occurs in. For this research, we only consider sources which contain a single occurrence of an ambiguous name. Thus, for all $s_{ei}, s_{ej} \in S_e, s_{ei} \cap s_{ej} = \emptyset$. Now, let $C = \{c_1, \ldots, c_k\}$ be a set of clusters of the sources in S_e. Furthermore, let $T = \{t_1, \ldots, t_k\}$ be the set of sources for each cluster in C.

The F-score is a performance measure, which uses the harmonic mean of precision and recall statistics for a multi-class classification system. In information retrieval, recall R is defined as the fraction of known relevant documents which were retrieved by the system. In contrast, precision P is defined as the fraction of the retrieved documents which are relevant. For a specific class in the system, which is simply an entity, we define recall and precision for an arbitrary cluster as $R(e_i, c_j) = |s_i \cap t_j|/|s_i|$ and $R(e_i, c_j) = |s_i \cap t_j|/|t_j|$. The F-score for an arbitrary entity-cluster pair, $f(e_i, c_j)$, which is referred to as the local F score, is taken as the harmonic mean of the recall and precision:

$$f(e_i, c_j) = \frac{2R(e_i, c_j P(e_i, c_j)}{R(e_i, c_j + P(e_i, c_j)}$$

While the local F score provides fit for a single entity class and a single cluster, it is the complete system partitioning which we are interested in. To measure the accuracy of the complete system we compute a global F-score, which is basically the sum of the largest local F-scores for each entity class. More specifically, the global F score for an E, C pair is:

$$F(E, C) = \frac{\sum_{s \in S_e} |s| \max_{c \in C}(f(e,c))}{|\bigcup_{s \in S_e} s|}$$

For the methods evaluated in this paper the global F-score is used to test the goodness of fit for a clustering.

4 Experiments

In this section, the disambiguation methods of the previous section are evaluated on a real world dataset.

4.1 Data Description. The dataset chosen to evaluate the disambiguation strategies consists was the Internet Movie Database (IMDB). A publicly available dataset [17] was downloaded from the IMDB's ftp site and was parsed into a relational database for processing purposes. The database contains approximately 115 years worth of actor lists for movies, television shows, straight to video and dvd. For resolution purposes, the IMDB staff labels every entity uniquely, so even entities with ambiguous names are provided with unique primary IDs in the form of an appended roman numeral (i.e. John Doe (I) vs. John Doe (II)). As a result, the underlying truth of the data is known for validation purposes. For this study, this information is only taken into account after disambiguation.

A subset of the IMDB dataset was chosen for evaluation purposes. This subset covered the ten year period 1994-2003 and consists of all movies with greater than 1 actor. For completeness purposes, the following summary statistics were gathered. There are 37,000 movies and 180,000 distinct entities. The distribution of number of movies per actor is depicted in Figure 3.A, and it can be validated that it follows a log-log linear model, or power law distribution. The average number of entities per movie is 8 with a standard deviation of 9.9. Furthermore, it can be validated that in Figure 3.B that the number of entities per movie follows a similar trend. As noted by Barabasi and Albert, the degree distribution of the actor-to-actor network constructed from IMDB data follows a power law distribution as well. [5]

To construct a set of k-ambiguous names, entities were grouped by last name. There are 85,000 distinct last names. The distribution of number of entities per last name also follows a power law distribution,

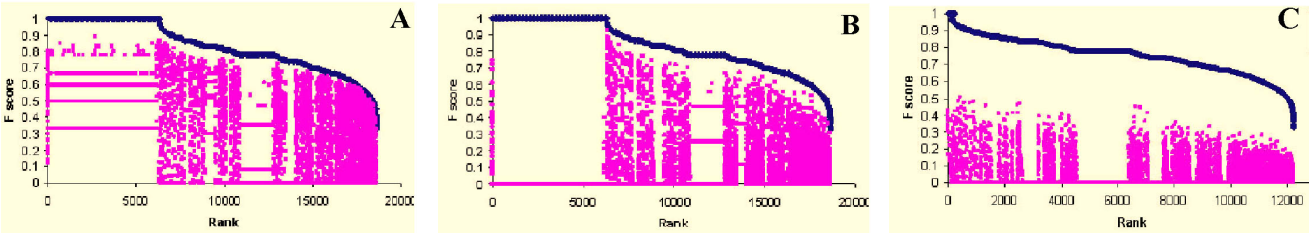

Figure 4: F-scores of hierarchical clustering of sources for each 2-ambiguous name. The topline corresponds to best F score observed during clustering. The plot below is the difference between the best F-score minus a baseline F-score of all sources as **A**) singletons and **B**) a single cluster. Image **C** depicts scores for names where the number of sources is greater than the number names. In this image, the baseline is the difference between the best F-score and the max F-score of both baselines.

as shown in image **C** of Figure 3. To put these numbers in perspective, there are approximately 12,000 2-ambiguous names.

4.2 Hierarchical Clustering Results. The IMDB dataset was subject to hierarchical clustering using the average linkage criteria described above. For clustering raw sources, we considered a continuum of similarity thresholds for stopping the clustering procedure. Figure 4 depicts the best global F-scores achieved for names from this dataset. The x-axis is ordered by number of entities per name, so 1-ambiguous names are on the left. The graph is then subordered by best observed F-score. The predicted F-scores were compared against several baseline methods. In Figure 4.A-C of , the upperline corresponds to the best observed F-score. In Figures 4.A and 4.B, the plot below the best score line corresponds to the difference between the best score and the baseline. The baseline method in Figure 4.A assumes all ambiguous names are distinct entities. In contrast, the baseline in Figure 4.B assumes all ambiguous names correspond to a single entity. These baselines are referred to as *AllSingletons* and *OneClusterOnly*, respectively. In 4, the first 70,000 points correspond to 1-ambiguous names, which explains is why the single cluster baseline predicts perfectly (i.e. F-score of 1).

To consider a more specific case where the baseline is not guaranteed to score perfectly, Figure 4.C depicts a disambiguation results for 2-ambiguous names, where the number of sources is greater than 2. In contrast to Figures 4.A and 4.B, the plot in 4.C presents the difference between the best F-score from hierarchical clustering and the maximum score achievable from a baseline method.

To an extent, the images of Figure 4 skew the clustering prediction results. Though Figure 4 implies that clustering provides F-scores above baseline scores, it must be taken into account that these are the best F-scores possible. The only way to discover the maximum F-score is to check the accuracy of each disambiguation prediction against the underlying truthful values. It is unfair to compare the power of hierarchal clustering to maximum F score of the baseline tests for similar reasons. Just as we cannot consider all partitions of the hierarchical clustering process simultaneously, we cannot simply take the max of both baselines - we must choose one or the other. In reality, an automated method must be able to find a point at which clustering automatically stops.

A simple method which was tested for automatic stopping was to average out the F-scores at various similarity threshold values. The resulting scores are demonstrated in Figure 5.A with the label "hc". In contrast to Figure 4, the average F-scores for all singletons and single cluster baselines are reported. The vertical line in the graph depicts one standard deviation around the average hierarchical clustering F-score. A threshold of 0 corresponds to the *OneClusterOnly* baseline and a threshold of 1 corresponds to the *AllSingletons* baseline. In Figure 4.A, as the threshold increases from 0 to 1, the F-score increases. The average F-score reaches a maximum value close to a similarity of 0.99, at which point the average F-score and all clusterings within 1 standard deviation achieve better than the best baseline of all singletons. This is very encouraging, except with such a high similarity threshold it is implied that we should only merge clusters with extremely high structural equivalence in their vectors. This is quite peculiar, and appears to be completely antithetical to the belief that community structures permit greater capability for disambiguation.

4.3 Random Walk Results. However, once we consider the results from the random walk clustering, the previous result appears to be less counter than initially implied. In the right plot of Figure 4, we present average

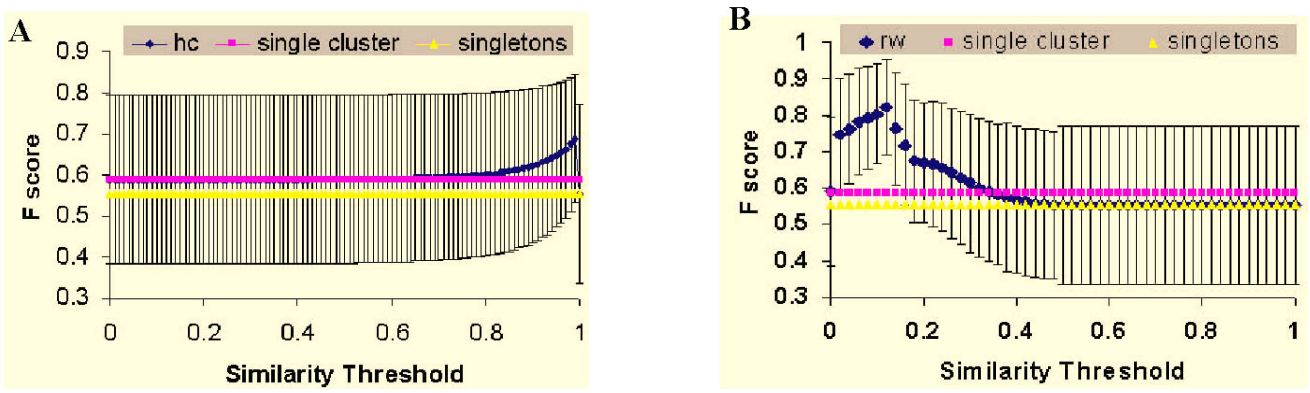

Figure 5: **A**) Average F-score of hierarchical clustering (hc), singletons, and single cluster baselines over continuum of cosine similarity threshold values. The vertical lines correspond to 1 standard deviation. **B**) Average F-score of random walk network partitioning, singletons, and single cluster baselines over continuum of cosine similarity threshold values. The vertical lines correspond to 1 standard deviation.

the F scores for random walk partitioning. There were 100 random walks initiated from each ambiguous node. Recall, similarity is actually the mean of the probability of walking between ambiguous name observations a and b within 50 steps. The graph is then thresholded, such that probabilities below the threshold are removed, and the resulting network components are set as the predicted clusters. From this plot, it is apparent that a maximum F-score is achieved at a relatively low threshold, specifically a probability of 0.12. Moreover, the average F score maximum at this point is greater that the maximum for simple hierarchical clustering by approximately 0.1. This is a significant improvement and supports the community structure hypothesis. Nodes and edges which are not directly related to the ambiguous names provide a significant amount of power for disambiguation purposes.

5 Discussion

The results of the previous section demonstrate community equivalence provides an advantage over exact equivalence for measuring similarity and, subsequently, disambiguation. While the datasets which these results are derived correspond to real world observations, the experiments and models of disambiguation are based on a highly controlled environment. Some of the limitations of this environment, and possibilities for extension are addressed in the following sections.

5.1 Building a Better Stopping Criteria. One limitation of this work stems from its dependency on a static threshold as a stopping criteria of the clustering process. This is a age old concern regarding hierarchical clustering and, for the most part, all stopping criteria are based on heuristics which are tailored to a researcher's respective environment. Airoldi and Malin have recently proposed a statistical test for stopping the clustering process based on geometric intuition regarding the growth rates of clusters. [1] In their research, clustering utilizes a single linkage criterion and thus has yet to be proven if such geometric insights hold for more complex clustering criteria such as the average linkage method employed for this paper's analysis. It is possible such tests could be adapted and in future research we hope to address this issue in more depth.

Though stopping criteria for hierarchical clustering may be difficult to define, it might be easier to derive an intuitive threshold for the random walk procedure. In this research, only similarity based on the probability of reaching one node from another was considered. However, this is an incomplete picture of the community surrounding an ambiguous name, and furthermore is a biased estimator. The information which random walks provide is much more substantial than the probability of reaching one node from another. In effect, there are at several additional features which can be accounted for to reduce bias in static thresholds. First of all, certain names are observed in more sources than other names. As a result, if the probability of reaching node b from node a is 0.2 and there are 20 sources in consideration, this is clearly a more probable occurrence than if the same probability was observed when 200 sources are considered.

Second, random walks provide the probability that a node will reach any node. Thus, we can consider the number of times a walk originating from an ambiguous node finds another ambiguous node, including itself, in the random walk. Note, there will be occurrences

when a random walk fails to find an ambiguous nodes. Such occurrences should not be discounted since they still communicate important indications of the distance between one ambiguous node and another. Thus, it is apparent that the probability $P(a \Rightarrow b)$ should be inversely correlated to the probability a node walks back to itself, or $P(a \Rightarrow a)$. Furthermore, we should negatively reweight if node b is a node which is reachable from many different nodes.

Third, the random walks were arbitrary specified to time out after 50 steps. By this construction, a walk completed successfully (i.e. reaches an ambiguous name node) in 2 steps is given equal weight in the similarity measure than a successful walk of 50 steps. It is possible that a discounting model may be more appropriate, such that as the number of steps increases, the score provided to a successful completion tends toward zero. In future research we expect to design more formal probabilistic representations of community similarity.

5.2 Towards More Realistic Models. In this paper, we introduced the concept of a k-ambiguous name. While there were almost 20,000 names with a k greater than 1, we controlled our clustering experiments to test on environments where the only uncertainty was associated with one particular name. Controlling for certainty is useful in the evaluation of the relative performance of disparate disambiguation procedures, but obviously this is an unrealistic assumption. In the real world, it is not clear if any observed name ever has complete certainty. This suggests that probabilistic models of certainty may be useful for disambiguating names when many names are ambiguous. For instance, expectation-maximization strategies over the graph are a potential route of research for resolution. [20, 21] With respect to this research, an extension to this research is to consider basic iterative methods, which can be used to cluster and classify relational data by leveraging names of high certainty, which can be fixed, or removed, during the learning process. By doing so, we can take advantage of high certainty knowledge to resolve lesser certain situations. We intend to investigate such models in future research.

Furthermore, as noted in previous works [5], the IMDB actor-to-actor network is variant of a random network with strong clustering features. In order to test disambiguation on a larger scale, we expect to test our methods on other types of social networks.

5.3 Making Search Engines More Social. Though there are limitations to the disambiguation research set forth in this paper, the results are promising and there exist potential applications for the next generation of search engines. This is especially so for search engines which archive and retrieve documents with large numbers of names. Clustering webpages based on their disambiguation properties can assist in making retrieval responses to queries more meaningful. Rather than rank pages by relevance using methods based on spectral decomposition properties, which are simply bag of words similarity comparisons, pages of relevance could be partitioned into clusters regarding the particular entities of interest. When results are displayed to the user, each ambiguous name could be qualified by key words extracted from the documents in the cluster. Obviously, this is speculation into an approach for search engines; nonetheless, the methods evaluated in this paper can provide a basis for future research and development of socially cognizant search engines.

6 Conclusions

This paper evaluated several methods for disambiguating names in a relational environment (actor collaborations in the Internet Movie Database) were presented. The first method was based on hierarchical clustering of sources in which ambiguous names are observed. The second method leveraged social networks constructed from all sources, such that random walks originating from ambiguous name nodes, were used to estimate posterior distributions of relations to partition the graph into components. We controlled social networks to study a single ambiguous name, and our findings suggest methods which leverage community, in contrast to exact, similarity provide more robust disambiguation capability. This research served as proof of concept for social network-based disambiguation, and in the future we will generalize our methods to account for networks that consist of more than one ambiguous names.

References

[1] E. Airoldi and B. Malin. Data mining challenges for electronic safety: the case of fraudulent intent detection in e-mails. In *Proc IEEE ICDM-2004 Workshop on Privacy and Security Aspects of Data Mining*. Brighton, England. 2004.

[2] R. Albert and A.L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*. 2002; 74: 47-97.

[3] A. Bagga and B. Baldwin. Entity-based cross-document coreferencing using the vector space model. In *Proc* 36^{th} *Annual Meeting of the Association for Computational Linguistics*. San Francisco, CA. 1998; 79-85.

[4] M. Banko and E. Brill. Scaling to very large corpora for natural language disambiguation. In *Proc* 39^{th}

Annual Meeting of the Association for Computational Linguistics. Toulouse, France. 2001.

[5] A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*. 1999; 286: 509-512.

[6] I. Bhattacharya and L. Getoor. Iterative record linkage for cleaning and integration. In *Proc* 9^{th} *ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. Paris, France. 2004; 11-18.

[7] I. Bhattacharya and L. Getoor. Deduplication and group detection using links. In *Proc 2004 ACM SIGKDD Workshop on Link Analysis and Group Detection*. Seattle, WA. 2004.

[8] E. Brill and P. Resnick. A rule-based approach to prepositional phrase attachment disambiguation. In *Proc* 15^{th} *International Conference on Computational Linguistics*. Kyoto, Japan. 1994; 1198-1204.

[9] P. Brown, S. Della Pietra, V. Della Pietra, R. Mercer. Word-sense disambiguation using statistical methods. In *Proc* 29^{th} *Annual Meeting of the Association for Computational Linguistics*. Berkeley, CA. 1991; 264-270.

[10] K.M. Carley, M. Dombroski, M. Tsvetovat, J. Reminga, and N. Kamneva. Destabilizing dynamic covert networks. In *Proc* 8^{th} *International Command and Control Research and Technology Symposium*. Washington, DC. 2000.

[11] S. Chan and J. Franklin. Symbolic connectionism in natural language disambiguation. *IEEE Transactions on Neural Networks*. 1998; 9(5): 739-755.

[12] G. Chao and M.G. Dyer. Word sense disambiguation of adjectives using probabilistic networks. In *Proc* 17^{th} *International Conference on Computational Linguistics*. Saarbrucken, Germany. 2000; 152-158.

[13] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification, 2nd Edition*. Wiley. New York. 2001.

[14] W.A. Gale, K.W. Church, and D. Yarowsky. A method for disambiguating word senses in large corpora. *Computers and Humanities*. 1992; 26: 415-439.

[15] F. Ginter, J. Boberg, J. Jarvinen, and T. Salakoski. New techniques for disambiguating in natural language and their application to biological text. *Journal of Machine Learning Research*. 2004; 5: 605-621.

[16] V. Hatzivassiloglou, P.A. Duboue, and A. Rzhetsky. Disambiguating proteins, genes, and RNA in text: A machine learning approach. *Bioinformatics*. 2001; 17:97-106.

[17] The Internet Movie Database. http://www.imdb.com.

[18] K. Hiro, H. Wu, and T. Furugori. Word-sense disambiguation with a corpus-based semantic network. *Journal of Quantitative Linguistics*. 1996; 3: 244-251.

[19] K. Jensen and J.L. Binot. Disambiguating prepositional phrase attachments by using on-line definitions. *Computational Linguistics*. 1987; 13(3-4): 251-260.

[20] D. Jensen and J. Neville. Iterative classification of relational data. *Papers of the AAAI-2000 Workshop on Learning Statistical Models From Relational Data*. AAAI Press. 2000.

[21] D. V. Kalashnikov and S. Mehrotra. A probabilistic model for entity disambiguation using relations. *Computer Science Department Technical Report TR-RESCUE-04-12*, University of California, Irvine. June 2004.

[22] J. Klienberg. The small-world phenomenon: An algorithmic perspective. In *Proc* 32^{nd} *Annual ACM Symposium on Theory of Computing*. Portland, OR. 2000.

[23] B. Larsen and C. Aone. Fast and effective text mining using linear-time document clustering. In *Proc* 5^{th} *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Diego, CA. 1999; 16-22.

[24] M. Lesk. Automatic sense disambiguation: How to tell a pine cone from an ice cream cone. In *Proc* 1986 *ACM SIGDOC Conference*. New York, NY. 1986; 24-26.

[25] G. Mann and D. Yarowsky. Unsupervised personal name disambiguation. In *Proc* 7^{th} *Conference on Computational Natural Language Learning*. Edmonton, Canada. 2003.

[26] M. Newman. The structure and function of complex networks. *SIAM Review*. 2003; 45, 167-256.

[27] H.T. Ng. Exemplar-based word sense disambiguation: Some recent improvements. In *Proc* 2^{nd} *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. Somerset, New Jersey. 1997; 208-213.

[28] H.T. Ng, B. Wang, and Y.S. Chan. Exploiting parallel texts for word sense disambiguation: an empirical study. In *Proc* 41^{nd} *Annual Meeting of the Association for Computational Linguistics*. Sapporo, Japan. 2003.

[29] E. Ravasz and A.L. Barabasi. Hierarchical organization in complex networks. *Phys. Rev. E.* 2003 ; 67, 026112.

[30] L. Sweeney. Finding lists of people on the Web. *ACM Computers and Society*. 2004; 34(1).

[31] V. Torvik, M. Weeber, D.W. Swanson, and N.R. Smalheiser. A probabilistic similarity metric for medline records: a model of author name disambiguation. *Journal of the American Society for Information Science and Technology*. 2004; 55(13): forthcoming.

[32] J. Vronis and N. Ide. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *Proc* 13^{th} *International Conference on Computational Linguistics*. Helsinki, Finland. 1999; 389-394.

[33] N. Wacholder, Y. Ravin, and M. Coi. Disambiguation of Proper Names in Text. In *Proc* 5^{th} *Applied Natural Language Processing Conference*. Washington, DC. 1997.

[34] D. Yarowsky. Word-sense disambiguation using statistical models of Roget's categories trained on large corpora. In *Proc* 30^{th} *Annual Meeting of the Association for Computational Linguistics*. Nantes, France. 1992; 454-460.