Linking Datasets on Organizations Using Half A Billion Open Collaborated Records

Anonymous Author(s)
Anonymous Institution(s)

April 12, 2022

Abstract

Scholars studying organizations would like to work with multiple datasets lacking shared unique identifiers or covariates. In such situations, researchers often turn to approximate string matching methods to combine datasets. String matching, although useful, faces fundamental challenges. String distance metrics are static, and do not explicitly maximize match probabilities. Moreover, many entities have multiple names that are dissimilar (e.g., "Fannie Mae" and "Federal National Mortgage Association"). This paper proposes two approaches to leveraging the massive amount of human-collaborated data from an employment-related social networking site (LinkedIn) to address this problem. The first approach builds a machine learning model for predicting matches, treating the trillion user-contributed organizational name pairs as a training corpus. The second approach uses community detection and treats user records as a network. We document substantial improvements over fuzzy matching in three organization name matching exercises. We make our methods available in an open-source R package (`LinkOrgs`).

Keywords Record-linkage · Interest groups · Natural language processing · Networks · Community detection

1 Record-linkage and Organizational Social Science

As large datasets on individual political behavior have become more common, scholars have focused increasing attention on the methodological problem of linking records from different sources (Enamorado, Fifield, and Imai 2019; Herzog, Scheuren, and Winkler 2010; Larsen and Rubin 2001). Record linkage methods are important because, through them, researchers can obtain outcome or covariate data about survey respondents, campaign contributions, or voters that would have been costly or impossible to obtain in previous eras (e.g. Ansolabehere and Hersh 2012; Figlio et al. 2014; Bolsen, Ferraro, and Miranda 2014; Hill and Huber 2017), to mention nothing of reduced bias and greater efficiency from the more successful linkage of records. When unique identifiers for observational units are unavailable for linking datasets directly, this recent literature has developed sophisticated algorithms that can find, with high probability, the same individual in two datasets using stable characteristics such as birth year and race (Enamorado, Fifield, and Imai 2019).

Despite this progress, these developments have had less of an impact so far on scholarship concerning organizational entities such as corporations, universities, trade associations, think tanks, religious groups, non-profits, and international associations—entities which are important players in theories of political economy (Wilson 1985), American politics (Nownes 2006), and other subfields. As with researchers on individuals, scholars of organizations also seek to combine multiple data streams to develop evidence-based models. However, in addition to lacking shared unique identifiers, datasets on organizations also often lack common covariate data that form the basis for probabilistic linkage algorithms. Therefore, scholars must rely heavily on exact or fuzzy string matching to link records on organizations.

To take a recent example from the *Ameican Political Science Review*, Crosson, Furnas, and Lorenz (2020) compare the ideology scores of organizations that have political action committees with ones that do not. The ideology scores are calculated from a dataset of interest group position taking compiled by a non-profit organization (Maplight), while the list of organizations with political action committees originates from Federal Election Commission (FEC) records. Maplight and the FEC do not refer to organizations using the same names and there is no covariate data to help with the linkage. The authors state that matching records in this situation is "challenging" (p. 32), and consider both exact and fuzzy matching. Ultimately, they perform exact matching on names after considerable pre-processing because of concerns about false positives, acknowledging that they inevitably do not link all records as a result. Indeed, the authors supplement the 545 algorithmic matches with 243 additional hand-matches, implying that their first algorithmic effort missed at least one in three correct matches.

The challenge faced by Crosson, Furnas, and Lorenz (2020) is typical for scholars studying organizations in the US or international context. Informally, our impression is that they have addressed it more effectively, and perhaps with greater transparency, than typical. Moreover, their example illustrates how string matching, although powerful, has fundamental limitations. Even though it can link records whose identifiers contain minor differences, string distance metrics do not explicitly maximize the match probability between organizational names. Moreover, string matching has trouble handling the diversity of monikers an organization may have. For example, "JPM" and "Chase Bank" refer to the same organization, yet these strings share no characters. Indeed, even *human coders* may struggle to connect records between some organizations. For example, string matching and research assistants may both fail to detect a relationship between Boalt Hall and Berkeley Law School. These problems are compounded when one attempts to link datasets from different source languages, as, for example, Chinese and English names will share *no* common characters. Research on organizations is hampered by these challenges connecting data sources.

Methodologists have begun to address the challenges that researchers face in matching organizational records. Mozer et al. (2020), for example, propose an adaptive learning algorithm that does many different kinds of fuzzy matching and uses a human-in-the-loop to "adapt" possible fuzzy matched data to the researcher's particular task. Yet the system still requires a researcher to do at least some amount of manual matching, and will also struggle to make connections where common characters are few and far between (e.g. Chase Bank and JPM). At the same time, there is an immense amount of data in the digital world about how people refer to organizations in different ways. In particular, users of social network sites, such as LinkedIn, invest time and energy discussing the wide range of organizations that serve as their employers—from small businesses to Fortune 500 companies to international organizations. This organizational networking data can provide considerable assistance for researchers who seek to link records on organizations, if only that data can be somehow processed.

In this paper, we develop methodology for linking organizational records that leverages abundant information available on networking sites such as LinkedIn. In particular, we develop two complimentary but distinct approaches to using half a billion open collaborated records. The first method is based on machine learning. The second is based on network analysis and community detection.

In our first approach, we treat organizational record linkage as a machine-learning problem and the open-collaborated records as a massive training corpus. We outline two assumptions that, once made, allow us to compute the match probability for any two organizational names. Using character-vector based representations for each text alias to avoid the problems due to abbreviations or spelling differences (e.g., between American and British English), we then construct a model that builds from the literature in natural language processing to predict the probability that two aliases refer to the same organization using the one trillion alias pair match/non-match training examples from the LinkedIn corpus.

The tool just described improves on fuzzy matching by explicitly modeling the match probability using the trillion training name pairs in the LinkedIn corpus, but suffers from the same problems as fuzzy matching in the context of cross-language matches or in the context of matching companies with multiple, disparate signifiers (such as the "Chase Bank", "JP Morgan" example). Therefore, we also develop an alternative approach that frames organizational linkage as a network analysis problem. In this approach, we use the database to construct a network of organizational aliases and URLs. In this formulation, "JPM" is one node in a graph, while "Chase Bank" is another. Relationships between nodes are also extracted from the data, for example because nodes share connections to the same organizational profile pages (e.g. `linkedin.com/company/chase`). A community of nodes with dense connections is likely to refer to the same organization.

Name	*Title*	*Organization*	*Organization URL*	*Start date*	*End date*
Michael Cohen	EVP & Special Counsel to Donald J. Trump	The Trump Organization	`linkedin.com/company/the-trump-organization`	20070501	20170418
Allen Weisselberg	EVP/CFO	The Trump Organization	`linkedin.com/company/the-trump-organization`		20170316
Michael Avenatti	Founding Partner	Eagan Avenatti, LLP		20070101	20170318
Michael Avenatti	Chairman	Tully's Coffee	`linkedin.com/company/tully%27s-coffee`	20120101	20170318
Michael Avenatti	Attorney	Greene Broillet & Wheeler, LLP		20030101	20070101
Michael Avenatti	Attorney	O'Melveny & Myers LLP	`linkedin.com/company/o%27melveny-&-myers-llp`	20000101	20030101

Table 1: Illustration of source data using three public figures.

Statistic	*Value*
# unique aliases	15,270,027
# unique URLs	5,950,995
Mean # of unique aliases per URL	2.88
Mean # of URL links per unique alias	1.12
Total # of alias pair examples	$> 10^{14}$

Table 2: Descriptive statistics for the LinkedIn network data.

If the algorithm is successful, "Chase Bank" and "JP Morgan" will find themselves placed in the same community. We apply two distinct community detection algorithms to alternative representations of this network, and use the detected communities to solve the record linkage problem, with significant performance gains over fuzzy matching.

Taken together, we propose several new tools which should allow researchers to better link datasets on organizations from over 100 countries and in dozens of languages. Intuitively, the approach uses the combined wisdom of millions of human beings with first-hand knowledge of these organizations. We illustrate the usefulness of this directory through three applications, two involving lobbying data and one involving a linkage task between English and Arabic, Chinese, and Japanese company names. An open-source package ("LinkOrgs") implements the computationally efficient methods we develop.

2 Social Networking Data as a Resource for Scholars of Organizational Politics

We first describe how human-generated records from social media companies contain a large amount of information that can potentially be useful to scholars of organizational politics. Although our approach will at times refer to specifics of our source data (LinkedIn) one may expect that other social networking site data would allow for similar approaches.

Users on LinkedIn provide a wealth of information about their current and previous employers. For the sake of our illustration, we will use a census of the LinkedIn network circa 2017, which acquired these records from a vendor, Datahut.co. Future researchers have the legal right to update this corpus using new "scrapes" , or comprehensive downloads, of the website: the Ninth Circuit Court of Appeals established in *HIQ Labs, Inc., v. LinkedIn Corporations* (2017) the right of firms to obtain and market such data. The dataset contains about 350 million mostly unique profiles drawn from over 200 countries—a size and coverage consistent with LinkedIn's internal estimates.

To construct a linkage directory for assisting dataset merges, we here use the professional experience category posted by users. In each profile on LinkedIn, a user may list the name of their employer as free-response text. We will refer to the free-response name (or "alias") associated with unit i as A_i. In this professional experience category, users also often post the URL link to their employer's LinkedIn page. In essence, the URL linkages serve as a kind of unique identifier for each organization. There are some complications we will have to deal with, however, as large internally differentiated organizations may sometimes have multiple valid profile URLs (e.g., ICPSR and the University of Michigan both have distinct organizational profile URLs on LinkedIn). Below, we provide examples of the professional experience data for several public figures, with the organizational alias used by these figures listed along with the profile URL to which they linked their profile.

We provide some descriptive statistics about the scope of the dataset as it relates to organizational aname usage in Table 2. The statistics reveal that, on average, users refer to each organization linguistically in about 3 different ways on average, and that most of the 15 million aliases link to just over one organizational URL.

We have described the human-contributed datasource that provides hundreds of millions of examples of people using organizational aliases, but how can this information be converted into something meaningful for social scientists seeking to link datasets on publicly traded firms, NGOs, or government agencies?

3 Modeling Organizational Linkage Using the LinkedIn Corpus

We now introduce two distinct ways in which researchers can conceptualize the record linkage problem using social neteworking data. In the first approach, the record linkage problem is framed as a prediction problem; machine learning tools are applied to numerical representations of the semantic content of the organizational names to learn the mapping between that content and the underlying probability that two aliases are matched to the same organization. In the second approach, the problem is framed in network terms: the task of record linkage in this framing is to find the community of aliases that all belong to the same community within the network of name-to-URL pairings.

3.1 Method 1: Organizational Record Linkage as Prediction Problem

We first discuss how the LinkedIn corpus can be used to linkage methods that seek to match organizational records by comparing the content in the name strings. We will use the linguistic features associated with one alias to form a match probability with another alias. Fuzzy string matching implicitly does this, placing a match probability of 1 between an alias and its nearest neighbor defined according to some similarity score based on the letters or letter combinations found in each alias (Li, Lu, and Lu 2008). Can we arrive at match probabilities in a more explicit way using the information encoded in the LinkedIn network? Can we learn the way that organizational names should be represented so as to maximize our ability to distinguish between matches and non-matches?

3.1.1 Setup and Assumptions

In this approach, we first assume that the alias-to-URL pairings are relevant for understanding alias-to-organization ties. That is, we assume validity:

$$\textit{Validity Assumption: } \Pr(O_i = O_j \mid A_i = a, A_j = a') = \Pr(U_i = U_j \mid A_i = a, A_j = a'), \tag{1}$$

where O_i, O_j refer to the organizations associated with index i and j, A_i, A_j refer to the aliases, and U_i, U_j refer to the URLs. This assumption states that the overall behavior of users linking their aliases to organizational URLs is the same as if they were linking to actual organizations. Intuitively, the assumption means that the LinkedIn name-to-URL patterns gives us information about the name-to-organization mapping.

Because, in organizational linkage tasks, the organizational unit of analysis is defined by the researcher, this assumption may be satisfied in some circumstances but not in others. Indeed, a match for one purpose may not be a match for others. For example, General Motors has divisions called Buick and Cadillac, which was originally a separate company. A researcher might want these entities to match each other in some dataset. Other researchers may not want these to match, however, and for plenty of purposes it may not make sense to make these entities connect. Therefore, whether alias-to-URL linkage is valid for a particular alias-to-organization match will depend on the way in which organizations are defined in the researchers' analysis.

Next, we assume semantic mapping, which states that the organizational match probability is a function of the semantic content of the aliases. That is,

$$\textit{Semantic Mapping Assumption: } \Pr(O_i = O_j \mid A_i, A_j) = f_p(f_n(A_i), f_n(A_j)), \tag{2}$$

where f_n represents a function that numericizes each alias string and f_p represents a function that transforms the alias string representations into overall match probabilities. This assumption, which depends on the existence of a single match function, is not trivial given the researcher-defined organizational unit of analysis. It also could be questionable for cross-language link tasks. Nevertheless, assuming that this function does existence, we will employ machine learning algorithms in an attempt to learn this function from the trillion or so LinkedIn organizational alias pairs.

Both the validity and semantic mapping assumptions would be vulnerable to errors in the training data. Regardless of how researchers define their organizational unit of analysis, inaccuracies due to the selection of incorrect URLs by users would limit our ability to capture the link probability using the alias and URL information and could, in principle, introduce systematic bias. However, as a professional networking site, users have an incentive to maintain accurate information about themselves on the page, hopefully limiting the degree of systematic bias.

Finally, to generate ground truth data for our prediction model training, we assume independence of indices i and j, which allows us to calculate $\Pr(O_i = O_j \mid A_i, A_j)$. This independence assumption would be violated if, for example, users could coordinate so that one use of an alias linking to a given URL gave additional information about another use. Given the wide geographic reach of the LinkedIn network, this sort of coordination is presumably rare, yet is difficult to evaluate for all pairs of indices. The assumption in any case is critical because, using it, we can calculate the overall organizational link probability:

$$
\begin{aligned}
\Pr(O_i = O_j \mid A_i = a, A_j = a') &= \Pr(U_i = U_j \mid A_i = a, A_j = a') \ \ \text{(by } \textit{Validity}\text{)} && (3)\\
&= \sum_{u \in \mathcal{U}} \Pr(U_i = u, U_j = u \mid A_i = a, A_j = a') && (4)\\
&= \sum_{u \in \mathcal{U}} \Pr(U_i = u \mid A_i = a) \Pr(U_j = u \mid A_j = a') \ \ \text{(by } \textit{Independence}\text{)} && (5)
\end{aligned}
$$

The two terms in the final expression can be calculated using the LinkedIn network using empirical frequencies (e.g., using the fraction of times alias a and a' link to u). LinkedIn corpus, as it contains 15 million unique aliases, contains over one trillion training examples of $\Pr(O_i = O_j \mid A_i = a, A_j = a')$ that we will use in training the name match prediction model described in the next section.

3.1.2 Our Modeling Approach

The semantic mapping assumption states that the overall organizational match probability is determined by two functions—one that numericizes the organizational name (f_n) and another that determines the match probability given the numerical representation of two names (f_p). Modeling assumptions about these functions is important because these assumptions determine (1) estimation performance in determining high-quality match estimates and (2) computational performance in applying the proposed method to large databases of interest to applied researchers.

In our modeling strategy, we assume the following functional form for the match probabilities in an effort to satisfy these two criteria:

$$
\textit{Distance-to-Probability Mapping:}\ \log\left(\frac{\Pr(O_i = O_j \mid A_i = a, A_j = a')}{1 - \Pr(O_i = O_j \mid A_i = a, A_j = a')}\right) \propto -||f_n(a) - f_n(a')||_2.
$$

In other words, the match probability is a function of the Euclidean distance between the numerical representations for aliases a and a'. This captures the intuition that intuitively close names like "apple" and "apple inc." are more likely to be matched than distant names like "apple" and "jp morgan".

Besides this intuition, this distance-to-probability mapping also has important computational benefits: if we seek to calculate the match probability between two datasets of 100 observations each, we do not have to apply a computationally expensive non-linear f_p function to all 5,050 possible pairs. Instead, we only need to execute a non-linear f_n function 100 times—once for each alias—after which we take the Euclidean distance between each pair of numerical representations in a step requiring little computational effort even for large data merge tasks.

Instead of making assumptions about how the organizational names should be numerically represented, we learn this representation from data. The specific modeling approach for generating each alias's numerical representation is motivated by recent advances in natural language sequence modeling. While we experimented with several possibilities (e.g., using Vaswani et al. (2017)) and are eager to see other approaches explored to further improve performance, the approach we employ here builds from the success of vector representations for words popularized in Mikolov et al. (2013). Word vectors enabled improved performance in natural language processing tasks (Egger 2022) in part because they overcome the sparsity issue present in other text representations (such as word counts): most documents do not use most words, but if each word is a vector of, say, 50 dimensions, information on every dimension is present for every document.

We generalize this insight, modeling each *character* as a vector (with each dimension representing some latent quality of that character), each *word* as a summary of some time series representation of character vectors, and each *organizational name* as a summary of some time series representation of learned word vectors. To model how the time series dynamics of character and word representations, we employ two Long Short-Term Memory (LSTM) neural network architectures (Sundermeyer, Schlüter, and Ney 2012): the idea

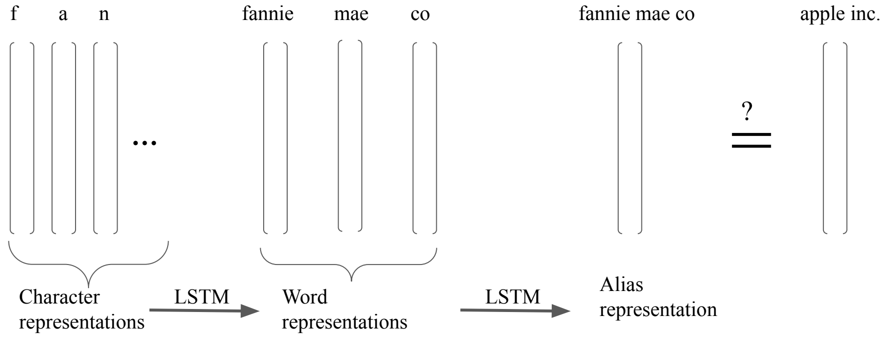

Figure 1: High-level illustration of the modeling strategy. We learn from data how to represent (a) the characters that make up words, (b) the words that make up organizational names, and (c) how to compare two organizational name representations.

is to "read" the sequence of characters (and then the sequence of words), where "read" here means combining information together non-linearly through a form of recursive non-linear regression. Once we have obtained the final representation for two aliases after "reading" the sequence of sequences, we form the overall match probability taking the normalized Euclidean distance between those pairs and applying a univariate logistic regression model that maps each distance to a number between 0 and 1. Figure 1 illustrates the modeling approach; additional details are found in Section 1.1 of the Appendix.

The latent vectors are model parameters, which are jointly updated using stochastic gradient descent to minimize prediction error on our training data computed using the LinkedIn corpus. Specifically, we minimize the KL divergence between the true match probability, $\Pr(O_i = O_j | A_i = a, A_j = a')$, and the estimated probability, where the KL divergence between two distributions, P and Q, denoted by $D_{\text{KL}}(P||Q)$, is a measure of how much information Q contains that P does not and is 0 when the two distributions contain the same information. That is, we minimize:

$$\text{minimize} \sum_{a,a' \in \mathcal{A}} D_{\text{KL}} \left(\widehat{\Pr}(O_i = O_j | A_i = a, A_j = a') \;||\; \Pr(O_i = O_j | A_i = a, A_j = a') \right), \tag{6}$$

where $\mathcal{A}$ denotes the set of all aliases. In this way, we learn from data how the aliases should be numerically represented—learning along the way the representation of characters, words, and how these things combine together in a name—with the overall goal of predicting the organizational link probability betwen each alias pair. We apply the approach here jointly to the entire LinkedIn corpus so that name representations in many languages are learned together.

3.1.3 Model Illustration

In Figure 2, we examine the output of the machine learning model. In the left panel of Figure 2, we see the distributions of match probabilities for non-matches and matches for an out-of-sample set of data points. The Kolmogorov-Smirnov (KS) test for the quality of the two distributions yield a test statistic of 0.83. If there were *no* overlap between the match/non-match distributions (so that match/non-match distance distributions were indistinguishable), the test statistic would be 0. If we could perfectly distinguish matches from non-matches, the test statistic would be 1. We are far closer to this second possibility, an indication of good performance.

In the right panel, we see pairs of aliases representing the same URL (pairs are represnted by the same graphical mark type). We project the 256-dimensional alias embedding space down to 2 dimensions via Principal Components Analysis (PCA) for plotting purposes. We see that aliases representing the same organization are generally quite close in this embedding space. For example, aliases "apple" and "apple co"

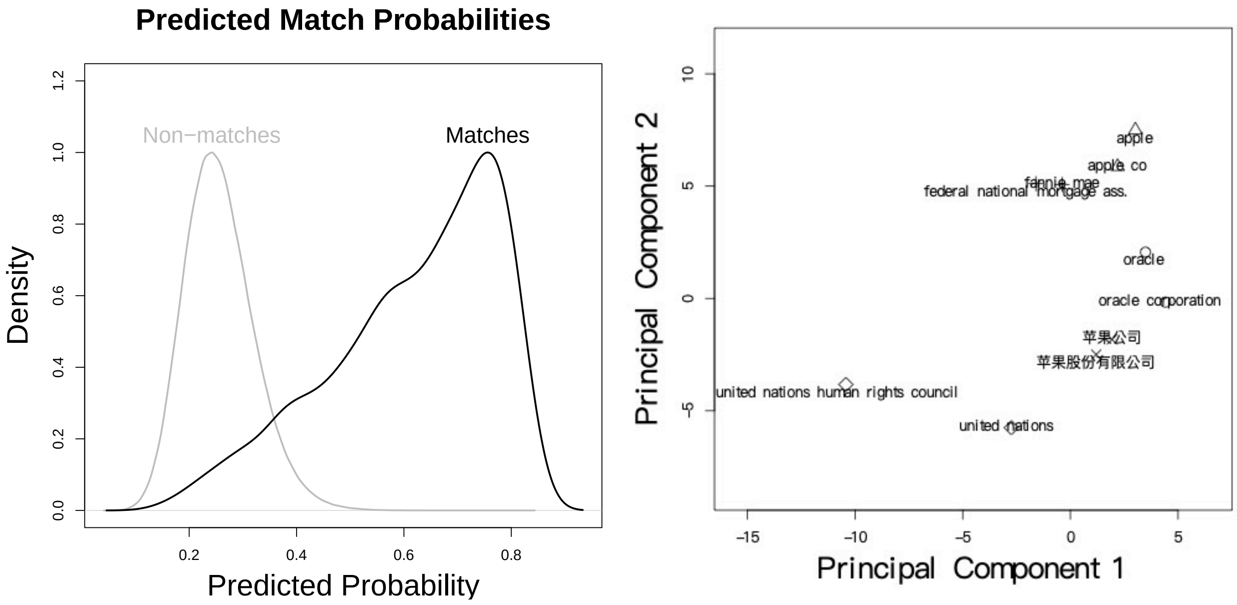

Figure 2: Visualizing the machine learning model output. The left panel shows how, on average, organizational alias matches have higher match probabilities compared with the set of all possible non-matches. The left panel shows how similar organizational names are close in this machine-learning generated vector space (which has been projected to 2-dimensions using PCA).

are near each other, as the two aliases are linguistically similar do indeed refer to the same organization. The model seems to be able to handle irrelevant linguistic information reasonably well: “oracle” and “oracle corporation” are quite close in this embedding space even though the presence of the long word “corporation” would substantially affect string distance measures based only on the presence or absence of letters or letter combinations. Also, two Chinese names for Apple Inc. (“苹果公司” and “苹果股份有限公司”) are quite close in this embedding space as well (indicating high match probability), as the model estimates match probabilities between organizational names across the many languages used on LinkedIn.

Despite these successes, there are true links which would remain hard to model using this prediction-oriented framework. For instance, the aliases “chase bank” and “jp morgan” have a low match probability, and cross language merge tasks still would struggle in the prediction framework due to the relative paucity of cross-language examples on the LinkedIn network. To handle these difficult linkage problems, we will need to frame the linkage problem in a new way—not in terms of prediction using lingusitic features, but, rather, as a community detection problem which can be solved using the network dynamics observed in the LinkedIn corpus.

3.2 Method 2: Organizational Record Linkage as Community Detection

When the semantic mapping assumption described in Equation 2 is violated, it is not feasible to predict organizational links using linguistic information in the alias names. However, we can still proceed by framing the linkage problem in terms of network-based community detection. This approach will allow us to weaken the semantic mapping assumption while exploiting other information present in the LinkedIn network.

In this second approach to linking organizations using the LinkedIn corpus, organizational aliases are nodes. Aliases may be connected by edges, which may be directed or undirected. A cluster of nodes with relatively dense connections is a “community” of aliases. Instead of viewing the record linkage tasks as matching two lists of organization directly, one can instead view the problem as placing a list of organization names into the organizational community in which it belongs. Implicitly, this approach assumes that each real organization has one community of aliases and one community of aliases identifies a discrete organization.

Because community detection is a well-studied problem that occurs in many different contexts (Rohe, Chatterjee, and Yu 2011), we consider two algorithms well-established in the literature: (a) Markov clustering algorithm (Van Dongen 2008) and (b) greedy clustering (Clauset, Newman, and Moore 2004). We focus on these algorithms because they model the network in two distinct ways and are computationally efficient.

3.2.1 Community Detection in the Probabilistic Network Representation

Probabilistic Network Representation Our first approach towards organizational alias community detection uses Markov clustering (Van Dongen 2008). The Markov clustering algorithm operates on weighted adjacency matrices that have a probabilistic interpretation. We thus begin our discussion with this network representation.

Using the same notation as in the previous section, we denote the organization names in the source data as aliases a. These aliases constitute nodes on our graph. As in Table 1, aliases appear simultaneously with profile URLs in our source data. One possible approach to defining links between aliases is to assume that two aliases are connected whenever they both link to the same profile URL. However, this approach would place as strong a connection on aliases that rarely link to the same profile URL as those that are frequently connected. For example, the URL `linkedin.com/company/university-of-michigan` has been associated with an alias 75,462 distinct times in our data. The vast majority of these connections occur via the alias University of Michigan, but a link with the alias Michigan State University does in fact occur a handful of times. In defining links on the network, it will help to up weight the former but down weight the latter in a data-driven way.

While there are many possible ways to ensure more frequently made connections are weighted higher than connections that are rarer, we adopt an approach inspired by naive Bayesian classification methods. We define the edge weight, $w_{aa'}$, between node a and a' using Equation 7:

$$w_{aa'} = \sum_{u \in \mathcal{U}} \frac{\text{\# of times } a \text{ co-occurs with } u}{\text{\# of times } u \text{ co-occurs with any alias}} \times \frac{\text{\# of times } u \text{ co-occurs with alias } a'}{\text{\# of times } a' \text{ occurs with any profile URL}} \tag{7}$$

These edge weights then make up the full adjacency matrix, $\mathbf{W}$, which captures the interrelationship between all pairs of aliases.

The connection of this expression to naive Bayesian classifiers requires some elaboration. As a pure formalism, one can write the law of total probability as

$$\Pr(A_i = a \mid A_j = a') = \sum_{u \in \mathcal{U}} \Pr(A_i = a \mid U = u, A_j = a') \Pr(U = u | A_j = a') \tag{8}$$

By asserting the equation as a formalism, we seek to skirt questions about what exactly the probability of A_i given A_j means in this context, which might distract us from the core task of showing how community detection methods on networks can be useful for record linkage problems. Taking this equation as given, then, suppose further one were to "naively" assume conditional independence of A_i and A_j given U (Rish et al. 2001), or more formally $\Pr(A_i = a \mid U = u, A_j = a') = \Pr(A_i = a \mid U = u)$, similar to what we assume in the independence assumption of Section 3.1.2. Equation 7 is then what would result if one also replaced the implied probabilities with sample counts.

There are several benefits of using this formula, especially over the more obvious "shared link" approach. Besides using much more of the data than the shared connection approach, the naive Bayesian calculation reweights edges in ways that are proportionate to the actual prevalence of relationships found in the data. One corollary is that the edge weights are asymmetric and the specificity with which an alias and link is shared matters a great deal. For example, ICPSR is a rare alias but when it does occur, it is very often is entered with `linkedin.com/company/university-of-michigan`, which in turn usually points to the University of Michigan alias. Therefore, the weight $w_{\text{University of Michigan,ICPSR}}$ will be close to 1 even if $w_{\text{ICPSR,University of Michigan}}$ is very small. By contrast, since Michigan State University and University of Michigan are both relatively common aliases, one spurious link between them will not result in large weight in either direction.

Detecting Communities in the Probabilistic Network Representation Having built $\mathbf{W}$, the probabilistic network of alias-to-alias links, we next apply the Markov clustering algorithm to it to find communities of aliases that tend to link to the same URLs.

Briefly, this clustering algorithm proceeds in two steps—expansion and inflation—that are repeated until convergence. In the expansion step, the network's adjacency matrix is multiplied by itself k times. This matrix multiplication simulates the diffusion of a Markov process on the nodes (i.e. "traveling" k steps on the

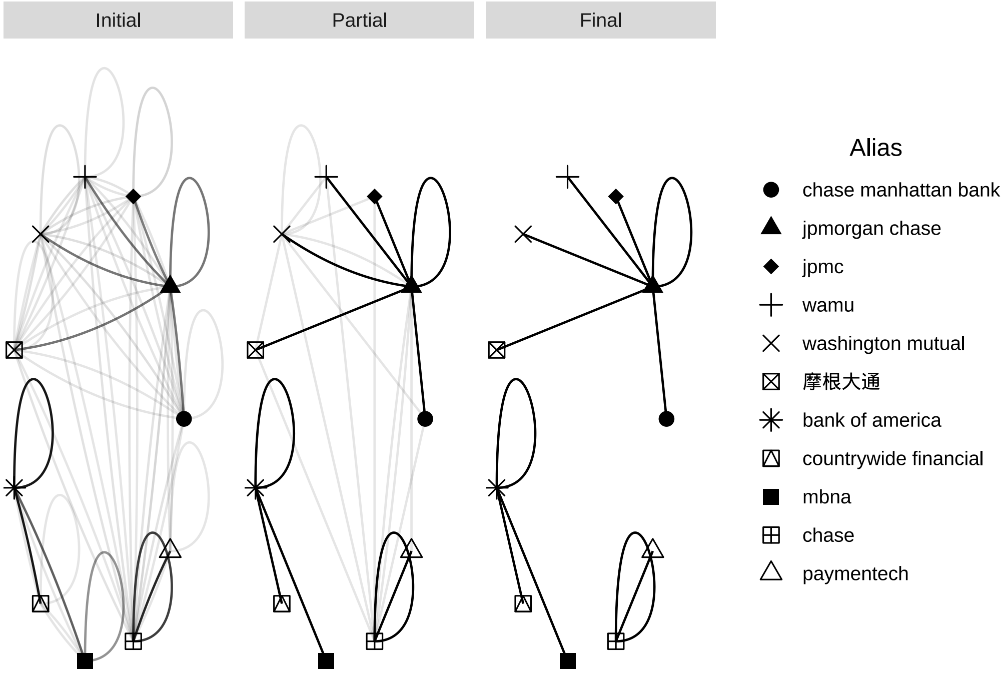

Figure 3: Illustration of Markov clustering.

network, with probabilities of where to go defined by $\mathbf{W}$). In the inflation step, entries of the resulting matrix are raised to some power p, and the matrix is renormalized into a valid Markov transition matrix. Since small probabilities shrink faster under exponentiation than large probabilities, the inflation step causes higher probability states to stand out (i.e. likely places are made even more likely). After alternating expansion and inflation, the output converges: row i will have only one non-zero valued entry in column j, which defines the representative node in the community. All rows which have a one in column j are a part of the same community of alias linking to (hopefully) the same organizational entity.

This clustering process involves hyperparameters, in particular, the number of matrix multiplications to do in the expansion step (k) and the power of exponentiation (p) in the inflation step. For both steps, k and p are frequently set to be 2. The total number of clusters is not explicitly specified but instead controlled indirectly through p and k. We adopt the common choice of $p = k = 2$ because it did not greatly influence the results.

Figure 3 illustrates the process on a subset of our data. Darker shades reflect heavier weights in the initial transition matrix. Some links are much stronger than others. In the initial weighting, two cliques reflect a set of names associated with "jp morgan chase." Another reflects names associated with "bank of america." However, these initial links are dense, making it difficult to distinguish one cluster of aliases from another. As the algorithm iterates, some links weaken and disappear while others strengthen. Eventually, each node links to exactly one other node. Notably, the final cliques contain lexographically dissimilar nodes that do indeed belong in the same cluster. For example, the "chase" clique contains "wamu", "paymenttech", and "摩根大通" which are all Chase affiliates. The "bank of america" clique includes "countrywide financial" and "mbna," both under the Bank of America umbrella. In this way, the semantic mapping assumption from Section 3.1.2 has been weakened and non-lexical information has been exploited to estimate organizational matches.

3.2.2 Community Detection in a Bipartite Network Representation

Although the network formulation just described has the benefit of a probabilistic interpretation, it involves processing the data with some assumptions. Here, we offer a second community detection approach while requiring no extra processing of the LinkedIn network data. Even so, it will turn out that this second clustering approach will perform similarly on the record linkage task using real data.

Here, we shall assume that both organization names and profile URLs are "aliases" for a real organization, which is synonymous with a community of aliases. Organization names do not link directly to one another, rather organization names are associated with URLs. Therefore, these associations can be understood as representing a bipartite network: we have alias nodes and URL nodes and undirected links between aliases and URLs represent weighted edges (with the weights being the number of times alias a is linked to URL u).

In this bipartite network formulation, linking datasets using name aliases is equivalent to finding the best alias-URL community structure, where aliases within community k all tend to link to URLs within that community too. The final clustering of aliases then provides a directory that researchers can use for determinig whether alias a and a' refer to the same organization or not.

There are many possible criteria for quantifying the quality of one clustering or another. We rely on previous literature, particularly Clauset, Newman, and Moore (2004), and use a criterion, called a modularity score, that (a) is 0 when community ties between aliases and URLs occur as if communities were assigned randomly and (b) gets larger when the proposed community structure places aliases that tend to link to the same URLs in the same community. Details are left to Section 1.2 in the Appendix. Here, the main hyperparameter is the stopping criterion for the greedy optimization. Like in the Markov clustering approach, the number of clusters is not itself a hyperparameter but instead the byproduct of the optimization and this stopping threshold.

4 Three Illustration Tasks Using Campaign Contribution, Lobbying, and Cross-language Data

4.1 Method Evaluation

Before we describe our three illustration tasks, we first need to introduce our competitive baseline and our evaluation metrics which we will use to understand the relative performace of the proposed methods and when applied researchers may want to consider them.

4.1.1 Fuzzy String Matching Baseline

We examine performance of our proposed methods against fuzzy string matching — which we believe to be the most commonly used method for merging organizational datasets — in order to establish a baseline. Fuzzy matching is a procedure that calculates string dissimilarities between all pairs of names in two datasets. Two entries are declared a match if the dissimilarity is below an acceptable threshold. While there are many ways to calculate dissimilarity, we focus on fuzzy string matching using the Jaccard distance measure in order to keep the number of comparisons reasonably small. Other string discrepancy measures, such as the cosine distance, produce similar results. Details are left to Section 1.3 in the Appendix.

4.1.2 Performance Metrics

We consider two measures of performance for each organizational matching algorithm. We will examine these measures as we vary the distance acceptance threshold for constituting a matching between two aliases.[1]

First, we examine the fraction of true matches discovered as we vary the acceptance threshold. This value is defined to be

$$\text{True positive rate} = \frac{\text{\# of true positives found}}{\text{Total \# of true positives}} \tag{9}$$

This measure is important because, in some cases, researchers may have the willingness or ability to manually evaluate the set of proposed matches, rejecting false positive matches. The true positive rate is therefore more relevant for the setting where scholars use an automated method as an initial processing step and then evaluate the resulting matches themselves, as may occur for smaller match tasks.

While the true positive rate captures our ability to find true matches, it does not weigh the cost involved in deciding between true positives and false positives. Therefore, we also examine a measure which considers the presence of true positives, false positives, and false negatives known as the F_β score (Lever 2016). This

[1] Results for the network-based linkage approaches also vary with this parameter because we first match aliases with entries in the directory in order to find the position of those aliases within community structure of the LinkedIn network.

score is defined as

$$F_\beta = \frac{(1+\beta^2)\cdot \text{true positive}}{(1+\beta^2)\cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}} \tag{10}$$

In the best case scenario, the F_β score is 1, which occurs when all true matches are found, with no false negatives/positives. In the worst case scenario, the score approaches 0, which occurs when few true positives are found or many false negatives/positives. The parameter β controls the relative costs of false negatives as compared with false positives. If β>1, false negatives are regarded as less costly than false positives; if β<1 then the reverse. In the matching context, errors of *inclusion* are typically less costly than errors of *exclusion*: the list of succesful matches is usually shorter and easier to double-check than the list of non-matched pairs. Therefore, we focus use the F_2 score, a common choice in evaluation tasks (e.g., Devarriya et al. (2020)). In contrast to the true positive rate, the F_2 measure is more relevant for the setting where the dataset to be merged are so large that scholars cannot reasonbly evaluate the resulting matches manually.

4.2 Task 1: Matching Performance on a Lobbying Dataset

We first illustrate the use of the organizational directory on a record linkage task involving lobbying and the stock market. Libgober (2020) shows that firms that meet with regulators tend to receive positive returns in the stock market after the regulator announces the policies on which they lobbied. These returns are significantly higher than the positive returns experienced by market competitors and firms that send regulators written correspondence. Matching of meeting logs to stock market tickers is burdensome because there are almost 700 distinct organization names described in the logs and around 7000 public companies. Manual matching typically involves research on these 700 entities using tools such as Google Finance. While the burden of researching seven hundred organizations in this fashion is not enormous, Libgober (2020) only considers meetings with one regulator. If one were to increase the scope to cover more agencies, or lobbying in Congress, the burden could become insurmountable.

We show how the incorporation of our directory into the matching process can improve performance, treating the human coded matches in Libgober (2020) as ground truth. Figure 4 shows that the three directory-based approaches almost always yield higher F_2 scores and true positives across the range of acceptance thresholds. The highest F_2 score is over 0.5, something achieved both by the machine-learning approach that uses the LinkedIn data as a massive training corpus for semantic learning and the bipartite community detection algorithm. Here, community detection algorithm using bipartite network achieves superior performance to the Markov clustering approach. The percentage gain in performance of the three LinkedIn-based approaches is higher when the acceptance threshold is closer 0, as we increase the threshold so that the matched dataset is 10 or more times larger than the truth matched dataset, the F_2 score for all algorithms approaches 0 and the true positive rate approaches 1. Overall, this illustrates how our methods would appear to yield superior performance compared to a leading alternative method in the common use case when researchers do not have access to shared covariates across organizational datasets.

It is also instructive to consider an example from this linkage task where fuzzy matching failed but the LinkIn approach, for example, the Markov clustering method, had success. Fuzzy matching fails to link the organizational log entry associated with “HSBC Holdings PLC” to the stock market data associated with “HSBC.” Their fuzzy string distance is 0.57, which is much higher than the distance of “HSBC Holdings PLC” to its fuzzy match (0.13 for “AMC Entertainment Holdings, Inc.”). “HSBC Holdings PLC”, however, has an exact match in the LinkedIn-based directory, so that the two organizations are successfully paired using the patterns learned from the LinkedIn corpus.

4.3 Task 2: Matching English and Non-English Company Names

In our next evaluation task, we examine the performance of our approach in a particularly challenging case where we seek to match organizational entities using their names written in English and in Mandarin Chinese. This match task is especially challenging because English and Mandarin Chinese are based on two entirely different kinds of writing systems (i.e. alphabetic and logographic).

The data for this task is from FluentU, an organization focusing on language learning (“FluentU” 2022). The organization has provided a directory of 76 companies with their English names paired with their Chinese names. Table 3 displays some of the companies found in this dataset, which is composed primarily of large multi-national corporations.

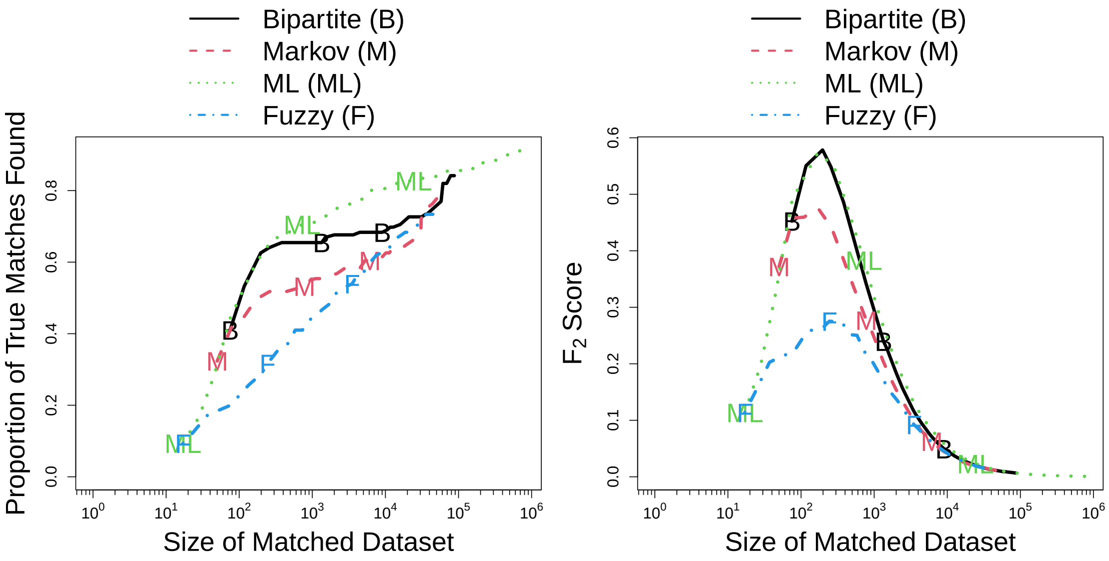

Figure 4: We find that dataset linkage using any one of the three approaches using the LinkedIn network obtains superior performance relative to fuzzy string matching both when examining only the raw percentage of correct matches obtained (left panel) and when adjusting for the rate of false positives and false negatives in the F_2 score (right panel).

English Name	Chinese Name
amazon	亚马逊
coca cola	可口可乐
jp morgan	摩根
marlboro	万宝路

Table 3: A sample of the the organizational entities in the cross-language dataset. We attempt to match the pool of English names to their associated names in Mandarin Chinese.

We compare linkage performance using our community detection algorithms based on the LinkedIn network and using our character-based machine learning and the baseline fuzzy matching approach. The fuzzy matching approach gives *no* matches (except when the fuzzy distance threshold is set to 1 so that all English names are matched with all Chinese names). This occurs because none of the English names share any Latin character combinations with any of the Chinese names.

The machine learning linkage approach based on the LinkedIn network fars somewhat better—yielding a true positive rate of 0.2 when the matched dataset size approaches 1,000. In contrast to fuzzy matching, it also achieves a non-zero F_2 score. However, the community detection approaches using the LinkedIn network fare better, successfully picking out 33 of the 76 matches with only 5 false positives. The left panel of Figure 5 shows how the Markov clustering approach somewhat outperforms the bipartite clustering, although both approaches achieve similar maximum true positive rates and F_2 scores just above 0.4. In this way, the network information present in the LinkedIn corpus has allows us to find almost half of the true matches, even though the names to be matched are written in entirely different language systems.

Still, it is once again instructive to examine names for which the community detection approaches succeeds and fails in this cross-language matching task. The approach succeeds in cases where organization has employees which use both the English and non-English organizational names while linking to the same company URL. For example, we successfully match the "Coca-cola" to "可口可乐" match, as Coca-cola has at least six Chinese offices and employees in these offices often use Coca-cola's Chinese name in their LinkedIn URL. However, we do not find the "lamborghini" to "兰博基尼" match. Lamborghini does operate offices in China, but its employees use the name "Lamborghini" in their LinkedIn profiles. In this way, the community-detection algorithms based on the LinkedIn network are, for cross-language merge tasks, best

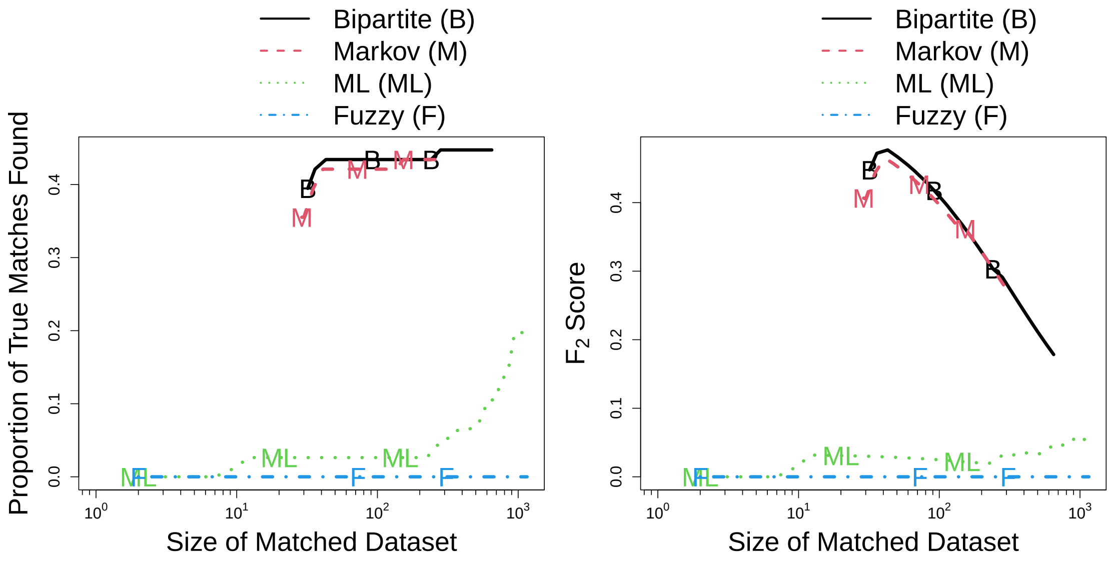

Figure 5: In the left panel, we see that the LinkedIn-based community detection algorithms finds a significant fraction of the true positive matches between the English and non-English aliases. In the right panel, we see that these true positive matches are found without introducing numerous false positives (so that the F_2 scores for the community detection algorithms approach 0.5). The machine learning approach yields some true positive matches; fuzzy matching yields none.

suited for the matching of organizational entities which have many employees across linguistic contexts and where non-English names are commonly used on LinkedIn.

4.4 Task 3: Linking Financial Returns and Campaign Contributions from Fortune 1000 Companies

In this final evaluation exercise, we focus on a substantive question drawn from the study of organizational lobbying: how do a company's economic resources correlate with political contribution behavior? Based on prior research (Chen, Parsley, and Yang 2015), we should expect a positive association between company size and lobbying activity: larger firms have more resources that they can use in lobbying, perhaps further increasing their performance (Ridge, Ingram, and Hill 2017; Eun and Lee 2021).

For this exercise, we use firm-level data on the total dollar amount spent between 2013-2018 on lobbying activity. This data has been collected by Open Secrets, a non-profit organization focused on improving access to publicly available federal campaign contributions and lobbying data ("Open Secrets" 2022). We match this firm-level data to the Fortune 1000 dataset on the largest 1000 US companies ("Fortune" 2022), where the data point of interest is the mean total company assets between the 2013-2018 period. The key linkage variable will be organizational names that are present in the two datasets. We manually obtained hand-coded matches to provide ground truth data.

In Figure 6, we explore the substantive implications of different matching choices, that is, how researchers' conclusions may be affected by the quality of organizational matches. In the left panel, we see that the true coefficient relating log organizational assets to log campaign contributions is about 2.5. In the dataset constructed using fuzzy matching, this coefficient is underestimated by about half. The situation is better for the datasets constructed using the three LinkedIn-based approaches, with the effect estimates being closer to the true value. Notice that, for all algorithms examined in the left panel of 6, there is significant attentuation bias in the estimated coefficient towards 0 as we greatly increase the size of the matched dataset. The inclusion of poor quality matches injects random noise into estimation, biasing the coefficient severely towards 0. Overall, we see from the right panel that match quality depends on algorithm choice as well as string distance threshold, with encouraging results for the approaches proposed here.

The right panel of Figure 6 presents a kind of plot that we would recommend researchers present when using a string matching algorithm before an analysis. For choice of matching algorithm, the figure presents the sequence of estimated coefficients as the size of the matched dataset varies with the distance threshold.

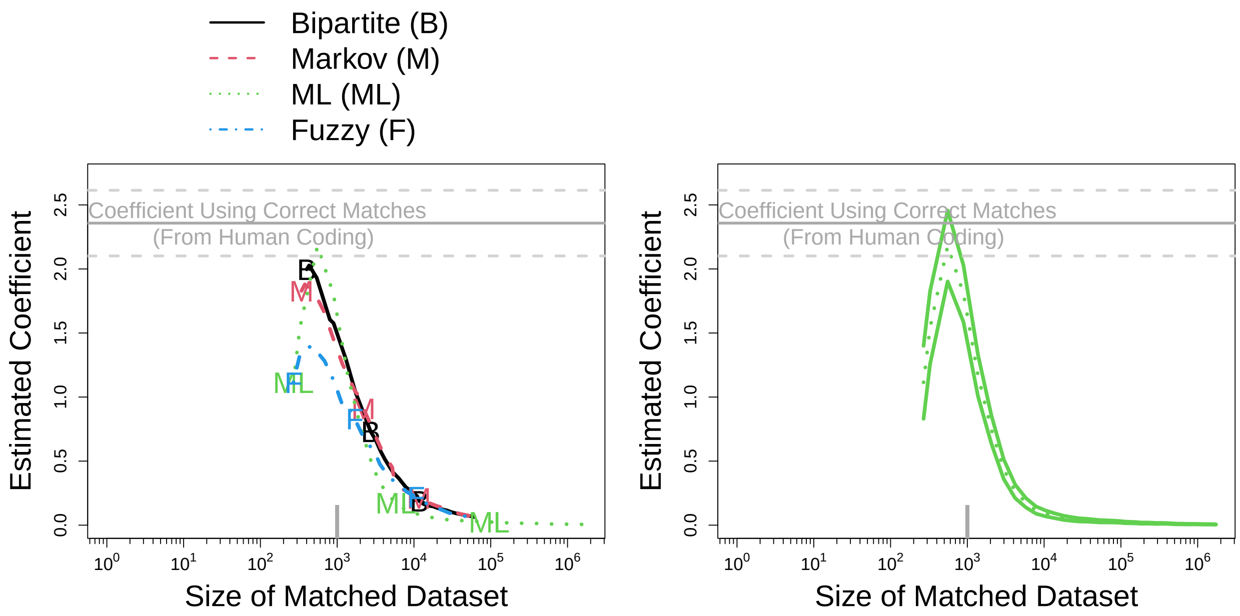

Figure 6: We see the coefficient on log(Assets) for predicting log(1+Contributions) using the ground truth data is about 2.5 (bold gray line, 95% confidence interval displayed using dotted gray lines). We see in the left panel that, at its best point, fuzzy matching underestimates this quantity by about half. The LinkedIn-based matching algorithms better recover the coefficient. The right panel illustrates the robustness figure researchers should consider including in their analyses following organizational matching tasks (the matching results from the machine learning algorithm are used).

It also presents 95% confidence intervals (obtained from heteroskedasticity-consistent standard errors) as a function of the distance as well. We can therefore visually decompose the point and uncertainty estimate as a function of the matched dataset size (where larger sizes—beyond a point—mean poorer data quality). Researchers should expect that effect estimates decay to 0 as the dataset size grows beyond a point due to the attenuation bias effect. In short, with this sort of illustration, researchers can visually examine the point at which attention bias begins to kick in, and could argue about the estimate reliability given their preferred choice of matched dataset size.

To assist researchers in the construction of these graphs, we have included a function in the `LinkOrgs` package which can construct graphs such as those in the right panel of Figure 6. We hope that, by enabling the easy construction of these robustness plots, researchers can further improve the transparency around their matching choises, regardless of whether they use the matching approaches based on the LinkedIn corpus.

Overall, this substantive analysis shows the importance of matching algorithm and distance threshold, and shows promising results for the organizational matching approaches proposed here.

5 Discussion

We have introduced two distinct modeling approaches for record linkage tasks involving organizations leveraging half a billion user contributed records from a prominent social networking site.

The first framework uses the networking data as a massive training set for learning how to distinguish matches name pairs from non-match pairs. We outlined assumptions on which the approach depends. Future research can improve on our work by making the modeling strategy more robust to these assumptions. For example, in the context of the semantic mapping assumption, new machine learning architectures could generate better approximate the function mapping two names into their match probability. Moreover, weighting strategies could weaken the validity strategy by up-weighting in the model training portions of the LinkedIn network most relevant to a particular dataset. While this strategy would come at additional computational cost, it could improve performance.

The second framework uses the LinkedIn data to build a network of organizational names. This network captures information about how the different names interrelate, and, by finding communities in that structure, we obtain can find organizational name pairs without using any linguistic information (which can be misleading for some name merge tasks) but instead using similarities in URL link behavior on the LinkedIn network. While we explore two distinct network modeling and community detection approaches, additional research could improve on this by more explicitly tying linguistic and network information in predicting name matches.

We apply these approaches to three different research tasks that are drawn from substantive papers or motivated by realistic merge tasks. We find peformance gains that are substantial relative to the current practices.

Our results suggest several strategies for applied researchers. Those performing an analysis on data where naming conventions are stable (so that the semantic mapping assumption is satisfied), we would recommend the machine learning approach, which by its design can rapidly match large datasets (about one minute to merge two datasets of size 687 and 6886). When naming conventions are not stable (or cross-language merges are needed), the network approaches are recommended. In uses of the proposed methods, we would generally expect them to succeed best in cases where the name entities were similar to those found in the LinkedIn network.

For methodologically interested researchers, we have released the database of name embeddings from the machine learning approach (available on a stable Harvard Dataverse page, URL to be released upon publication). That is, we have applied the model from Section 3.1.2 and obtained numerical representations for each LinkedIn organization name. These could be useful for other research endeavors (e.g., are organizations with similar embeddings similar in other ways?). Second, we have also made the LinkedIn name pair training data available for others to, hopefully, build upon this work. Due to computational effort required in calculating the LinkedIn match probabilities from Equation 5, we can only make available a subset of the match data. Still, this dataset still comprises over 6.3 million match examples (available on the Harvard Dataverse, URL to be released upon publication).

While the methods introduced here would seem to improve organizational match performance on real data tasks, but there are many avenues for future extensions in addition to those already mentioned.

First, to incorporate auxiliary information and to adjust for uncertainty about merging in post-merge analyses, probabilistic linkage models are an attractive option for record linkage tasks on individuals (Enamorado, Fifield, and Imai 2019). In such models, a latent mixing variable indicates whether a pair of records does or does not represent a match. This latent mixing variable is inferred through an Expectation Maximization (EM) algorithm incorporating information about the agreement level for a set of variables such as birthdate, name, place of residence, and, potentially, employer. Information from our three LinkedIn-based algorithms can be readily incorporated these algorithmes in inferring match probabilities in databases of individuals.

The methods described here can also incorporate covariate information about companies. For instance, researchers can incorporate covariate information in the final layer of our machine learning model and retrain that layer using a small training corpus. This process, an application of transfer learning and an option to be available in `LinkOrgs`, enables extra information to be brought to bear on the merge process while also retaining the rich numerical representations obtained from the original training process performed on our massive LinkedIn dataset.

6 Conclusion

Data sources that are important to scholars of organizational politics often lack common covariate data (such as race or gender), making it difficult to apply probabilistic linkage methods and motivating the widespread use of fuzzy matching algorithms. We have presented new tools for improving the matching of organizational entities using over a half-billion open collaborated employment records from a prominent online social network. This approach can match organizations which contain no common linguistic identifiers or which are written in different language systems. We validate the approach on three example tasks. We show superior performance to the most common alternative method (fuzzy matching), with gains of up to 60% on English-to-English matching tasks and even greater for cross-language matching. We also illustrated how substantive insights can be improved when match quality is increased, with better statistical precision and predictive accuracy. As a novel application of community detection and machine learning to this area, we have erred on the side of simplicity and transparency in extracting an organizational directory from these open collaborated records. Future work may attempt more sophisticated approaches to machine learning architecture, graph construction, community detection, and hybrid matching approaches while using this unique and useful data source. □

7 References

Ansolabehere, Stephen, and Eitan Hersh. 2012. "Validation: What big data reveal about survey misreporting and the real electorate." *Political Analysis* 20 (4): 437–59. `https://doi.org/10.1093/pan/mps023`.

Bolsen, Toby, Paul J. Ferraro, and Juan Jose Miranda. 2014. "Are voters more likely to contribute to other public goods? Evidence from a large-scale randomized policy experiment." *American Journal of Political Science* 58 (1): 17–30. `https://doi.org/10.1111/ajps.12052`.

Chen, Hui, David Parsley, and Ya-Wen Yang. 2015. "Corporate Lobbying and Firm Performance." *Journal of Business Finance & Accounting* 42 (3-4): 444–81.

Clauset, Aaron, Mark EJ Newman, and Cristopher Moore. 2004. "Finding Community Structure in Very Large Networks." *Physical Review E* 70 (6): 066111.

Crosson, Jesse M, Alexander C Furnas, and Geoffrey M Lorenz. 2020. "Polarized Pluralism Organizational Preferences and Biases in the American Pressure System." *American Political Science Review. Accepted.*

Devarriya, Divyaansh, Cairo Gulati, Vidhi Mansharamani, Aditi Sakalle, and Arpit Bhardwaj. 2020. "Unbalanced Breast Cancer Data Classification Using Novel Fitness Functions in Genetic Programming." *Expert Systems with Applications* 140: 112866.

Egger, Roman. 2022. "Text Representations and Word Embeddings." In *Applied Data Science in Tourism*, 335–61. Springer.

Enamorado, Ted, Benjamin Fifield, and Kosuke Imai. 2019. "Using a probabilistic model to assist merging of large-scale administrative records." *American Political Science Review* 113 (2): 353–71. `https://doi.org/10.1017/S0003055418000783`.

Eun, Jihyun, and Seung-Hyun Lee. 2021. "Aspirations and Corporate Lobbying in the Product Market." *Business & Society* 60 (4): 844–75.

Figlio, David, Jonathan Guryan, Krzysztof Karbownik, and Jeffrey Roth. 2014. "The effects of poor neonatal health on children's cognitive development?" *American Economic Review* 104 (12): 4205–30. `https://doi.org/10.1257/aer.104.12.3921`.

"FluentU." 2022. `https://www.fluentu.com/blog/chinese/business-chinese-vocabulary-list-foreign-companies-i`

"Fortune." 2022. `fortune.com/`.

Herzog, Thomas H., Fritz Scheuren, and William E. Winkler. 2010. "Record linkage." *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (5): 535–43. `https://doi.org/10.1002/wics.108`.

Hill, Seth J., and Gregory A. Huber. 2017. "Representativeness and Motivations of the Contemporary Donorate: Results from Merged Survey and Administrative Records." *Political Behavior* 39 (1): 3–29. `https://doi.org/10.1007/s11109-016-9343-y`.

Kim, Jaekyeom, Hyoungseok Kim, and Gunhee Kim. 2020. "Model-Agnostic Boundary-Adversarial Sampling for Test-Time Generalization in Few-Shot Learning." In *European Conference on Computer Vision*, 599–617. Springer.

Kuhn, Max, and Kjell Johnson. 2013. "Remedies for Severe Class Imbalance." In *Applied Predictive Modeling*, 419–43. Springer.

Larsen, Michael D., and Donald B. Rubin. 2001. "Iterative automated record linkage using mixture models." *Journal of the American Statistical Association* 96 (453): 32–41. `https://doi.org/10.1198/016214501750332956`.

Lever, Jake. 2016. "Classification Evaluation: It Is Important to Understand Both What a Classification Metric Expresses and What It Hides." *Nature Methods* 13 (8): 603–5.

Li, Chen, Jiaheng Lu, and Yiming Lu. 2008. "Efficient Merging and Filtering Algorithms for Approximate String Searches." In *2008 IEEE 24th International Conference on Data Engineering*, 257–66. IEEE.

Libgober, Brian. 2020. "Meetings, Comments, and the Distributive Politics of Rulemaking." *Quarterly Journal of Political Science*.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." *arXiv Preprint arXiv:1310.4546*.

Mozer, Reagan, Luke Miratrix, Aaron Russell Kaufman, and L Jason Anastasopoulos. 2020. "Matching with Text Data: An Experimental Evaluation of Methods for Matching Documents and of Measuring Match Quality." *Political Analysis* 28 (4): 445–68.

Navarro, Gonzalo, and Leena Salmela. 2009. "Indexing variable length substrings for exact and approximate matching." In *International Symposium on String Processing and Information Retrieval*, 214–21.

Nownes, Anthony J. 2006. *Total Lobbying: What Lobbyists Want (and How They Try to Get It).* Cambridge University Press.

"Open Secrets." 2022. opensecrets.org/.

Ridge, Jason W, Amy Ingram, and Aaron D Hill. 2017. "Beyond Lobbying Expenditures: How Lobbying Breadth and Political Connectedness Affect Firm Outcomes." *Academy of Management Journal* 60 (3): 1138–63.

Rish, Irina et al. 2001. "An Empirical Study of the Naive Bayes Classifier." In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence*, 3:41–46. 22.

Rohe, Karl, Sourav Chatterjee, and Bin Yu. 2011. "Spectral Clustering and the High-Dimensional Stochastic Blockmodel." *The Annals of Statistics* 39 (4): 1878–1915.

Santurkar, Shibani, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. 2018. "How Does Batch Normalization Help Optimization?" *Advances in Neural Information Processing Systems* 31.

Sundermeyer, Martin, Ralf Schlüter, and Hermann Ney. 2012. "LSTM Neural Networks for Language Modeling." In *Thirteenth Annual Conference of the International Speech Communication Association.*

Van Dongen, Stijn. 2008. "Graph clustering via a discrete uncoupling process." *SIAM Journal on Matrix Analysis and Applications* 30 (1): 121–41.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *Advances in Neural Information Processing Systems* 30.

Wilson, Graham K. 1985. *Business and Politics: A Comparative Introduction.* Macmillan International Higher Education.

Appendix

1.1 Machine Learning Model Details

1.1.1 Step 1. Obtaining Organizational Name Representations

We first outline the modeling strategy for each organizational name.

Input: An organizational name (e.g., "j.p. morgan chase")

1. Break each name into words.
   - For each word of length l_w, query the vector representation for each character to form a $l_w \times 128$-dimensional representation for that word.
   - Apply batch normalization to the feature dimension.
   - Apply a bidirectional LSTM model to each word to obtain a representation for each word.
2. For each alias of l_a words, query the word vectors obtained from the previous step to form a $l_a \times 128$-dimensional representation for the alias.
   - Apply batch normalization to the feature dimension.
   - Apply a bidirectional LSTM model to each alias to obtain a representation for each sequence of words.
3. Apply two feed-forward neural network layers to the initial alias representation to form the final alias representation.

Output: 256-dimensional representation of the organizational name

1.1.2 Step 2. Transforming Name Representations into Match Probabilities

With the name representations in hand, we now discuss in greater detail how we form the final match probabilities.

Input: Numerical representations/embeddings for two organizational names

1. Take the normalized Euclidean distance between the two embeddings, e_i, e_j:

$$d_{ij} = \frac{1}{\sqrt{\dim(e_i)}} ||e_i - e_j||.$$

2. Form the match probability from the distance using unidimensional logistic regression:

$$\Pr(O_i = O_j | A_i = a, A_j = a') = \frac{1}{1 + \exp(-[\beta_0 + \beta_1 d_{ij}])}.$$

Output: A match probability indicating the probability that two names refer to the same organization

The batch normalization layers are included to avoid exploding gradients (e.g., large values are centered and scaled) and to make the learning more robust to initialization (Santurkar et al. 2018). The ∼1.3 million parameters used in steps 1 and 2 are updated jointly using a modified form of stochastic gradient descent. We next give more details about the sampling procedure involved in the model training.

1.1.3 Sampling Design Details for Training of the Machine Learning Model

Recall that we seek to optimize the following:

$$\text{minimize} \sum_{a,a' \in \mathcal{A}} D_{\text{KL}}\left(\widehat{\Pr}(O_i = O_j | A_i = a, A_j = a') \mid\mid \Pr(O_i = O_j | A_i = a, A_j = a')\right).$$

One challenge in performing this optimization in practice is that most a' are very distinct semantically from a, so that the learning process is slowed by the vast number of “easy” predictions (e.g., where $\Pr(O_i = O_j | A_i, A_j)$ is clearly 0). To make matters worse, most match probabilities are 0, a kind of imbalance that can impair final model performance (Kuhn and Johnson 2013).

We adopt two measures to address these two problems. First, we implement a balanced sampling scheme: in the optimization, we ensure half of all training points have $\Pr(O_i = O_j | A_i = a, A_j = a') = 0$ and half have $\Pr(O_i = O_j | A_i = a, A_j = a') > 0$.

Second, we select a, a' pairs that will be maximally informative so that the model is able to more quickly learn from semantically similar non-matches (e.g., “the university of chicago” from “the university of colorado”) or semantically distinct matches (e.g., “massachusetts institute of technology” and “mit”). In our adversarial sampling approach (which is somewhat similar to Kim, Kim, and Kim (2020)), we select, for non-matches in our training batch, alias pairs that have $\Pr(O_i = O_j | A_i = a, A_j = a') = 0$ but have the largest $\widehat{\Pr}(O_i = O_j | A_i = a, A_j = a')$. Likewise, for matches, we select pairs that have lowest predicted probability. More formally, we find the negative a and a' pairs by solving:

$$\text{argmax}_{a,a' \text{ s.t. } \Pr(O_i = O_j | A_i = a, A_j = a') = 0} D_{\text{KL}}\left(\Pr(O_i = O_j | A_i = a, A_j = a') \mid\mid \widehat{\Pr}(O_i = O_j | A_i = a, A_j = a')\right),$$

with, *mutatis mutandis*, the same approach applied for positive pairs.

We achieve this in practice by, at the current state of the model, finding the closest alias to a' in the alias vector space for which $\Pr(O_i = O_j | A_i = a, A_j = a')$ is 0. A similar approach is done to obtain high information positive matches. Intuitively, this process upweighs the importance of similar (very different) aliases that refer to different (the same) organization in the learning process so as to learn how to better learn the semantic mapping of aliases to match probabilities.

We found this adaptive sampling scheme to be important in learning to quickly distinguish similar aliases that refer to different organizations, although only use this adversarial approach to obtain half of each training batch to mitigate against potential problems with this approach (such as occurs if the model is fit on only high surprise mistake alias pairs that occur when users incorrectly link to the same URL).

1.2 Greedy Bipartite Clustering Details

Following Clauset, Newman, and Moore (2004), our modularity score for bipartite community detection is

$$\text{Modularity Score} = \frac{1}{2m} \sum_{a,u} \left[B_{au} - \frac{k_a k_u}{2m} \right] \mathbb{I}\{c_a = c_u\}, \tag{11}$$

where m denotes the total number of edges in the graph, B_{au} denotes the au-th entry in the weighted bipartite network (i.e., the number of ties between alias a and URL u), and k_a denotes the degree of node a (i.e. $k_a = \sum_{a \in \mathcal{A}} B_{au}$). The indicator variable, $\mathbb{I}\{\cdot\}$, is 1 when the community for a equals the community for u (these communities are denoted as c_a and c_u).

The intuition in Equation 11 is that the modularity score is maximized when the number of ties between a and u is large (i.e. B_{au} is big) and a, u are placed in the same cluster (so that $c_a = c_u$). We obtain community estimates based on the greedy optimization of Equation 11 (see Clauset, Newman, and Moore (2004) for details).

1.3 Fuzzy String Matching Details

Fuzzy string distances in our baseline linkage method are calculated as follows. Let $\tilde{a}$ and $\tilde{a}'$ denote the decomposition of organizational aliases a and a' into all their q-character combinations (known as q-grams). For example, if $q = 2$, "bank" would be decomposed into the set {"ba","an","nk"}. The Jaccard measure is then defined to be

$$d(a, a') = 1 - \frac{|\tilde{a} \cap \tilde{a}'|}{|\tilde{a} \cup \tilde{a}'|}.$$

If all q-grams co-occur within a and a', this measure returns 0. If none co-occur, the measure returns 1. If exactly half co-occur, the measure returns 0.5. Following (Navarro and Salmela 2009), we set $q = 2$.