Political Science Research and Methods (2025), **13**, 923–942
doi:10.1017/psrm.2024.55

ORIGINAL ARTICLE

Linking datasets on organizations using half a billion open-collaborated records

Brian Libgober[1] and Connor T. Jerzak[2]

[1]Department of Political Science and Institute of Policy Research, Northwestern University, Evanston, IL, USA and
[2]Department of Government, The University of Texas at Austin, Austin, TX, USA
Corresponding author: Connor Jerzak; Email: connor.jerzak@austin.utexas.edu

(Received 26 July 2023; revised 6 March 2024; accepted 9 June 2024)

Abstract

Scholars studying organizations often work with multiple datasets lacking shared identifiers or covariates. In such situations, researchers usually use approximate string ("fuzzy") matching methods to combine datasets. String matching, although useful, faces fundamental challenges. Even where two strings appear similar to humans, fuzzy matching often struggles because it fails to adapt to the informativeness of the character combinations. In response, a number of machine learning methods have been developed to refine string matching. Yet, the effectiveness of these methods is limited by the size and diversity of training data. This paper introduces data from a prominent employment networking site (LinkedIn) as a massive training corpus to address these limitations. By leveraging information from the LinkedIn corpus regarding organizational name-to-name links, we incorporate trillions of name pair examples into various methods to enhance existing matching benchmarks and performance by explicitly maximizing match probabilities. We also show how relationships between organization names can be modeled using a network representation of the LinkedIn data. In illustrative merging tasks involving lobbying firms, we document improvements when using the LinkedIn corpus in matching calibration and make all data and methods open source.

Keywords: Interest groups; record linkage; text as data; unstructured data

1. Introduction

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 (Larsen and Rubin, 2001; Herzog *et al.*, 2010; Ruggles *et al.*, 2018; Enamorado *et al.*, 2019). Record linkage is an all too common task for researchers building datasets. When a unique identifier, such as a social security number, is shared between data collections and made available to researchers, the problem of record linkage is significantly reduced. Errors in linkage, presumably rare, may be regarded as sources of noise. In cases where unique identifiers like social security numbers are not available, recent literature has developed probabilistic linkage algorithms that can find the same individual in two datasets using stable characteristics such as birth year and race, or even mutable characteristics such as address (Enamorado *et al.*, 2019). The rise of such techniques has paved the way for research that would have been costly or impossible to conduct in previous eras (e.g., Figlio *et al.*, 2014; Bolsen *et al.*, 2014; Hill and Huber, 2017).

These new techniques have had less of an impact so far on scholarship concerning organizational entities, such as corporations, universities, trade associations, think tanks, religious groups,

nonprofits, and international associations—entities that are important players in theories of political economy, American politics, and other (sub-)fields. Like researchers on individuals, scholars studying 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 (and do) rely heavily on exact or fuzzy string matching based on names to link records on organizations—or, alternatively, bear the significant costs of manually linking datasets.

To take an example from the applied political science literature, Crosson *et al.* (2020) compare the ideology scores of organizations with political action committees (PACs) to those without. Scores are calculated from a dataset of position-taking interest groups compiled by a nonprofit (Maplight). The list of organizations with PACs comes from Federal Election Commission (FEC) records. Maplight and the FEC do not refer to organizations using the same names. There is no covariate data to help with linkage. The authors state that matching records in this situation is "challenging" (p. 32) and consider both exact and fuzzy matching as possibilities. Ultimately, they perform exact matching on names after considerable preprocessing because of concerns about false positives, acknowledging that they do not link all records as a result. Indeed, the authors supplement the 545 algorithmic matches with 243 additional hand matches, implying that the first algorithmic approach missed about one in three correct matches.

The challenge faced by Crosson *et al.* (2020) is typical for scholars studying organizations in the US or other contexts. Given the manageable size of their matching problem, the authors are able to directly match the data themselves and bring to bear their subject matter expertise. In many cases, where the number of matches sought is not in the hundreds but in the thousands, practical necessity requires using computational algorithms like fuzzy matching or hiring one or more coders (e.g., undergraduates or participants in online markets such as Amazon's Mechanical Turk).

Both string matching and reliance on human coders have limitations. Even though string distance metrics can link records whose identifiers contain minor differences, they do not optimize a matching quality score and have 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. Likewise, string matching and research assistants would both have difficulty detecting a relationship between Fannie Mae and the Federal National Mortgage Association. Such complex matches can be especially difficult for human coders from outside a study's geographic context, as these coders may lack the required contextual information for performing such matches.

Methodologists have started tackling the challenges that researchers face in matching organizational records. Kaufman and Klevs (2022), 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. While this approach represents an improvement over contemporary research practices, an adaptive system based on fuzzy matching still requires researchers to invest time in producing manual matches and may also struggle to make connections in the relatively common situation where shared characters are few and far between (e.g., Chase Bank and JPM) or where characters are shared but the strings have very different lengths (e.g., Fannie Mae and Federal National Mortgage Association). Scholars are also turning to large language models for performing name linkage tasks (Agrawal *et al.*, 2022); however, these large language models have not been fine-tuned on match tasks, so they may struggle to produce matches similar to how fuzzy matching does.

In this paper, we leverage a data source containing half a billion open-collaborated records from the employment networking site LinkedIn, which can serve as a resource for scholars who seek to link records about organizations. We show how this dataset can assist in three distinct kinds of record linkage methods—the first approach based on machine learning, the second based on network analysis and community detection, and the third based on a combination of

network and machine learning methods. Intuitively, each approach uses the combined wisdom of millions of human beings with first-hand knowledge of these organizations. Our argument is that this combined wisdom from trillions of real-world name-pair examples can, if incorporated into a given linkage strategy, improve matching performance at relatively little cost.

In what follows, Section 2 describes the massive training dataset we constructed from a scrape of LinkedIn. Section 3 describes how the LinkedIn data can be used to improve linkage given two distinct representations of the data stream. Section 4 illustrates the use of these linkage methods on three tasks revolving around the role of money in politics. Section 5 and Section 6 conclude. An open-source package (`LinkOrgs`) implements the methods we discuss. We make the massive LinkedIn name-match corpus available in a Dataverse (doi.org/10.7910/DVN/EHRQQL).

2. Employment networking data as a resource for scholars of organizational politics

In this section, we explain how records created by users on LinkedIn, a leading professional networking platform, hold a wealth of information relevant for researchers studying organizational politics, particularly in the ubiquitous yet challenging task of assembling datasets.

The key insight for the data asset we built is that LinkedIn users provide substantial information about their current and previous employers. For the sake of our illustration, we will use a near census of the publicly visible LinkedIn network circa 2017, which we acquired from the vendor `Datahut.co`. Researchers do have the legal right to scrape this website and use the updated corpus (as the Ninth Circuit Court of Appeals established in *HIQ Labs, Inc., v. LinkedIn Corporation* (2017)). That said, these data do not come cheaply, and, informally, it seems to us that costs have increased as a result of greater investment in anti-scraping technology by site owners in the wake of the decision. Although we do not have a more recent scrape available to us at this time, there are vendors with more recent versions (e.g., using LinkDB (Goh, 2022)). We expect over time that the approaches we take to the 2017 data will be applicable to later scrapes as they become available to the field. The dataset we use contains about 350 million unique public profiles drawn from over 200 countries—a similar size and coverage to LinkedIn's estimates reported during its 2016 acquisition by Microsoft.[1]

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 a 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, which we can denote as U_i. This URL link serves as an identifier for each organization (Table 1).

Table 2 provides descriptive statistics about the scope of the dataset as it relates to organizational name usage. The statistics reveal that, on average, users refer to organizations in about three different ways and that, on average, each of the 15 million aliases links to slightly more than one organizational URL. The table also notes that there are more than 10^{14} alias pairs. The database contains a large number of ways names can refer to the same or different organizational URLs.

3. Individual and ensemble approaches to record linkage using the LinkedIn corpus

We begin our discussion of how to best use the LinkedIn corpus with an example. Suppose we have two datasets, **X** and **Y**. **X** contains data about Wells Fargo Bank, JP Morgan Chase Bank, and Goldman Sachs. **Y** contains data about Wells Fargo Advisors, Washington Mutual (at one time a wholly owned subsidiary of JP Morgan Chase), and Saks Fifth Avenue. Ideally, manual linkage would successfully match Wells Fargo Bank with Wells Fargo Advisors and perhaps even JP

[1]At the time of acquisition, 433 million total members and 105 million unique visitors per month were reported (Microsoft News Center, 2016). We are not able to find authoritative counts of the number of publicly visible profiles.

Table 1. Illustration of source data using three public figures

Name	*Title*	*Organization*	*Organization URL Path* (linkedin.com/company/)	*Start date*	*End date*
Michael Cohen	EVP & Special Counsel to Donald J. Trump	The Trump Organization	the-trump-organization	2007 0501	2017 0418
Allen Weisselberg	EVP/CFO	The Trump Organization	the-trump-organization		2017 0316
Michael Avenatti	Founding Partner	Eagan Avenatti, LLP		2007 0101	2017 0318
Michael Avenatti	Chairman	Tully's Coffee	tully's-coffee	2012 0101	2017 0318
Michael Avenatti	Attorney	Greene Broillet & Wheeler, LLP		2003 0101	2007 0101
Michael Avenatti	Attorney	O'Melveny & Myers LLP	o'melveny-&-myers-llp	2000 0101	2003 0101

Table 2. Descriptive statistics for the LinkedIn data

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	$>10^{14}$

Morgan Chase Bank with Washington Mutual, while rejecting all other matches—including between Goldman Sachs and Saks Fifth Avenue, despite some passing phonetic similarity between the names.

Figure 1 presents a checkered flag diagram illustrating our task. Names in the **X** dataset are on the left side. Names in the **Y** dataset are on the right. Each pair of names is represented by a node. To perform linkage, investigators apply an algorithm that assigns scores to all the nodes. They then consider a node to represent a match if it clears some numeric threshold (or, alternatively, investigators can re-weight links in accordance with their match probability). There are many possible functions to score pairs. For example, exact matching scores each node as 1 if A_i and A_j are equal, 0 if unequal, accepting all pairs where the score is 1. In this example, exact matching would fail to link any of the organizations. The figure presents scores using the fuzzy matching approach, in this case, with the Jaccard metric.

These scores present a familiar trade-off. If a cutoff of 0.7 is adopted, nothing matches. If 0.5 is selected, then Wells Fargo Bank successfully matches to Wells Fargo Advisors, but JP Morgan Chase Bank does not match to Washington Mutual. If a cutoff of 0.35 is selected, all the correct matches are included but also several wrong ones. If a cutoff of 0.2 is selected, everything matches everything. There are no perfect options.

The familiar trade-off facing researchers in this example comes about in part because the scores are too similar between pairs that we do and do not wish to match. Our focus is on algorithmic interventions that can produce scores that make it easier to distinguish between pairs that should match and those that should not at any particular cutoff.

While ultimately we will propose an ensemble of two distinct approaches, we begin here by discussing the intuition underlying each. The first idea is that, to the extent that an algorithm trained on the LinkedIn corpus can reward similarities in latent meanings (e.g., "bank" and "mutual") and punish dissimilarities (e.g., "bank" and "avenue"), it stands a good chance of

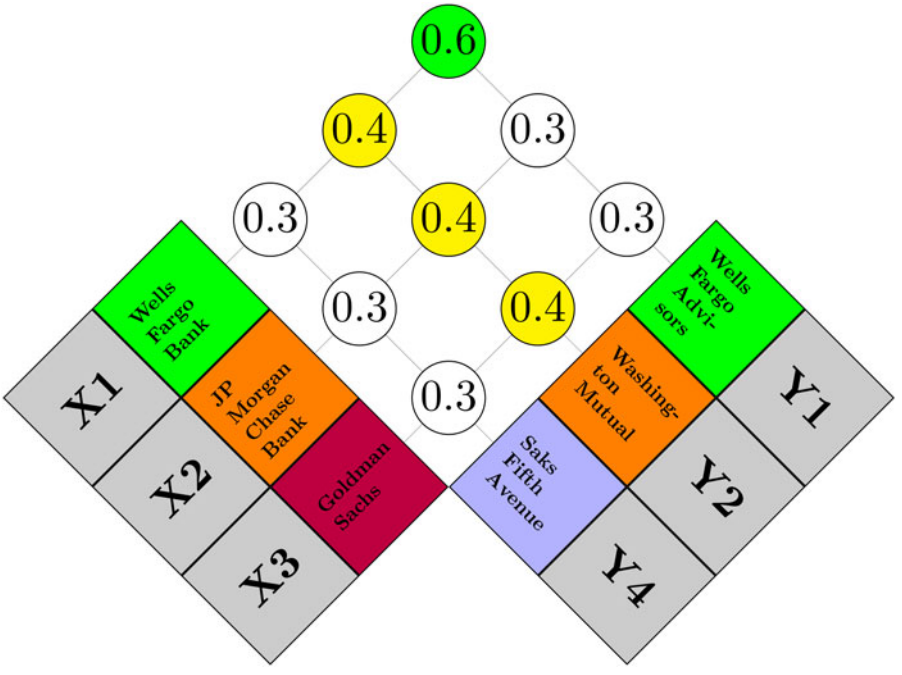

Figure 1. Checkered flag diagram describing the organizational linkage problem.

improving upon existing character similarity methods. Machine learning approaches to this task are appealing, and we focus on applying these methods using the LinkedIn network. Despite the increasing sophistication of learning algorithms, these methods do have limitations, as the words in a name do not always have semantic value. Indeed, without specialized domain knowledge, it may be easy to miss matches between organizations having multiple acronyms. To account for this limitation, an approach that utilizes community detection algorithms is desirable and complements the first approach by leveraging alias-to-URL links more explicitly. We provide more details on each approach below and conclude by describing how to unify them in an ensemble.

3.1 Machine learning approach

Machine learning continues to make so many advances that it is becoming hard to choose, let alone justify as best, any particular framework. The future will no doubt yield improvements in any machine learning approach for modeling name match probabilities, as the rapid progress in large language models has made apparent (Wei *et al.*, 2022; Jiang *et al.*, 2023). That said, to make progress we need to make and explain our choices.

We set up the machine learning problem as the task of learning a function, f_n, that maps a textual alias, a, to a point in a high-dimensional, real-valued vector space. The distance between two aliases, a and a', is to be calculated as $|f_n(a) - f_n(a')|$. We are mindful that one major benefit of learning a map from the space of strings to numerical vectors is faster matching. String similarity algorithms, such as the Jaccard algorithm, typically require an operation on each combination of entries that one wishes to match in sets $\mathbf{X}$ and $\mathbf{Y}$; the calculation of a single score can be quite time-consuming. By contrast, applying f_n to all the entries of $\mathbf{X}$ and $\mathbf{Y}$ generates two sets of points in a vector space. Calculating a distance between all pairs of points is typically a much faster computation than the equivalent string distance calculation on all the pairs.

The function, f_n, that our algorithm will learn is, to a degree, a black box. It optimizes over many parameters by seeking to best fit some target outcome; hence, the way we structure the target influences the ultimate algorithm we produce. Perhaps the simplest approach to setting up an outcome would be to look at the set of all alias pairs, $\mathcal{P} = \{\mathcal{A}_i\} \times \{\mathcal{A}_j\}$, and assert that two aliases are linked if two people used those aliases to refer to the same URL and not linked if no one ever did. A limitation of this "lookup table"-like approach is that it has no sensitivity to the number of links in the data, so a person who mistakenly writes "Goldman Sachs" as their employer and links to the Saks Fifth page would get equal weight to the much more common case where employees write that they work at "Goldman Sachs" and link to the Goldman page.

Incorporating information about the relative number of links is clearly desirable, but requires care. As a starting point, we borrow ideas from naive Bayes classifiers to calculate a probability that two aliases are indeed true matches using the depth of ties between links. In Section A.I.1.1, we explain assumptions that would allow the interpretation of this outcome variable as a probability, written as:

$$\begin{aligned} \Upsilon_{ij} &= \Pr\big(i \text{ and } j \text{ match} | A_i = a, A_j = a'\big) \\ &= \sum_{u \in \mathcal{U}} \Pr\left(U_i = u \mid A_i = a\right) \Pr\left(U_j = u \mid A_j = a'\right) \end{aligned} \tag{1}$$

To explicate this formula, note that for each URL u, the term $\Pr\left(U_i = u \mid A_i = a\right) \Pr\left(U_j = u \mid A_j = a'\right)$ reflects the proportion of occurrences of u given alias a times the proportion of occurrences of u given the alias a'. As an illustration, consider a profile URL like LinkediIn.com/company/wellsfargo and the two aliases, "JP Morgan Chase Bank" and "Wells Fargo Advisors." Whenever "JP Morgan Chase Bank" is used, it generally occurs with a different company profile, so this particular URL contributes little to the overall probability even though "Wells Fargo Advisors" almost always links to this particular profile URL. By contrast, if "Wells Fargo Bank" and "Wells Fargo Advisors" both typically link to this same profile page, then the probability of a match will be calculated as high. The overall loss function we seek to minimize is

$$Loss = \sum_i \sum_j \mathrm{KL}(\widehat{\Upsilon}_{ij}, \ \Upsilon_{ij}), \tag{2}$$

where the KL divergence computes the distance in probability space between $\widehat{\Upsilon}_{ij}$, the predicted match probability, and Υ_{ij}, the match probability as computed using the LinkedIn corpus.

We now discuss how we structure the f_n function that ultimately generates $\widehat{\Upsilon}_{ij}$ (for details, see Section). Our approach builds from work on the vector representations of words (Mikolov *et al.*, 2013). In our case, we build a model for organizational name matches from the characters on up.[2]

Figure 2 provides an illustration of the model's structure. In particular, we model each *character* as a vector (with each dimension representing some latent quality of that character), each *word* as an ordered sequence of character vectors, and each *organizational name* as an ordered sequence of word vectors learned from their character constituents. That is, first, we learn a good representation of words based on ordered characters. Then, we learn a good representation of organization names based on ordered words. Finally, we repeatedly optimize the system through backpropagation to minimize the loss function above.

Here, an important parameter is the dimension of the vector representation. We adopt a 1,024-dimensional numeric representation of organizational names, balancing computational efficiency with informational richness. Similar to other word embedding approaches, each

[2]This sequential approach pays more attention to the order of characters/words than the traditional bag-of-words/characters approaches that historically saw wide use in political science text analysis; for discussion of word-vector approaches, see Rodriguez and Spirling (2022).

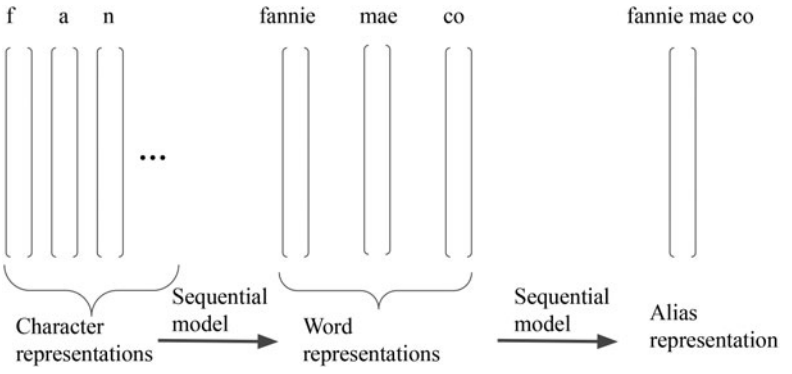

Figure 2. A high-level illustration of the multi-level neural network's architecture. We learn from data how to represent, in a vector space, (a) the characters that constitute words and (b) the words that constitute organizational names. Each lower level is used to generate a higher-level representation in vector space via a flexible model, with better lower-level representations learned by tuning higher-order representations.

dimension of the ultimate vector has a latent semantic value, although that value may be hard to interpret.

In Figure 3, we examine how the algorithm has mapped aliases into the embeddings space. Due to the difficulties of visualizing multi-dimensional data, we project the alias embedding space down to two dimensions via Principal Component Analysis (PCA). Pairs of aliases representing the same organization are represented by the same graphical mark type. We see that aliases representing the same organization are generally quite close in this embedding space. The model seems to be able to handle less salient information 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/absence of discrete letter combinations. While researchers may drop common words like "corporation" based on intuition, our optimized model learns which words to emphasize or ignore from data.

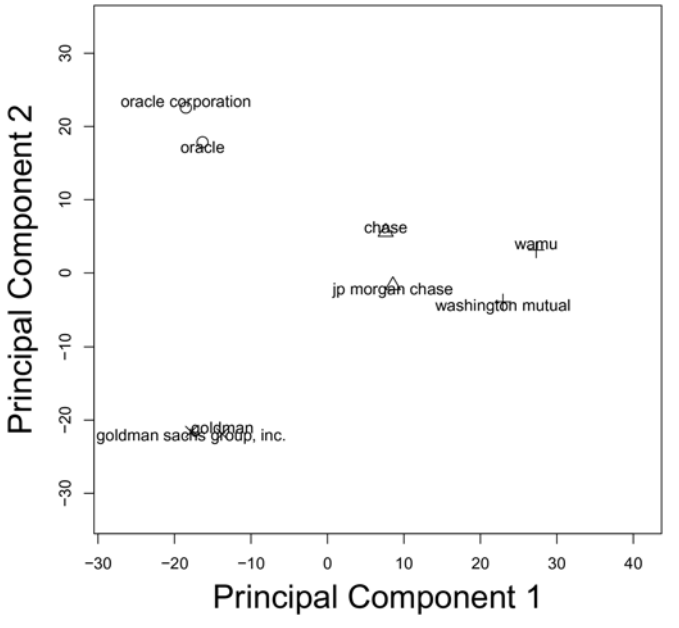

Figure 3. Visualizing the machine learning output: Similar organizational names are close in this vector space, which has been projected to two dimensions using PCA.

While these examples are interesting and encouraging, they do not present a particularly rigorous test of the algorithm's performance. In Figure 4, we examine how well names that should match do match and how well names that should not match do not match according to the estimated model. In particular, we hold out from training 2000 randomly chosen pairs of aliases sharing a URL ("matches") and 2000 randomly chosen pairs of aliases where a URL is not shared ("non-matches"). For the set of matches and non-matches, we provide density plots of the match quality under fuzzy matching and under our learning model.

The right panel of Figure 4 considers predicted match probabilities for the out-of-sample set of match and non-match examples from the LinkedIn corpus. In particular, it shows density plots of the predicted probabilities of pairs that are matches and, separately, non-matches. If the algorithm is working as it should then the overall distribution of match probabilities for matches and non-matches should differ greatly. Indeed, this is what the figure finds. A KS test for assessing whether the probabilities are drawn from the same distribution yields a test statistic of 0.87 ($p < 10^{-16}$). A statistic of 0 indicates complete distribution overlap, while 1 signifies perfect separation. Encouragingly, we are closer to this second case. The left panel shows results with Jaccard-distance fuzzy matching, which yields KS test statistics ranging from 0.47 to 0.55, depending on the character q-grams used.

Despite these successes, there are true links that would remain hard to model using this prediction-oriented framework. For instance, the aliases "Chase Bank" and "JP Morgan" have a low match probability. To handle such cases, we next show how the LinkedIn data introduced in this paper can be used in a network-oriented approach to improve organizational record linkage.

3.2 Network-based linkage algorithms with LinkedIn data

As we have already seen, organizational names sometimes contain little semantic information; as a result, methods that focus on uncovering these meanings have a ceiling. Often, the relationship between two aliases for an organization is something that one simply has to know. The question then is how to best leverage the knowledge implicit in the LinkedIn network, bearing in mind that the raw data may not reveal the full depth of knowledge in the network.

Instead of viewing the record linkage task as matching two lists of organization names directly, one can instead view it as connecting these names on a graph. A tricky point, of course, is that the

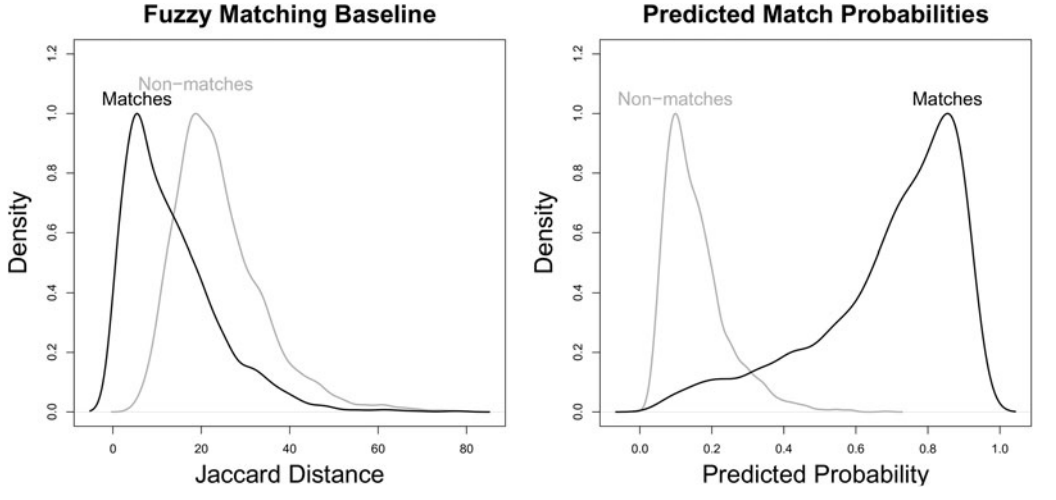

Figure 4. Visualizing the machine learning model output: Left. Fuzzy matching as a baseline generates distances between strings that have trouble distinguishing matches from non-matches in some cases Right. On average, organizational alias matches have higher match probabilities compared with the set of non-matches.

names of the organizations in one's lists may not actually be on the graph. We address this issue below when thinking about an ensemble strategy (and also through our third application). But even assuming the names are on the graph, one must consider the sense of connectedness that is most useful.

The simplest concept of connectedness would assert that two aliases are linked if someone has attributed the same URL to both names. This notion would often fail to follow transitivity. In other words, if A and B refer to the same organization and B and C refer to the same organization, then A and C should refer to the same organization—but, despite this, they may not share a URL. So, without care, we may miss this connection. It is tempting then to insist on transitivity in alias names. Implicitly, doing so casts the problem of record linkage as placing an organization in a particular connected component of the LinkedIn network. An issue here is that, if there are spurious links, then many components will be merged where there is little evidence to support such an action. An approach that is in between, allowing for *some* transitivity when the evidence is sufficiently strong but not when the evidence is weak, is desirable. Community detection methods aim to find this sweet spot.

Because community detection is a well-studied problem that occurs in many applied contexts (Rohe *et al.*, 2011), we consider two algorithms established in the methodological literature: Markov clustering (Van Dongen, 2008) and greedy clustering (Clauset *et al.*, 2004). We focus on these algorithms because they model the network in distinct ways and are computationally efficient. Implementation details are in Appendix II; here, we offer a brief sketch of each.

Figure 5 shows how we can represent the data source explicitly as a network for Markov clustering. Here, 11 organizational aliases are presented as nodes. Aliases are connected by edges. In principle, these could be directed or undirected, weighted or unweighted depending on one's modeling strategy. The figure shows edges with weights that follow a naive Bayesian strategy for calibrating the amount of information between names (we use a similar probability calculation as in Equation 1). Under this approach, the probability that A and B is connected is the same as the probability that B and A are connected. Therefore, this yields a weighted, undirected graph. Notable, we see the strong ties where the connection is surprising based on semantic information, such as "chase" and "washington mutual." Visually, it is clear that there are two to three clusters where links are denser, but there are also occasional ties across the clusters that ex ante are hard to identify as spurious or real. These clusters of nodes with relatively dense connections are the "community" of aliases we wish to discover.

Markov clustering applies arithmetic operations to the edges of the graph that alternately diminish weak links in the graph and enhance strong ones. The middle panel of Figure 5 shows the partial completion of this algorithm while the right panel shows it at convergence, where each organization is placed in a single "community" identified by the alias most prominent in it (for example, "jp morgan chase" and "bank of america").

In contrast to Markov clustering, greedy clustering is an iterative algorithm that begins by assuming each node is its own community and then merges communities that would result in the largest increase in the overall "quality" of the network structure. We use one of the most ubiquitous quality measures called a modularity score. This score is 0 when community ties between aliases and URLs occur as if communities were assigned randomly. It gets larger when the proposed community structure places aliases that tend to link to the same URLs in the same community (Clauset *et al.*, 2004). While the Markov clustering algorithm requires edges to have probability weights, greedy clustering does not, which enables community detection with a bipartite (as opposed to adjacency) representation.

Ultimately, we find somewhat better performance with this bipartite representation of the LinkedIn network where the names and URLs are both considered nodes and the links only occur between names and URLs if there is an attribution in the LinkedIn database (with edge weights given by the number of times two attributions are made).

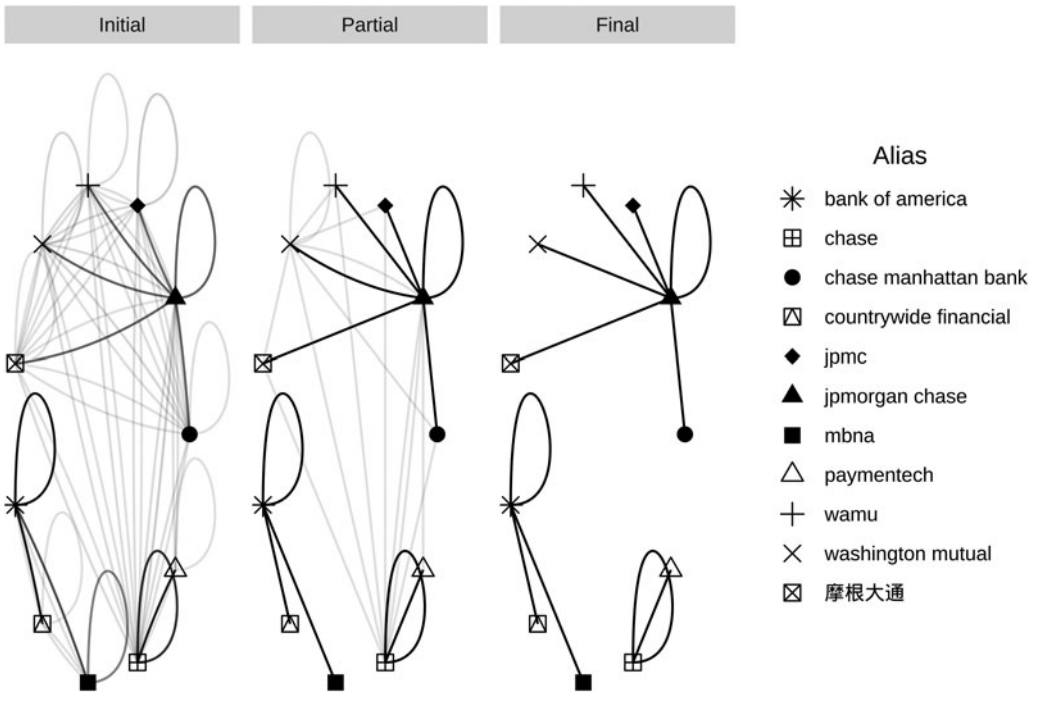

Figure 5. LinkedIn Name Network and Community Detection Algorithm. The figure illustrates how the community detection algorithm links organizational aliases for 11 aliases in the banking space. LEFT: A weighted graph derived from raw counts of connections in the LinkedIn database. Due to some stronger connections between some nodes, the figure suggests some clusters visually, but there are also connections across clusters, which makes clustering a non-trivial problem. MIDDLE: The Markov clustering algorithm proceeds toward convergence. The connections between nodes are now stronger within and weaker across communities. A "Bank of America" community appears to have been detected. RIGHT: Community detection has converged to three clusters. The tight connection between JP Morgan and its subsidiaries contrasts with what is found through word embeddings, where these names are rather distant in the machine learning-based vector space (compare the positioning of some of the same aliases in Figure 3).

3.3 Joint network and prediction-based record linkage using the LinkedIn corpus

The LinkedIn-calibrated machine learning model uses complex semantic information to assist matching but does not make use of graph-theoretic information. The network-based methods use network information but do not use semantic information to help link names in that network. To get the best of both worlds, we propose a third, unified approach that uses both the semantic content and graph structure of the LinkedIn corpus.

The unified approach is an ensemble of both network and machine learning methods and involves three steps. Figure 6 returns to the example at the beginning of this section involving the merge of a dataset about Wells Fargo Bank, JP Morgan Chase Bank, and Goldman Sachs (**X**) with another dataset about Wells Fargo Advisors, Washington Mutual, and Saks Fifth Avenue (**Y**). The figure presents several checkered flags illustrating the multi-step approach. In step (a), machine learning-assisted name linkage is directly applied between the two datasets. Similar to fuzzy string matching, scores are calculated on the cross product of two sets of names; scores exceeding a threshold (set to 0.5 in the figure) are said to match. In this particular example, fuzzy matching could produce similar results, but the thresholds and scores would differ, and, as a result, so too would performance. In step (b), machine learning-assisted name linkage is applied to an intermediary directory built using community detection. We attempt to place **X** in their proper community and **Y** in their proper community, and then we consider entries in **X** and **Y** as linked if they are placed in the same community.

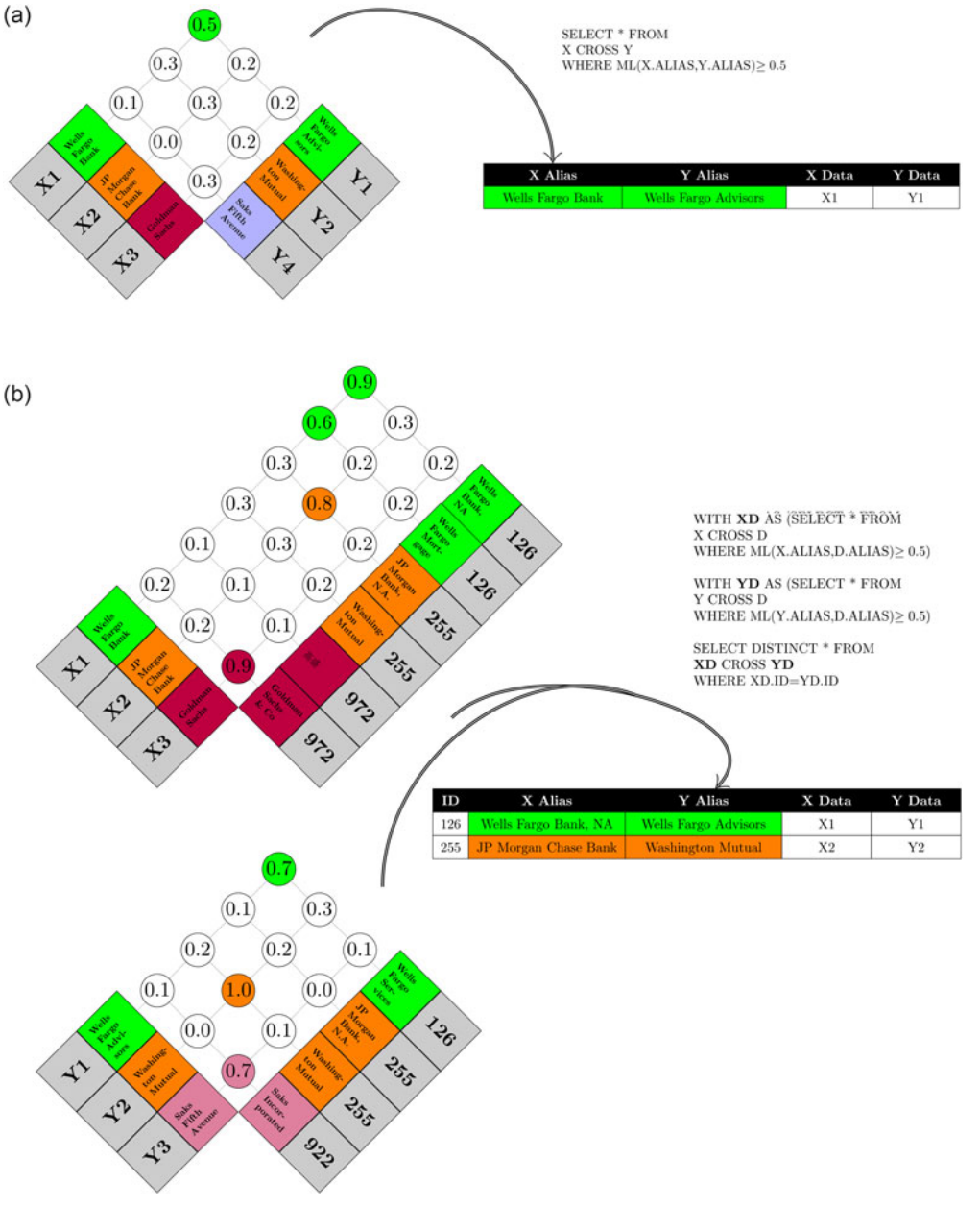

Figure 6. Checkered flag diagrams illustrating a unified approach to name record linkage using the LinkedIn corpus, (a) Direct linkage through machine learning-optimized string matching, (b) Indirect linkage through a directory constructed using community detection.

This example shows how the unified approach can put to use the potential of both methods presented thus far. Through step (a), it can link datasets for organizations that do not appear on LinkedIn but whose naming conventions are similar. Through step (b), the unified method picks up on relationships that are not apparent based on names and require specialized domain expertise to know. We now turn to the task of assessing how these different algorithmic approaches and representations of the LinkedIn corpus perform in examples from contemporary social science.

4. Evaluation tasks

4.1 Method evaluation

Before we describe the illustrative tasks, we first introduce our comparative baseline and evaluation metrics. This introduction will help put the performance of the methods into context.

4.1.1 Fuzzy string matching baseline

We examine the performance of the LinkedIn-assisted methods against a fuzzy string-matching baseline. While there are many ways to calculate string similarity, we continue to focus on fuzzy string matching using the Jaccard distance measure to keep the number of comparisons manageable. Other string discrepancy measures, such as cosine distance or edit distance, produce similar results.

4.1.2 A machine learning baseline

We also examine the performance of the LinkedIn-assisted methods against a machine learning baseline, "DeezyMatch," that uses a recurrent neural network-based fuzzy matching approach outlined in Hosseini *et al.* (2020), with hyperparameters left at their defaults. This approach will provide a helpful baseline for contextualizing performance.

4.1.3 A network approach baseline

We also examine performance against a simple method (hereafter, "lookup") that uses the LinkedIn data as a giant lookup table for organizations to assess the relative value-added of the clustering algorithms. In this approach, we consider two aliases as matched if they link to the same URL at least once in the LinkedIn corpus.

4.1.4 Performance metrics

We consider two measures of performance. First, we consider the fraction of true matches discovered as we vary the acceptance threshold. This value is defined to be

$$\text{True positive rate} = \frac{\#\text{ of true positives found}}{\#\text{ of true positives in total}} \tag{3}$$

This measure is relevant because, in some cases, researchers may be able to manually evaluate the set of proposed matches, rejecting false positive matches. The true positive rate is therefore more relevant for the setting in which 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 (i.e., matches the algorithm finds that are not, in fact, real). Failure to consider the costs of false positives can lead to undesirable conclusions about the performance of algorithms. "Everything matches everything" is a situation that ensures all true matches are found, but the results are not informative. Given such concerns, we also examine a measure that considers the presence of true positives, false positives, and false negatives known as the F_β score, defined as

$$F_\beta = \frac{(1+\beta^2)\cdot \text{true positive}}{(1+\beta^2)\cdot \text{true positive} + \beta^2 \cdot \text{false negative} + \text{false positive}}, \tag{4}$$

where the β parameter controls the relative cost of false negatives compared to false positives (Lever, 2016). In the matching context, errors of *inclusion* are typically less costly than errors of *exclusion*: the list of successful matches is easier to double-check than the list of non-matched

pairs. For this reason, we examine the F_2 score, a choice used in other evaluation tasks (e.g., Devarriya *et al.* (2020)), which weighs false negatives more strongly than false positives.

4.1.5 Comparing algorithm performance across acceptance thresholds

Approximate matching algorithms have a parameter that controls how close a match must be to be acceptable.[3] Two algorithms might perform differently depending on how the acceptance threshold parameter is set. This threshold is not directly comparable across algorithms. For instance, a change of 0.1 in the match probability tolerance under the ML algorithm implies a much different change in matched dataset size than a 0.1 change in the Jaccard string distance tolerance. To compare the performance of these algorithms, our figures and discussion focus on the size of matched datasets induced by an acceptance threshold. The most stringent choice produces the smallest dataset (i.e., consisting of the exact matches), while the lowest possible acceptance threshold produces the cross-product of the two datasets (i.e., everything matches everything). Between the two, different thresholds produce datasets of different sizes. By comparing performance across matched dataset sizes, we can evaluate how the algorithms perform for different acceptance thresholds.

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 for which those firms lobbied. These returns are significantly higher than the positive returns experienced by market competitors and firms that send regulators written correspondence. Matching meeting logs to stock market tickers is burdensome because there are almost 700 distinct organization names described in the lobbying records and around 7000 public companies listed on major US exchanges. Manual matching typically involves research on these 700 entities using tools such as Google Finance. While the burden of researching 700 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 all lobbying efforts in Congress, the burden could become insurmountable.

Treating the human-coded matches in Libgober (2020) as ground truth, results show how the incorporation of the LinkedIn corpus into the matching process can improve performance. Figure 7 shows that the LinkedIn-assisted approaches almost always yield higher F_2 scores and true positives across the range of acceptance thresholds. The highest F_2 score is over 0.6, which is achieved through the unified approaches, the machine learning approach, and the bipartite graph-assisted matching. The best-performing algorithm across the range of acceptance thresholds is the unified approach using the bipartite network representation when combined with the distance measure obtained via machine learning. The percentage gain in performance of the LinkedIn-based approaches is higher when the acceptance threshold is closer to 0; as we increase the threshold so that the matched dataset is ten or more times larger than the true matched dataset, the F_2 score for all algorithms approaches 0, and the true positive rate approaches 1.

For illustration, let us examine a case where it successfully identifies a correct match that fuzzy matching fails to detect. 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

[3]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 the community structure of LinkedIn.

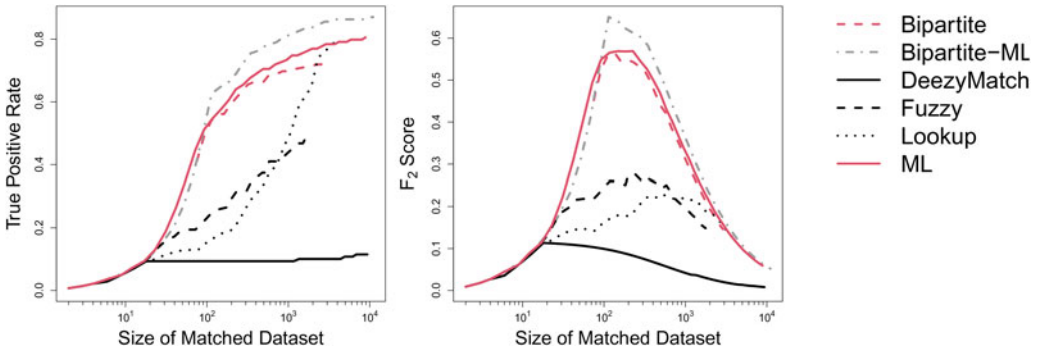

Figure 7. We find that dataset linkage using any one of the approaches with the LinkedIn network obtains favorable performance relative to fuzzy string matching both when examining only the raw percentage of correct matches obtained (Left Panel) and when adjusting for the rate of false positives and false negatives in the F_2 score (Right Panel). In both figures, higher values along the *Y*-axis are better. The "Bipartite" refers to the Bipartite network-based approaches to linkage. "ML" refers to the machine learning approach introduced above. "Fuzzy," "DeezyMatch," and "Lookup" refer to the string distance, machine learning, and network baselines. "Bipartite-ML" refer to the ensemble of "Bipartite" and "ML." See Figure A.VII.1 for full results with Markov approaches included.

match in the LinkedIn-based directory, so the two organizations are successfully paired using the patterns learned from the LinkedIn corpus.

Another relevant consideration in applied matching tasks is compute time. In Section A.VI.1.1, we document the runtime of each approach in the different applications. As expected, the network-based approaches have the greatest computational cost, as some measure of distance between each candidate observation must be computed against all of the hundreds of thousands of entities in the LinkedIn corpus. For these network approaches, the runtime is on the order of several hours for this roughly 700 by 7000 name merge. By contrast, fuzzy matching runs in less than 1 minute; the machine learning approach without the combined network approach runs in roughly 5 min on 2024 hardware. Scaling the best methods is, therefore, a potential concern as one reaches datasets with organizations numbering in the tens or hundreds of thousands. Back-of-the-envelope calculations suggest that a 10,000 by 10000 organization match would potentially have a 2-3 day runtime using full Bipartite-ML, which is long but not unacceptable, as is it performed once in the course of an entire project without much researcher intervention.

Additional strategies would likely be necessary to scale to a 100000 by 100000 name-matching problem, as the best performing, but slowest, algorithm would run somewhere around 255 days, a wait time not realistic for the research iteration process. Two such strategies are parallelization and locality-sensitive hashing. While parallelization can speed up easily subdivided problems like name matching, most computational costs are spent checking pairs that have low match probabilities. Techniques such as locality-sensitive hashing can improve speed by avoiding comparisons between unlikely matches (Green, 2023) (Table 3).

Overall, the results from this task illustrate how the LinkedIn-assisted methods appear to yield better performance than commonly used alternative methods, such as fuzzy matching, in the ubiquitous use case when researchers do not have access to shared covariates across organizational datasets.

4.3 Task 2: linking financial returns and lobbying expenditures from fortune 1000 companies

In the next evaluation exercise, we focus on a substantive question drawn from the study of organizational lobbying: do bigger companies lobby more? Prior research (Chen *et al.*, 2015) leads us to 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 *et al.*, 2017; Eun and Lee, 2021). Our reason for focusing on an application

Table 3. Runtime on the meetings data analysis

Algorithm	Runtime (mins)
Bipartite	13.12
Bipartite-ML	251.38
DeezyMatch	0.24
Fuzzy	0.27
Lookup	1.35
Markov	8.61
Markov-ML	113.00
ML	1.63

where there are *such* strong theoretical expectations is to illustrate how results from different organizational matching algorithms can influence one's findings—something that would not be possible without well-established theory about what researchers should find.

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, where the measure of firm size we focus on is the average total assets in the 2013-2018 period. The key linkage variable will be organizational names that are present in the two datasets, that is to say, the name of the company according to Fortune and according to OpenSecrets. We manually obtained hand-coded matches to provide ground truth data.

In Figure 8, we explore the substantive implications of different matching choices—how researchers' conclusions may be affected by the quality of organizational matches. We see that the coefficient relating log organizational assets to log lobbying expenditures using the human-matched data 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

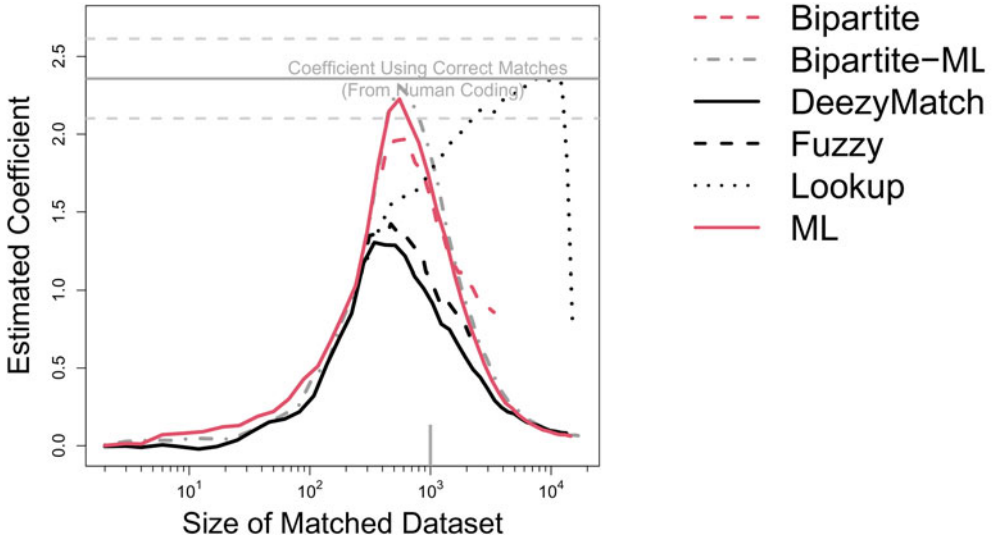

Figure 8. The coefficient of log(Assets) for predicting log(1+Expenditures) using the ground truth data is about 2.5 (depicted by a bold gray line; 95 percent confidence interval is displayed using dotted gray lines). At its best point, fuzzy matching underestimates this quantity by about half. The LinkedIn-based matching algorithms recover the coefficient better. See Figure A.VII.3 for full results with Markov approaches included.

LinkedIn-assisted approaches, with the effect estimates being closer to the true value. For all algorithms examined in Figure 8, there is significant attenuation bias towards 0 in the estimated coefficient as we increase the size of the matched dataset, as poor-quality matches inject noise into estimation. Overall, we see from the right panel that match quality depends on algorithm choice as well as string distance threshold, with the LinkedIn-based approaches capable of estimating a coefficient within the 95 percent confidence bounds of the ground truth estimate. Fuzzy matching and DeezyMatch, at their best, find an estimate that is only half as large in magnitude as the true value.

4.4 Task 3: out-of-sample considerations: merging YCombinator & PPP data

A final question is about how these methods perform in "out-of-sample" testing, which is performed with organizations that we do not expect to be well-represented in the current version of the LinkedIn data for whatever reason. While our methods could be adapted to use more recent versions of the LinkedIn data, LinkedIn data from 2017 cannot directly describe organizations that did not exist then. In this task, we analyze data from the period after the main data collection took place in an effort to understand the strengths and limitations of the various linkage strategies in this context.

Here, we first examine data from a YCombinator directory on incubator startups. The dataset contains a collection of startups involved in the YCombinator seed-funding program, detailing their name, website, business model, team size, and development stage. This data provides a snapshot of the companies' early progression, from inception to public trading or acquisition. The YCombinator program was launched in 2005; for a purer out-of-sample test, we subset the data to the 2017–2024 period.

We merge these startups to the Paycheck Protection Program (PPP) loan database. The PPP (2020–2021) was a program aimed at providing financial relief to businesses during the COVID-19 pandemic. The dataset includes entries with key financial metrics, such as loan amount, approval date, borrower name and address, and employment impact. The task of matching startups to the PPP loan data could be relevant for evaluating the role of these loans on long-term firm survival as well as for thinking about the regulatory advantages that come from affiliation with a business network like YCombinator. Importantly, none of these covariates overlap with the Y-Combinator data. We subset both sets of data to target businesses within the San Francisco area.

As expected, we find in Figure 9 that the linkage approaches only using the directory of firms with established LinkedIn pages as of 2017 provide no gain in the overall F_2 score relative to fuzzy matching. If these methods were rebuilt with a subsequent scrape of the LinkedIn database, they would likely do better with these new organizations. That said, the machine learning approach still provides a boost over fuzzy matching: the approach has inferred more enduring information about the link probability between companies based on the semantic content of names.

5. Discussion: limits and future of the LinkedIn data in improving record linkage

We have shown how to use half a billion user-contributed records from a prominent employment networking site to help link datasets about organizations. Researchers studying organizations frequently find themselves in situations where they must link datasets based on shared names and without common covariates (Crosson *et al.*, 2020; Thieme, 2020; Rasmussen *et al.*, 2021; Carpenter *et al.*, 2021; Stuckatz, 2022; Abi-Hassan *et al.*, 2023; González and You, 2024). Existing methods, notably human coding and fuzzy matching, or some combination of the two, are costly to apply and often involve ad hoc decision-making by scholars about what seems to be working well (or well enough). We have shown how the LinkedIn corpus can be

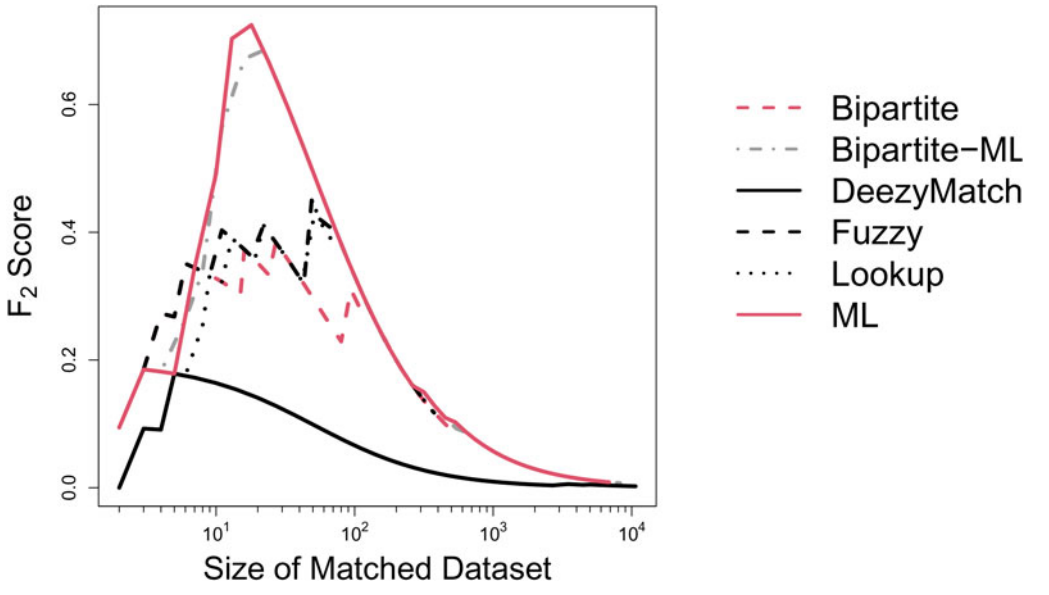

Figure 9. In this YCombinator example, we see that the network-based approaches offer no relative benefit in terms of true positives when adjusting for false positives, yet the machine learning approach that uses the LinkedIn corpus performs well over fuzzy matching. Higher values along the *Y*-axis are better. See Figure A.VII.2 for full results with Markov approaches included.

used, either via machine learning or network detection, to improve organizational record linkage. These approaches are summarized in Table 4.

These results may have implications for applied work. In our second application, the choice of record linkage method is consequential for the ultimate regressions that one runs and intends to present to other scholars. Using a unified approach, we were able to estimate a coefficient of theoretical interest within a 95 percent confidence interval using ground truth. Using other methods, particularly fuzzy matching, we were unable to recover the coefficient of interest. Although the sign was correct, the magnitude was statistically and substantively different.

Table 4. Comparing different approaches to organizational record linkage

	Fuzzy String Matching	*LinkedIn-Calibrated ML*	*LinkedIn Network Approaches*	*Combined ML+Network Approach*
Character				
Optimized for organizational name matching?	No	Yes	No	Partially
Text representation	Discrete	Continuous	Discrete	Continuous
Information used	Semantic	Semantic	Graph theoretic	Semantic + graph theoretic
Hyper-parameters	Acceptance threshold; *q*-gram settings	Acceptance threshold; ML model architecture	Acceptance threshold; *q*-gram settings; clustering hyperparameters	Acceptance threshold; ML model architecture; clustering hyperparameters
Data Requirements				
Requires access to saved matching model parameters?	No	Yes	No	Yes
Requires access to saved alias clustering?	No	No	Yes	Yes

Typically, scholars do not have access to ground truth and, therefore, will not have a sense of how well or how badly they are doing in the aggregate. This is a potentially serious problem affecting research on organizations; however, we do not believe that this application alone should cast substantial doubt on what scholars have been doing. Typically, researchers use a mix of hand-coding and automated methods, and we expect that this kind of approach will do better than a purely automated approach (especially one relying on string distance metrics alone). For linkage problems that are too big for mixed workflows ($>10^5$ observations), the work here suggests it is important to test sensitivity to linkage and hyperparameter choice. We provide some examples of how that might be done.

While the integration of the LinkedIn corpus here would seem to improve organizational match performance on real data tasks, 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 *et al.*, 2019). In such models, a latent variable indicates whether a pair of records does or does not represent a match, inferred via Expectation Maximizing using information about the agreement level for a set of variables, such as birth date, name, residence, and, potentially, employer. Information from these LinkedIn-assisted algorithms can be readily incorporated into these algorithms for estimating match probabilities on individuals.

The methods described here might also incorporate covariate information about companies. For instance, researchers can incorporate such information in the final layer of the LinkedIn-based machine learning model and re-train that layer using a small training corpus. This process, an application of transfer learning, enables extra information to be brought to bear while also retaining the rich numerical representations obtained from the original training process performed on the massive LinkedIn dataset. Finally, the approaches here are complementary to those described in Kaufman and Klevs (2022), and it would be interesting to explore possible combined performance gains.

6. Conclusion

Datasets that are important to scholars of organizational politics often lack common covariate data. This lack of shared information makes it difficult to apply probabilistic linkage methods and motivates the widespread use of fuzzy matching algorithms. Yet fuzzy matching is often an inadequate tool for the task at hand, while human coding is frequently costly, particularly if one wants human coders with the specialized domain knowledge necessary to generate high-quality matches. We have introduced a novel data source for improving the matching of organizational entities using half a billion open-collaborated employment records from a prominent online employment network. We show how this data can be used to match organizations that contain no common words or even characters.

We validate the approach on example tasks. We show favorable performance to the most common alternative automated method (fuzzy matching), with gains of up to 60 percent. We also illustrated how increased match quality can yield improved substantive insights and better statistical precision and predictive accuracy. Our primary contribution to the research community is providing a data source that can, in ways explored here and hopefully refined in future work, improve organizational record linkage while using this unique and useful corpus. □

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/psrm.2024.55. To obtain replication material for this article, https://doi.org/10.7910/DVN/APXALF

Acknowledgements. We thank Beniamino Green, Kosuke Imai, Gary King, Xiang Zhou, members of the Imai Research Workshop, and two anonymous reviewers for valuable feedback. We would also like to thank Neil Arora, Danny Guo, Gil Tamir, and Xiaolong Yang for excellent research assistance. We also thank Daniel Carpenter for making this project possible.

References

Abi-Hassan S, Box-Steffensmeier J, Christenson D, Kaufman A and Libgober B (2023) The ideologies of organized interests and amicus curiae briefs: large-scale, social network imputation of ideal points. *Political Analysis* **31**, 396–413.

Agrawal M, Hegselmann S, Lang H, Kim Y and Sontag D (2022) Large Language Models are Zero-shot Clinical Information Extractors. preprint arXiv:2205.12689.

Bolsen T, Ferraro PJ and Miranda JJ (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**, 17–30.

Carpenter D, Dagonel A, Judge-Lord D, Kenny CT, Libgober B, Waggoner J, Rashin S and Yackee SW (2021) Inequality in administrative democracy: Large-sample evidence from american financial regulation. American Political Science Association Annual Conference.

Chen H, Parsley D and Yang Y-W (2015) Corporate lobbying and firm performance. *Journal of Business Finance & Accounting* **42**, 444–481.

Clauset A, Newman ME and Moore C (2004) Finding community structure in very large networks. *Physical Review E* **70**, 1–6.

Crosson JM, Furnas AC and Lorenz GM (2020) Polarized pluralism organizational preferences and biases in the american pressure system. *American Political Science Review*. **114**, 1117–1137.

Devarriya D, Gulati C, Mansharamani V, Sakalle A and Bhardwaj A (2020) Unbalanced breast cancer data classification using novel fitness functions in genetic programming. *Expert Systems with Applications* **140**, 112866.

Enamorado T, Fifield B and Imai K (2019) Using a probabilistic model to assist merging of large-scale administrative records. *American Political Science Review* **113**, 353–371.

Eun J and Lee S-H (2021) Aspirations and corporate lobbying in the product market. *Business & Society* **60**, 844–875.

Figlio D, Guryan J, Karbownik K and Roth J (2014) The effects of poor neonatal health on children's cognitive development?. *American Economic Review* **104**, 4205–4230.

Goh S (2022) LinkDB - Exhaustive Dataset of LinkedIn People & Company Profiles. Accessed: 2024-03-02.

González JP and You HY (2024) Money and cooperative federalism: evidence from epa civil litigation. *Journal of Law, Economics, & Organization* Forthcoming.

Green B (2023) Zoomerjoin: Superlatively-Fast Fuzzy Joins. *Journal of Open Source Software* **89**, 5693.

Herzog TH, Scheuren F and Winkler WE (2010) Record linkage. *Wiley Interdisciplinary Reviews: Computational Statistics* **2**, 535–543.

Hill SJ and Huber GA (2017) Representativeness and motivations of the contemporary donorate: results from merged survey and administrative records. *Political Behavior* **39**, 3–29.

Hosseini K, Nanni F and Ardanuy MC (2020) DeezyMatch: A Flexible Deep Learning Approach to Fuzzy String Matching. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 62–69.

Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas D d. l., Bressand F, Lengyel G, Lample G, Saulnier L, Lavaud LR, Lachaux M-A, Stock P, Le Scao T, Lavril T, Wang T, Lacroix T and El Sayed W (2023) Mistral 7b, preprint arXiv:2310.06825.

Kaufman AR and Klevs A (2022) Adaptive fuzzy string matching: how to merge datasets with only one (messy) identifying field. *Political Analysis* **30**, 590–596.

Larsen MD and Rubin DB (2001) Iterative automated record linkage using mixture models. *Journal of the American Statistical Association* **96**, 32–41.

Lever J (2016) classification evaluation: it is important to understand both what a Classification metric expresses and what it hides. *Nature Methods* **13**, 603–605.

Libgober B (2020) Meetings, comments, and the distributive politics of rulemaking. *Quarterly Journal of Political Science* **15**, 449–481.

Microsoft News Center (2016). Microsoft to Acquire LinkedIn. https://news.microsoft.com/2016/06/13/microsoft-to-acquire-linkedin/.

Mikolov T, Sutskever I, Chen K, Corrado G and Dean J (2013) Distributed Representations of Words and Phrases and Their Compositionality, preprint arXiv:1310.4546.

Open secrets (2022). opensecrets.org/. Accessed: 2022-01-01.

Rasmussen AB, Buhmann-Holmes N and Egerod B (2021) The executive revolving door: new dataset on the career moves of former danish ministers and permanent secretaries. *Scandinavian Political Studies* **44**, 487–502.

Ridge JW, Ingram A and Hill AD (2017) Beyond lobbying expenditures: how lobbying breadth and political connectedness affect firm outcomes. *Academy of Management Journal* **60**, 1138–1163.

Rodriguez PL and Spirling A (2022) Word embeddings: what works, what doesn't, and how to tell the difference for aplied research. *The Journal of Politics* **84**, 101–115.

Rohe K, Chatterjee S and Yu B (2011) Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics* **39**, 1878–1915.

Ruggles S, Fitch CA and Roberts E (2018) Historical census record linkage. *Annual Review of Sociology* **44**, 19–37.

Stuckatz J (2022) How the workplace affects employee political contributions. *American Political Science Review* **116**, 54–69.

Thieme S (2020) Moderation or strategy? political giving by corporations and trade groups. *The Journal of Politics* **82**, 1171–1175.

Van Dongen S (2008) Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications* **30**, 121–141.

Wei J, Tay Y, Bommasani R, Raffel C, Zoph B, Borgeaud S, Yogatama D, Bosma M, Zhou D, Metzler D, Chi EH, Hashimoto T, Vinyals O, Liang P, Dean J and Fedus W (2022) Emergent Abilities of Large Language Models, preprint arXiv:2206.07682.

Cite this article: Libgober B and Jerzak CT (2025) Linking datasets on organizations using half a billion open-collaborated records. *Political Science Research and Methods* **13**, 923–942. https://doi.org/10.1017/psrm.2024.55