Linking Datasets on Organizations Using Half a Billion Open-Collaborated Records *

Brian Libgober † Connor T. Jerzak‡

July 3, 2024

Abstract

Scholars studying organizations often work with multiple datasets lacking shared unique 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. We show how, by leveraging information from LinkedIn regarding organizational name-to-name links, we can improve upon existing matching benchmarks, incorporating the trillions of name pair examples from LinkedIn into various methods to improve performance by explicitly maximizing match probabilities inferred from the LinkedIn corpus. 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: Record linkage; Interest groups; Text as data; Unstructured data

Word count: 9,337

*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 Gil Tamir and Xiaolong Yang for excellent research assistance.

†Assistant Professor of Political Science and Law, Department of Political Science, Northwestern University. Email: brian.libgober@northwestern.edu; URL: BrianLibgober.com; ORCID: 0000-0001-9638-0228.

‡Assistant Professor, Department of Government, The University of Texas at Austin. Email: connor.jerzak@austin.utexas.edu; URL: ConnorJerzak.com; ORCID: 0000-0003-1914-8905.

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 (Enamorado et al., 2019; Herzog et al., 2010; Larsen and Rubin, 2001; Ruggles et al., 2018). 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 sophisticated 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., Bolsen et al., 2014; Figlio et al., 2014; Hill and Huber, 2017).

Despite progress on the record linkage problem for data on political behavior, 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, nonprofits, and international associations—entities that are important players in theories of political economy, American politics, and other subfields. Similar to 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 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 interest group position-taking 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 the 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 pre-processing[1] 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 the first algorithmic approach missed at least one in three correct matches.

The challenge faced by Crosson, Furnas, and Lorenz 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 dis-

[1]The literature offers little guidance on what pre-processing steps are desirable in these cases, so researchers do their best to make the choices that seem reasonable at the time.

tance metrics can link records whose identifiers contain minor differences, they do not optimize a matching quality function 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.[2]

Methodologists have started tackling the challenges that researchers face in matching organizational records. Kaufman and Klevs (2021), 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 their approach represents an advance 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 situations 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 increasingly 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 that can serve as an asset 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 simply 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 the method 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 publicly accessible Dataverse repository (`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 data beneficial 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

[2]These problems are compounded when one attempts to link datasets from different source languages, as, for example, Chinese and English names will not share common characters. Research on organizations is hampered by such challenges connecting data sources, which impose substantial start-up costs.

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, this data did 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, we are aware of scholars who have more recent versions and also know of readily available sources for acquiring more recent updates (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.[3]

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 U_i. This URL link serves as an identifier for each organization.

Table 1: *Illustration of source data using three figures who obtained public notoriety several years after the data was collected.*

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`	20070501	20170418
Allen Weisselberg	EVP/CFO	The Trump Organization	`the-trump-organization`		20170316
Michael Avenatti	Founding Partner	Eagan Avenatti, LLP		20070101	20170318
Michael Avenatti	Chairman	Tully's Coffee	`tully's-coffee`	20120101	20170318
Michael Avenatti	Attorney	Greene Broillet & Wheeler, LLP		20030101	20070101
Michael Avenatti	Attorney	O'Melveny & Myers LLP	`o'melveny-&-myers-llp`	20000101	20030101

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 each of the 15 million aliases, on average, links to slightly more than one organizational URL. The table also notes that there are more than 10^{14} alias pairs. Besides containing many examples of alias pairs $\{A_i, A_j\}$ that are presumptively the same, as $U_i = U_j$, the database includes a large number of negative examples.

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	$> 10^{14}$

[3]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 from that time.

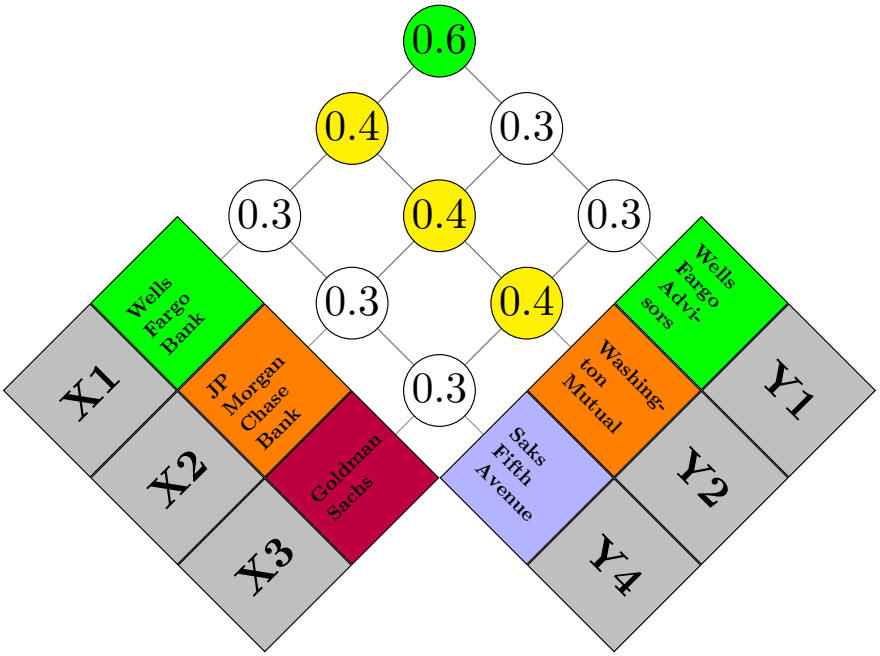


Figure 1: *Checkered flag diagram describing the organizational linkage problem.*

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 for illustration. Suppose one has 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 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 this problem. 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, one applies an algorithm that assigns scores to all the nodes and then considers a node to represent a match if it clears some numeric threshold (or, alternatively, we re-weigh links in accordance with their match probability). There are many different possible functions to score the pairs. For example, exact matching scores each node as 1 if A_i and A_j are equal, 0 if unequal, and accepts 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 a more common fuzzy matching approach. In this case, scores are calculated using the Jaccard metric. These scores present a trade-off familiar to researchers who engage with the linkage problem. If a cutoff of 0.7 is adopted, nothing matches anything. 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 uncomfortable and 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 we ultimately will propose an ensemble of two distinct approaches, we begin by discussing the intuition underlying each approach. 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 improving upon naive character similarity methods. Machine learning approaches for this particular task are now familiar, and we focus on applying and tuning these methods using the LinkedIn network. Despite the increasing sophistication of machine-learning algorithms, these methods do have fundamental limitations, as the characters and words in a name do not always have much semantic value. Indeed, one could speculate that, without specialized domain knowledge, many machine-learning algorithms would miss the connection between JPM and Washington Mutual (not to mention matches between organizations having multiple acronyms). For this reason, a more explicit attempt to leverage the LinkedIn network is desirable. In particular, an approach that utilizes community detection algorithms would complement 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 as an ensemble.

3.1 Machine Learning Approach

Machine learning continues to make so many startling 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 these match probabilities, as the rapid progress in large language models has made apparent (Jiang et al., 2023; Wei et al., 2022). 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 in terms of the speed of performing matching. String similarity algorithms, such as the Jaccard algorithm, typically require an operation on each combination of entries in set $\mathbf{X}$ and set $\mathbf{Y}$ that one wishes to match, and the calculation of a single score is potentially 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 the distance between all pairs of points is typically a much faster computation than the equivalent string distance calculation on all the pairs, especially if that calculation involves another machine learning model.

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 the outcome data; hence, the way we structure the outcome data greatly influences the ultimate algorithm we produce. Probably the simplest approach to setting up the outcome data would be to look at the set of all alias pairs $\mathcal{P} = \{A_i\} \times \{A_i\}$ 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 to modeling the data 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 as the much more common case where employees write that they work at "Goldman Sachs" and link to the Goldman Sachs page. Incorporating information about the relative number of links in the outcome is clearly desirable, but how to combine this information sensibly requires care.

To incorporate more sensitivity about the depth of ties between aliases, we borrow some ideas from the analysis of naive Bayes classifiers to calculate a probability that two aliases are indeed true matches from data. In Section A.I.1.1, we explain assumptions that would allow the interpretation of this outcome variable as a probability. Consistent with the naive Bayesian classification approach, they are indeed quite naive. However, they are less naive than ignoring the depth of ties entirely while still being easy enough to produce from a contingency table produced from Table 1. Moreover, the fact that the outcome our function seeks to learn is a probability does simplify interpretation. In particular,

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

To explicate this formula, note that for each URL u, the term $\Pr(U_i = u \mid A_i = a) \Pr(U_j = u \mid A_j = a')$ reflects the proportion of occurrences of u alongside alias a times the proportion of occurrences of u alongside alias a'. As an illustration, consider a profile URL like `LinkediIn.com/company/wellsfargo` and two aliases, "JP Morgan Chase Bank" and "Wells Fargo Advisors." Most of the time "JP Morgan Chase Bank" is used, it 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

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

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 $\widehat{\Upsilon}_{ij}$ and the inputs that we seek to optimize. The approach we employ is indebted to the methods for vector representations of words popularized in Mikolov et al. (2013). In our case, we build a model for organizational name matches from the characters on up.[4] Figure 2 provides an abstract illustration of the learning model's structure. In particular, we model each *character* as an ordered vector (with each dimension representing some latent quality of that character), each *word* as a summary of a time series representation of character vectors, and each *organizational name* as a summary of a time series representation of learned word vectors. Each step in the learning problem involves tuning an optimal representation for the next stage. 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, and we repeatedly optimize the system through backpropagation to minimize the loss function above. In this structural framework, an important parameter is the dimension of

[4]Note that this sequential approach pays far more attention to the order of characters and words than the traditional bag of words or bag of characters approaches that historically saw wide use in text analysis in political science (with more recent adoption of word vectors-type approaches (Rodriguez and Spirling, 2022)).

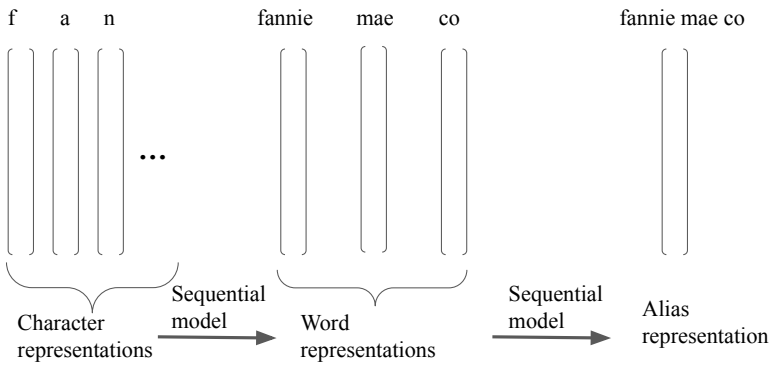


Figure 2: *A high-level illustration of the multi-level neural network's architecture. We learn from data how to represent in a vector space (a) the characters that constitute words, (b) the words that constitute organizational names, and (c) how to compare two organizational name representations efficiently.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

the vector representation. We ultimately adopt a 1,024-dimensional numeric representation of all possible names for organizations (i.e., all strings), where this choice of dimensions is an arbitrary but necessary choice. If we had chosen fewer dimensions, it would have enhanced computational efficiency with less informational richness. If we had chosen additional dimensions, then the reverse. Similar to other word embedding approaches, each dimension of the ultimate vector has a latent semantic value, although that value may be hard to interpret intuitively. For example, one dimension may relate in some ways to how specific or abstract the term is, how it positions in a hierarchy, and so forth. We will not spend much effort to interpret the dimensions, but at points we will consider the positions of particular organizations in the space into which we embed these names. A more technical and specific description of our modeling approach is found in Section of the Appendix.

In Figure 3, we examine how the algorithm has mapped a set of aliases into the space of alias embeddings. The particular organizations that we focus on are Oracle, Chase, Washington Mutual, and Goldman Sachs. As is common with such output, and due to the difficulties of visualizing multi-dimensional data, we project the high-dimensional 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 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 discrete letter combinations. While, in some situations, researchers may simply drop common, presumably low-signal words like

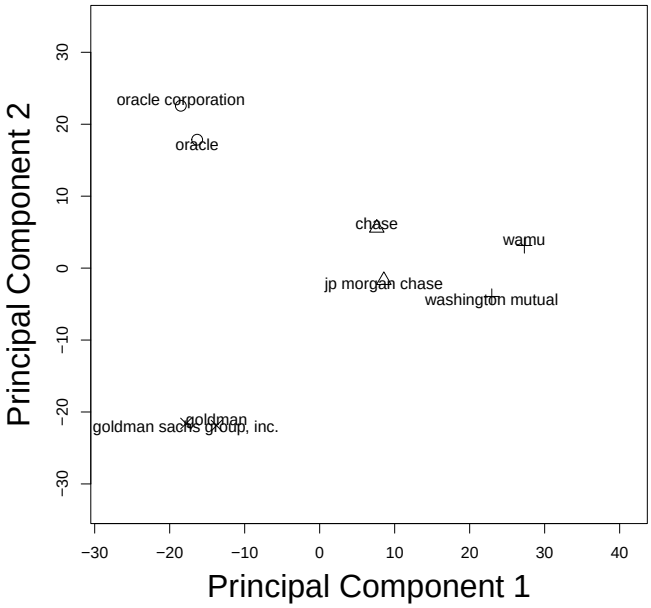


Figure 3: *Visualizing the machine-learning model output: Similar organizational names are close in this machine-learning generated vector space (which has been projected to two dimensions using PCA).*

"corporation", such choices are likely to be ad hoc, and researchers will likely feel uncertain about their justification in making choices about what words to drop. The optimized matching model learns to ignore or emphasize certain words or character combinations from the data: for example, "goldman sachs group inc." is close to "goldman".

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 2,000 randomly chosen pairs of aliases sharing a URL and 2,000 randomly chosen pairs of aliases where a URL is not shared. This is not the gold standard of hand-coded ground truth we will use in our applications that follow, but for this illustration, it suffices as the former will contain far more true matches than the latter. For each set of likely and unlikely matches we provide density plots of the match quality under fuzzy matching and under our machine-learning model.

The right panel of Figure 4 considers the predicted probability of 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. In thinking about scoring algorithms, it is basically inevitable that some pairs that do not match will score relatively well, and that some pairs that do match will score relatively badly. If the algorithm is working as it should, however, then the predicted probability of a match for those that shouldn't match will be closer to 0, the predicted probability of a match for those that should be closer to 1, and the overall distribution 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 statistic of 0.87 ($p < 10^{-16}$). If there were *total* overlap between

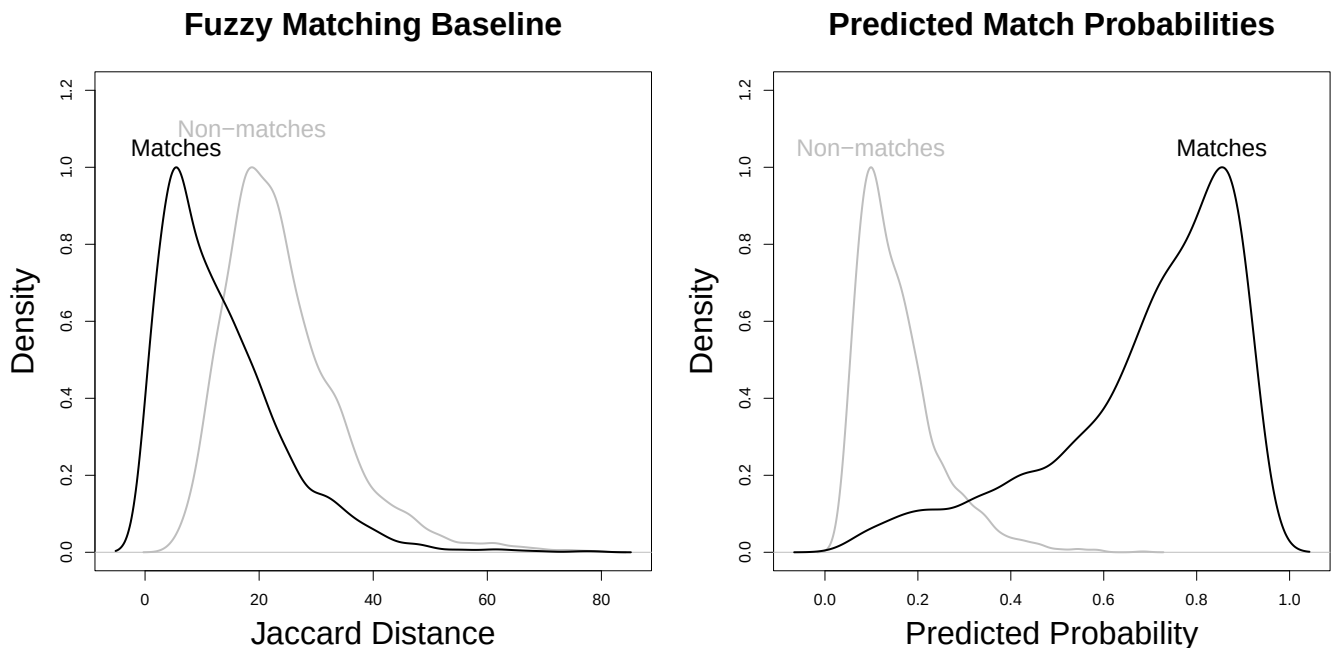


Figure 4: *Visualizing the machine learning model output:* LEFT. *Fuzzy matching as a baseline generates distances between strings that have trouble distinguishing matches from non-matches in some cases* RIGHT. *On average, organizational alias matches have higher match probabilities compared with the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

these two distributions, the test statistic would be 0. If we could perfectly distinguish matches from non-matches, the test statistic would be 1. Encouragingly, we are far closer to this second possibility. The left panel shows what the results would have looked like using fuzzy matching. A KS test on the same data using the Jaccard distance metric varies 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 relatively low match probability. To handle such difficult cases, we next show how the LinkedIn data introduced in this paper can be used in a network-sensitive approach to improve organizational record linkage.

3.2 Improving Network-based Linkage Algorithms with LinkedIn Data

As we have already seen, it is sometimes the case that there is little semantic information contained in organizational names, and as a result methods that focus on uncovering these meanings have a ceiling. Sometimes, 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 depth of knowledge in the network as fully as possible.

Instead of viewing the record linkage tasks as matching two lists of organization names directly, one can instead view the problem as connecting these names on a graph. A tricky point, of course, is that the names of organizations in one's lists may not actually be on the graph. We shall address this issue below in thinking about an ensemble strategy and also through one of our applications.

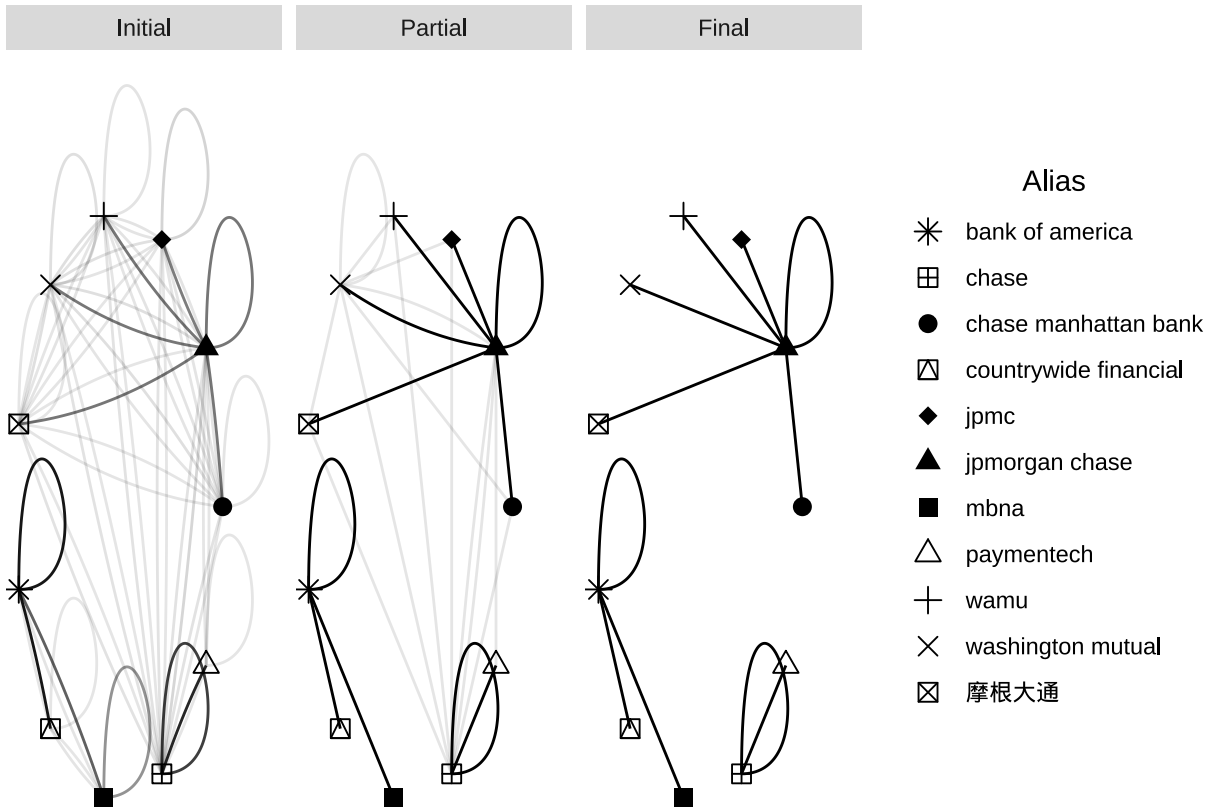


Figure 5: LinkedIn Name Network and Community Detection Algorithm*. The figure illustrates how the community detection algorithm links organizational aliases for 11 aliases in the banking space.* ***Left:*** *A weighted graph derived from raw counts of connections in the LinkedIn database. Due to some stronger connections between particular 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 is proceeding toward convergence. The connections between nodes within communities are now stronger and across communities are now weaker. 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 generated vector space (compare the positioning of some of the same aliases in Figure 3).*

But even assuming the names are on the graph, one must consider the sense of connectedness that is most useful. The simplest notion of asserting that two aliases are linked if someone has attributed the same URL to both names would often fail to have the basic property of 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 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. The problem here is that if there are spurious links, then many components will be merged where there is little evidence to do so. 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.

Figure 5 shows how we can represent the data source explicitly as a network. Here, 11 organizational aliases are presented as nodes. Aliases are connected by edges. In principle, these could be directed or undirected or weighted or unweighted depending on one's modeling strategy. The figure shows edges with weights that follow the naive Bayesian strategy for calibrating the amount of information between names and we a similar probability calculation as in Eq. 1. Under this approach, the probability that A and B is connected is the same as the probability than B and A are connected, therefore this yields a weighted, undirected graph. Notable in the context of the prior discussion about why machine learning methods struggle are the strong ties between inscrutable abbreviations such as "jpmc" and "jpmorgan chase" or "mbna" and "bank of america" (its parent), or names where the connection is surprising based on semantic information such as "chase" and "washington mutual". The comparison between the strength of ties in Figure 5 and the distance between some of the same names in Figure 3 is illustrative of where one might hope to make some performance gains. Visually, it is clear that there are 2 to 3 clusters where links are denser, but 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.

Because community detection is a well-studied problem that occurs in many different contexts (Rohe et al., 2011), we consider two algorithms established in the literature: Markov clustering (Van Dongen, 2008) and greedy clustering (Clauset et al., 2004). We focus on these algorithms because they model the network in two distinct ways and are computationally efficient. Technical details on the implementation of each clustering algorithm are in the Appendix (see Section A.I.1.5 and Section A.III.1.1) for additional information; here, we offer a brief and intuitive description of each. The Markov clustering algorithm 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."). The greedy clustering algorithm is an iterative algorithm that begins by assuming each node is its own community and then merges communities that 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 (ibid.). While the Markov clustering algorithm requires edges have probability weights, greedy clustering does not, which allows us to consider different network representations

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

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 assistance 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 ($\mathbf{X}$) with another dataset containing data about Wells Fargo Advisors, Washington Mutual, and Saks Fifth Avenue ($\mathbf{Y}$). 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 (in the figure, set to 0.5) are said to match. In this particular example, fuzzy matching could produce similar results, although 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 $\mathbf{X}$ in their proper community, $\mathbf{Y}$ in their proper community, and then we consider entries in $\mathbf{X}$ and $\mathbf{Y}$ as linked if they are placed in the same community.

This example shows how the unified approach can maximize 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 scientific practice.

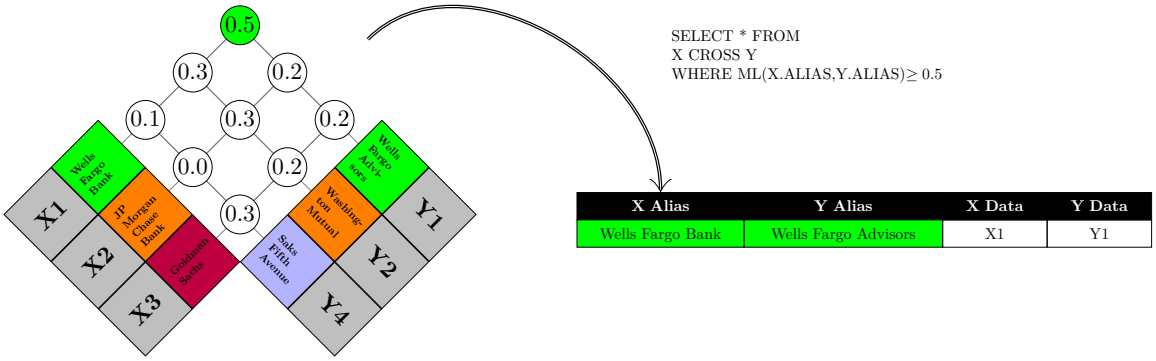
4 Illustration Tasks Using Campaign Contribution and Lobbying Data

4.1 Method Evaluation

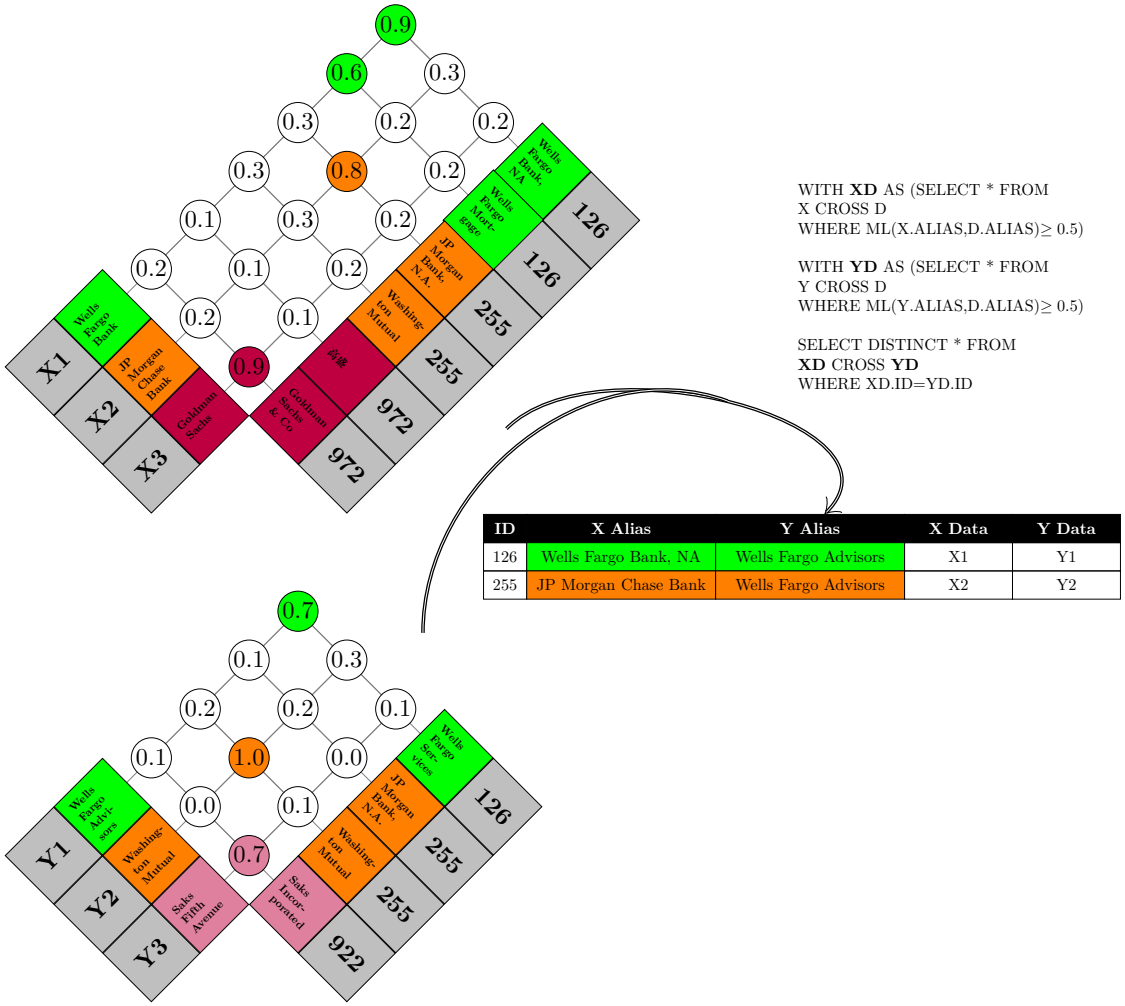
Before we describe the illustrative tasks, we first introduce our comparative baseline and evaluation metrics. This introduction will help put the performance of the methods in context.

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.



Step a) *Direct linkage through machine learning-optimized string matching.*



Step b) *Indirect linkage through directory constructed using community detection.*

Figure 6: *Checkered flag diagrams illustrating a unified approach to name record linkage using the LinkedIn corpus.*

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 all parameters 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 into the same URL at least once in the LinkedIn corpus.

4.1.4 Performance Metrics

We consider two measures of performance for each organizational matching algorithm. We will examine these measures as we vary the distance threshold for accepting a match between two aliases.[5]

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

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

This measure is important 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 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 (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 an outcome that ensures all true matches are found, but the results are not useful. 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, where the β parameter controls the relative cost of false negatives compared to false positives (Lever, 2016) and is defined as

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

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)), weighing false negatives more strongly than false positives.

[5]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 the LinkedIn network.

4.1.5 Comparing Algorithm Performance Across Acceptance Thresholds

Approximate matching algorithms have a parameter that controls how close is close enough to deem a match acceptable. Two algorithms might perform differently depending on how the acceptance threshold parameter is set. This threshold is not directly comparable across algorithms. 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 dataset 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 larger and smaller datasets. 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 on 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 7,000 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 seven hundred organizations in this fashion is not enormous, Libgober (ibid.) 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 (ibid.) 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, something achieved by both 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 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.

It is also instructive to consider an example from this linkage task where fuzzy matching failed, but the LinkedIn 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.

Another potentially important consideration in applied matching tasks is computation time. We

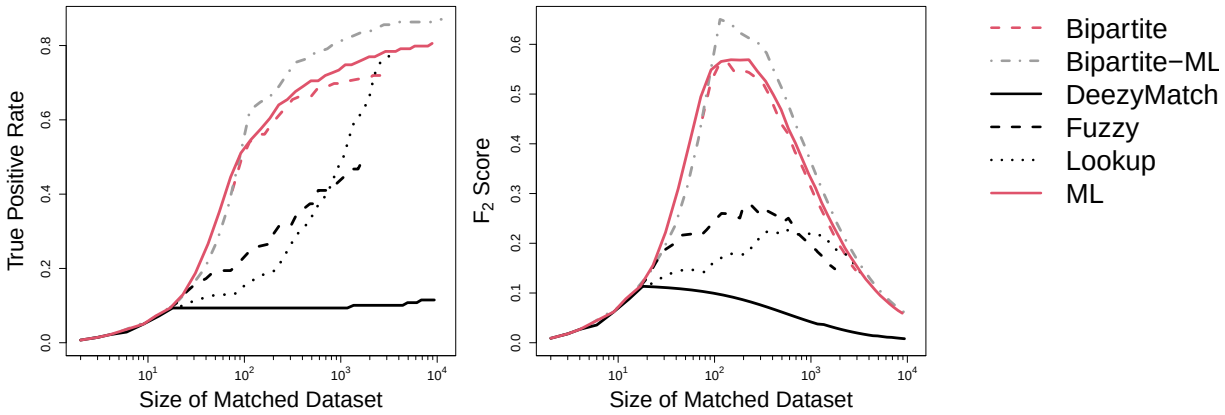


Figure 7: *We find that dataset linkage using any one of the approaches using the LinkedIn network obtains favorable performance relative to fuzzy string matching both when examining only the raw percentage of correct matches obtained (left panel) and when adjusting for the rate of false positives and 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 method baseline methods. "Bipartite-ML" refer to the ensemble of "Bipartite" and "ML". See Figure A.VII.1 for full results with Markov approaches included.*

document in Section A.VI.1.1 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, runtime is on the order of several hours for this roughly 700 by 7,000 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 minutes on 2024 hardware. Scaling the best methods is, therefore, a potential concern as one reaches datasets in the 10,000's or 100,000's of organizations. Back-of-the-envelope calculations suggest that a 10,000 by 10,000 organization match would potentially have a 2-3 day runtime using full Bipartite-ML, which is long but not unacceptable as something done once without much researcher intervention in the course of an entire project.

Additional strategies would likely be necessary to scale to a 100,000 by 100,000 name-matching problem as the best performing but slowest algorithm running somewhere around 255 days, potentially, which is too long. Two such tricks are parallelization and locality sensitive hashing. Cluster computing is known to be able to make short work of easily subdivided problems like name matching. What is unfortunate about parallelization is that the vast bulk of the computational costs are spent checking pairs that have low match probabilities. Techniques such as locality sensitive hashing have the potential to provide dramatic speed improvements by avoiding nearly redundant queries or those with low probability of success (Green, 2023).

Overall, results from this task illustrate how the LinkedIn-assisted methods would appear to yield good performance compared to a leading alternative method in the common use case when researchers do not have access to shared covariates across organizational datasets.

Table 3: *Run time on the meetings data analysis.*

Algorithm	Run Time (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

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? Common sense, and indeed prior research (Chen et al., 2015), lead 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 (Eun and Lee, 2021; Ridge et al., 2017). Our reason for focusing on an application 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 a strong prior 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 1,000 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, that is, 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 LinkedIn-assisted approaches, with the effect estimates being closer to the true value. Notice that, for all algorithms examined in Figure 8, there is significant attenuation bias in the estimated coefficient towards 0 as we increase the size of the matched dataset. The inclusion of poor-quality matches injects noise into estimation, biasing the coefficient towards 0. 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% 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.

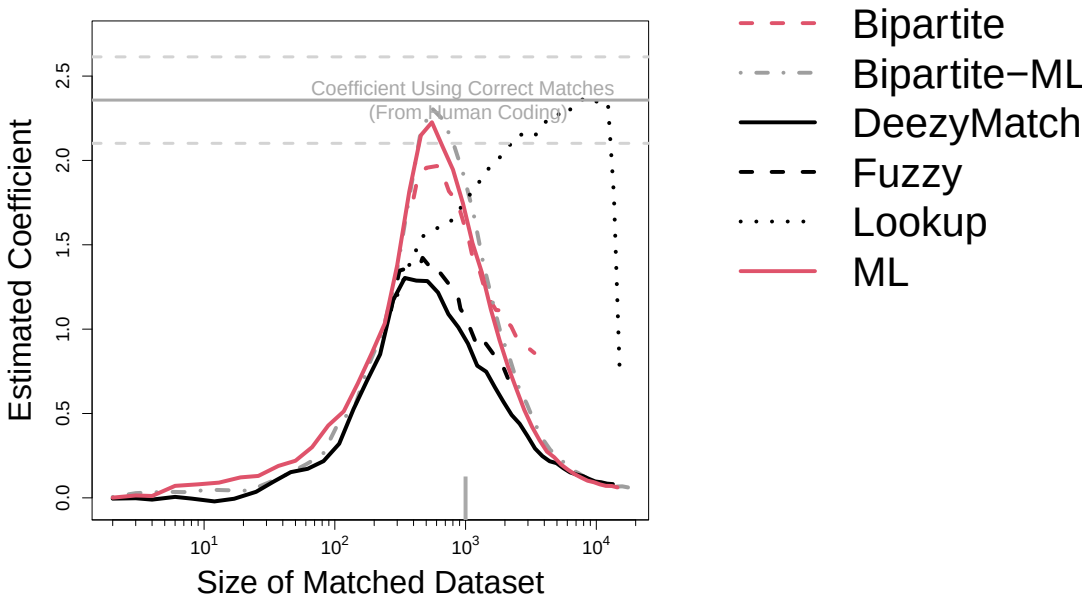


Figure 8: *The coefficient on log(Assets) for predicting log(1+Expenditures) using the ground truth data is about 2.5 (bold gray line, 95% confidence interval displayed using dotted gray lines). At its best point, fuzzy matching underestimates this quantity by about half. The LinkedIn-based matching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

4.4 Task 3: Probing Temporal Dynamics with YCombinator & PPP Data

A final question is about how these methods perform "out-of-sample", that is to say 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, future data may represent less well-known organizations that are farther removed in historical time, while our LinkedIn data from 2017 cannot directly describe organizations that did not yet exist. In this task, we analyze data from the period after the main data collection in an effort to understand the strengths and limitations of the various linkage strategies.

We seek to merge, firstly, 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' growth phase, ranging from active operations 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 Paycheck Protection Program (2020-2021) was a program aimed at providing financial relief to businesses during the COVID-19 pandemic; we subset both sets of data to target businesses within the

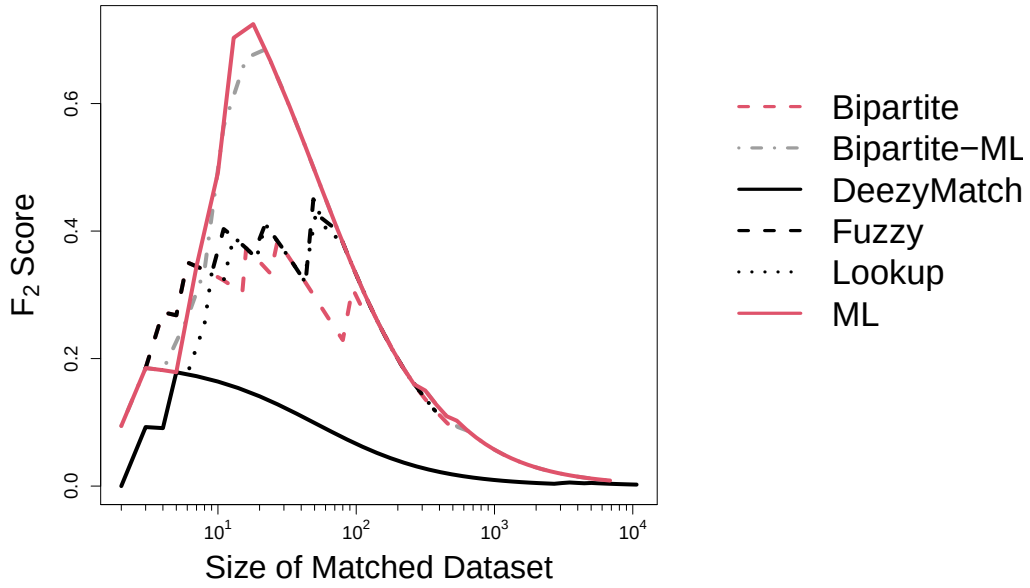


Figure 9: *In this YCombinator example, we see that the network-based approaches offer no relative benefit in terms of true positives when adjusting for false positives, yet the machine-learning-assisted approaches using the LinkedIn corpus perform well over fuzzy matching. Higher values along the Y-axis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

San Francisco area. The dataset includes entries with key financial metrics such as loan number, date approved, borrower name and address, employment impact, and the amounts pertaining to initial approval, current approval, disbursed funds, and loan forgiveness. Importantly, none of these covariates overlap with the Y-Combinator data. The task of matching startups to the PPP loan data could be important for evaluating the role of these loans on long-term firm survival, as well as thinking about the regulatory advantages that come from affiliation with a business network like YCombinator.

As expected, we find in Figure 9 that the linkage approaches that require passage through a directory of firms with established pages on LinkedIn in 2017 provide no gain in overall F_2 score relative to fuzzy matching. Likely, if our methods were rebuilt with a subsequent scrape of the LinkedIn database it would do better with these emerging organizations. That said, the machine-learning-based approach still provides a significant boost over fuzzy matching, as 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 Uses 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. Contemporary researchers studying organizations frequently find themselves in the situation where they must link datasets based on

shared names and without common covariates (Abi-Hassan et al., 2023; Carpenter et al., 2021; Crosson et al., 2020; González and You, 2024; Rasmussen et al., 2021; Stuckatz, 2022; Thieme, 2020). 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 used, either via machine learning or network detection, to improve organizational record linkage. These approaches are summarized in Table 4.

We first illustrated how the LinkedIn data as a massive training corpus for learning how to distinguish matching name pairs from non-matching pairs. We outlined assumptions on which the approach depends and illustrated how the LinkedIn data can improve linkage in a machine learning model that uses continuous, as opposed to discrete, character representations. Future research can likely improve upon the machine learning models as these approaches become more sophisticated all the time. For example, in the context of the semantic mapping assumption, new machine learning architectures could better approximate the function mapping two names into their match probability. Moreover, weighting strategies could improve the robustness of the validity assumption by up-weighting in the model training portions of the LinkedIn network most relevant to a particular dataset. For example, we have taken all user-contributed records at face value, but lower-quality records might be down-weighted and thereby improve aggregate performance. While such strategies would come at an additional computational cost, they could improve performance.

We then illustrated how the LinkedIn data can be used to build a directory of organizational name matches by using the data's intrinsic network structure. This approach captures information about how the different names interrelate, and, by finding communities in that network structure, we can find organizational name pairs without using linguistic information. 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 in a joint manner. In contrast, a unified approach predicts name matches using machine learning and builds name communities separately, although both approaches are then applied sequentially.

We apply these approaches to different research tasks drawn from substantive papers or questions. We find performance gains from incorporating the LinkedIn data into linkage. For example, in the second application, regression coefficients from the LinkedIn-assisted merges fall within the confidence interval found when using the human-coded datasets, but not when using non-LinkedIn calibrated fuzzy matching.

Even apart from the algorithms we test, our results have implications for applied researchers. In our second application, the choice of record linkage method is potentially 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% confidence interval of the estimate arrived at with 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. 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 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). We think that mixed workflows will still likely make sense with

LinkedIn-assisted approaches and expect that the higher number of true positives and a better mix of true positives to false positives that these methods provide will substantially reduce researcher costs of linking data on organizations. For those linkage problems that are too big for mixed workflows, the work here suggests it is important to do as well as possible and also test sensitivity to linkage approaches. We provide some initial 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 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 birth date, name, place of residence, and, potentially, employer. Information from these LinkedIn-assisted algorithms can be readily incorporated into these algorithms for inferring match probabilities on individuals.

The methods described here can also incorporate covariate information about companies. For instance, researchers can incorporate such information in the final layer of the LinkedIn-trained 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 (2021), and it would be interesting to explore possible combined performance gains. In short, there are numerous ways in which large-scale LinkedIn data on organizational name linkage could be useful in practice.

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 in methods matching organizations containing 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%. We also illustrated how substantive insights could be improved when match quality is increased, with better statistical precision and predictive accuracy. Our primary contribution is providing to the research community a data source that can, in ways explored here and to be developed in future work, improve organizational record linkage. Future work may attempt more sophisticated approaches to transfer learning using large language models, machine learning architecture, graph construction, community detection, and hybrid approaches while using this unique and useful corpus. □

7 Data Availability Statement

The LinkedIn data corpus is available in a Harvard Dataverse:

`doi.org/10.7910/DVN/EHRQQL`

as well as a Hugging Face repository:

`HuggingFace.co/datasets/cjerzak/LinkOrgs`

All methods are accessible in an open-source codebase available at

`GitHub.com/cjerzak/LinkOrgs-software`

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

References

Abi-Hassan, S. et al. (2023): "The Ideologies of Organized Interests and Amicus Curiae Briefs: Large-Scale, Social Network Imputation of Ideal Points". In: *Political Analysis*, no. 3, vol. 31, pp. 396–413. DOI: 10.1017/pan.2022.34.

Agrawal, Monica et al. (2022): "Large Language Models are Zero-shot Clinical Information Extractors". In: *arXiv preprint arXiv:2205.12689*.

Bolsen, Toby et al. (2014): "Are Voters More Likely to Contribute to Other Public Goods? Evidence From a Large-scale Randomized Policy Experiment". In: *American Journal of Political Science*, no. 1, vol. 58, pp. 17–30. ISSN: 15405907. DOI: 10.1111/ajps.12052.

Carpenter, Daniel et al. (2021): "Inequality in Administrative Democracy: Large-Sample Evidence from American Financial Regulation". In: American Political Science Association Annual Conference.

Chen, Hui et al. (2015): "Corporate Lobbying and Firm Performance". In: *Journal of Business Finance & Accounting*, no. 3-4, vol. 42, pp. 444–481.

Clauset, Aaron et al. (2004): "Finding Community Structure in Very Large Networks". In: *Physical Review E*, no. 6, vol. 70, p. 066111.

Crosson, Jesse M et al. (2020): "Polarized Pluralism Organizational Preferences and Biases in the American Pressure System". In: *American Political Science Review*. No. 4, vol. 114, pp. 1117–1137.

Devarriya, Divyaansh et al. (2020): "Unbalanced Breast Cancer Data Classification Using Novel Fitness Functions in Genetic Programming". In: *Expert Systems with Applications*, vol. 140, p. 112866.

Enamorado, Ted et al. (2019): "Using a Probabilistic Model to Assist Merging of Large-scale Administrative Records". In: *American Political Science Review*, no. 2, vol. 113, pp. 353–371. ISSN: 15375943. DOI: 10.1017/S0003055418000783.

Eun, Jihyun and Seung-Hyun Lee (2021): "Aspirations and Corporate Lobbying in the Product Market". In: *Business & Society*, no. 4, vol. 60, pp. 844–875.

Figlio, David et al. (2014): "The Effects of Poor Neonatal Health on Children's Cognitive Development?" In: *American Economic Review*, no. 12, vol. 104, pp. 4205–4230. ISSN: 00028282. DOI: 10.1257/aer.104.12.3921.

Goh, Steven (Sept. 2022): *LinkDB - Exhaustive Dataset of LinkedIn People & Company Profiles*. Accessed: 2024-03-02. URL: https://nubela.co/blog/linkdb-an-exhaustive-dataset-of-linkedin-members-and-companies/ (visited on 03/02/2024).

González, Juan Pablo and Hye Young You (2024): "Money and Cooperative Federalism: Evidence from EPA Civil Litigation". In: *Journal of Law, Economics, & Organization*. Forthcoming.

Green, Beniamino (2023): "Zoomerjoin: Superlatively-Fast Fuzzy Joins". In: *Journal of Open Source Software*, no. 89, vol. 8, p. 5693.

Herzog, Thomas H. et al. (2010): "Record Linkage". In: *Wiley Interdisciplinary Reviews: Computational Statistics*, no. 5, vol. 2, pp. 535–543. ISSN: 19395108. DOI: 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". In: *Political Behavior*, no. 1, vol. 39, pp. 3–29. ISSN: 01909320. DOI: 10.1007/s11109-016-9343-y.

Hosseini, Kasra et al. (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, Albert Q et al. (2023): "Mistral 7B". In: *arXiv preprint arXiv:2310.06825*.

Kaufman, Aaron R. and Aja Klevs (2021): "Adaptive Fuzzy String Matching: How to Merge Datasets with Only One (Messy) Identifying Field". In: *Political Analysis*, pp. 1–7. DOI: 10.1017/pan.2021.38.

Larsen, Michael D. and Donald B. Rubin (2001): "Iterative Automated Record Linkage Using Mixture Models". In: *Journal of the American Statistical Association*, no. 453, vol. 96, pp. 32–41. ISSN: 1537274X. DOI: 10.1198/016214501750332956.

Lever, Jake (2016): "Classification Evaluation: It Is Important to Understand Both What a Classification Metric Expresses and What It Hides". In: *Nature Methods*, no. 8, vol. 13, pp. 603–605.

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

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

Mikolov, Tomas et al. (2013): "Distributed Representations of Words and Phrases and Their Compositionality". In: *arXiv preprint arXiv:1310.4546*.

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

Rasmussen, A. Bbitex et al. (2021): "The Executive Revolving Door: New Dataset on the Career Moves of Former Danish Ministers and Permanent Secretaries". In: *Scandinavian Political Studies*, vol. 44, pp. 487–502. DOI: 10.1111/1467-9477.12214.

Ridge, Jason W et al. (2017): "Beyond Lobbying Expenditures: How Lobbying Breadth and Political Connectedness Affect Firm Outcomes". In: *Academy of Management Journal*, no. 3, vol. 60, pp. 1138–1163.

Rodriguez, Pedro L and Arthur Spirling (2022): "Word Embeddings: What Works, What Doesn't, and How to Tell the Difference for Aplied Research". In: *The Journal of Politics*, no. 1, vol. 84, pp. 101–115.

Rohe, Karl et al. (2011): "Spectral Clustering and the High-Dimensional Stochastic Blockmodel". In: *The Annals of Statistics*, no. 4, vol. 39, pp. 1878–1915.

Ruggles, Steven et al. (2018): "Historical Census Record Linkage". In: *Annual review of sociology*, vol. 44, pp. 19–37.

Stuckatz, Jan (2022): "How the Workplace Affects Employee Political Contributions". In: *American Political Science Review*, no. 1, vol. 116, pp. 54–69. DOI: 10.1017/S0003055421000836.

Thieme, Sebastian (2020): "Moderation or Strategy? Political Giving by Corporations and Trade Groups". In: *The Journal of Politics*, no. 3, vol. 82, pp. 1171–1175. DOI: 10.1086/707619.

Van Dongen, Stijn (2008): "Graph Clustering Via a Discrete Uncoupling Process". In: *SIAM Journal on Matrix Analysis and Applications*, no. 1, vol. 30, pp. 121–141.

Wei, Jason et al. (2022): "Emergent Abilities of Large Language Models". In: *arXiv preprint arXiv:2206.07682*.