Research Policy 44 (2015) 1672–1701

Contents lists available at ScienceDirect

Research Policy

journal homepage: www.elsevier.com/locate/respol

Seeing the non-stars: (Some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records

Samuel L. Ventura[a], Rebecca Nugent[a], Erica R.H. Fuchs[b,*]

[a] Department of Statistics, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, United States
[b] Department of Engineering and Public Policy, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, United States

ARTICLE INFO

Article history:
Received 21 July 2014
Received in revised form 2 December 2014
Accepted 28 December 2014
Available online 7 February 2015

Keywords:
Record linkage
Disambiguation
Patents
Supervised learning
Random forests

ABSTRACT

To date, methods used to disambiguate inventors in the United States Patent and Trademark Office (USPTO) database have been rule- and threshold-based (requiring and leveraging expert knowledge) or semi-supervised algorithms trained on statistically generated artificial labels. Using a large, hand-disambiguated set of 98,762 labeled USPTO inventor records from the field of optoelectronics consisting of four sub-samples of inventors with varying characteristics (Akinsanmi et al., 2014) and a second large, hand-disambiguated set of 53,378 labeled inventor records corresponding to a subset of academics in the life sciences (Azoulay et al., 2012), we provide the first supervised learning approach for USPTO inventor disambiguation. Using these two sets of inventor records, we also provide extensive evaluations of both our algorithm and three examples of prior approaches to USPTO disambiguation arguably representative of the range of approaches used to-date. We show that the three past disambiguation algorithms we evaluate demonstrate biases depending on the feature distribution of the target disambiguation population. Both the rule- and threshold-based methods and the semi-supervised approach perform poorly (10–22% false negative error rates) on a random sample of optoelectronics inventors – arguably the closest of our sub-samples to what might be expected of the majority of inventors in the USPTO (based on disambiguation-relevant metrics). The supervised learning approach, using random forests and trained on our labeled optoelectronics dataset, consistently maintains error rates below 3% across all of our available samples. We make public both our labeled optoelectronics inventor records and our code to build supervised learning models and disambiguate inventors (see http://www.cmu.edu/epp/disambiguation). Our code also allows users to implement supervised learning approaches with their own representative labeled training data.

1. Introduction

Disambiguation, or the process of linking records of unique individuals or entities within a single data source, is a subset of the broader "Record Linkage" field, which is generally used to link records of unique individuals or entities across multiple data sources. In 1969, Ivan Fellegi and Alan Sunter introduced the first mathematical model for record linkage (Fellegi and Sunter, 1969); this model is still the basis for many of the most common approaches to record linkage used today. In the field of technology, innovation, and entrepreneurship (TIE), record linkage and disambiguation are used to link records of assignees (the companies, organizations, individuals, or government agencies to which a patent is assigned) and, notably, to link records of inventors in the United States Patent and Trademark Office (USPTO) database. However, many USPTO disambiguation approaches fail to take advantage of the latest methodological advancements in statistics, such as adaptations of the Fellegi and Sunter (1969) approach for record linkage (e.g. Fleming et al., 2007; Lai et al., 2009). More importantly, many existing USPTO inventor disambiguation algorithms often use ad hoc weights, thresholds, and decision rules to determine which records should be linked (e.g. Lai et al., 2009) instead of leveraging information from "labeled inventor records," or USPTO inventor records for which the true identity of the inventor is known, during disambiguation. Such approaches may introduce prevalent and systematic errors in the disambiguation results, which might be avoided by leveraging information from labeled inventor records.

Using two sets of labeled USPTO inventor records from different scientific and institutional contexts (98,762 records from the field of optoelectronics consisting of four sub-samples of inventors

[*] Corresponding author. Tel.: +1 412 268 1877.
E-mail addresses: sventura@stat.cmu.edu (S.L. Ventura), rnugent@stat.cmu.edu (R. Nugent), erhf@andrew.cmu.edu (E.R.H. Fuchs).

http://dx.doi.org/10.1016/j.respol.2014.12.010

with varying characteristics (Akinsanmi et al., 2014) and 53,378 records corresponding to superstar academics in the life sciences with patents (Azoulay et al., 2012)), we make two contributions to the TIE field and the USPTO inventor disambiguation literature. First, we evaluate three commonly used inventor disambiguation approaches (Fleming et al., 2007; Lai et al., 2009, 2014), arguably representative of the range of approaches used to disambiguate USPTO inventors to-date, to determine the rates of false positive and false negative errors in their disambiguation results. These three approaches include two examples of unsupervised, rule- and threshold-based approaches: The first, Fleming et al. (2007), is similar also to past approaches such as those by Singh (2005) and Jones (2005). The second, Lai et al. (2009), is similar also to past approaches such as those by Trajtenberg et al. (2006), Lissoni et al. (2006), and Miguelez and Gomez-Miguelez (2011). We also evaluate one semi-supervised learning algorithm trained on statistically generated artificial labels, Lai et al. (2014). Second, we contribute the first supervised learning approach to the USPTO inventor disambiguation problem. Here, we build and evaluate statistical classification models for inventor disambiguation using information from the labeled inventor records to inform the algorithm. We then compare the disambiguation results of the best-performing classification model to the unsupervised and semi-supervised approaches described above. For the purposes of this study, we consider false negative errors and false positive errors to be equally unfavorable in the results of any disambiguation algorithm, though there are some contexts where one type of error may be favorable to the other (Fegley and Torvik, 2013). We define a splitting metric to assess false negatives where a single inventor is "split" into multiple inventor IDs, and a lumping metric to assess false positives where multiple inventors are "lumped" into one inventor ID. Our goal is to consistently achieve a balance of both low splitting errors and low lumping errors across the range of labeled sub-samples with different disambiguation features available to us. Here, consistent performance across contexts is equally important to balance, as a disambiguation algorithm that performs inconsistently across contexts would provide results that suggest differences across, for example, institutional or industrial contexts (or particular types of inventors) that are created by the disambiguation algorithm rather than being a reality in the original data. To summarize, we choose to pursue *consistency* across contexts and *balanced* splitting and lumping in the interest of pursuing the most generally useful disambiguation results across the wide range of research questions and contexts that might be explored using the data, rather than optimizing the results to what might be most useful to a particular context or question.

While the three past disambiguation algorithms we evaluate perform well in certain contexts, they perform inconsistently (e.g. demonstrate biases) across contexts depending on the feature distribution of the target disambiguation population. We find that the Fleming et al. (2007) has high splitting rates when evaluated against both the optoelectronics (OE) and the academic life sciences (ALS) labeled datasets. Lai et al. (2009) (based on publicly posted results where the algorithm is run on the full USPTO) relatively accurately disambiguates the set of academics in the life sciences with patents, but continues to display high splitting rates for disambiguating optoelectronic inventors. An important difference between the OE and ALS datasets is that in the ALS dataset, inventors appear to submit relatively consistent information to the USPTO (something we hypothesize may be more likely for academics and non-mobile inventors), include their middle initial, and are primarily U.S.-based. In contrast, in the optoelectronics dataset, middle names and other fields are frequently missing, and the proportion of U.S. inventors is (as in the full USPTO) only approximately half of all inventors in the sample, making it more difficult to disambiguate. The semi-supervised Lai et al. (2014) algorithm (again, based on publicly posted results where the algorithm is run on the full USPTO) at first appears to outperform all other inventor disambiguation algorithms, including slightly outperforming our supervised learning approach, when evaluated on the full optoelectronics and the full academic life sciences datasets. However, when we unpack the performance of the rule- and threshold-based methods and the semi-supervised Lai et al. (2014) algorithm on individual subsets of the OE dataset we once again find that they performs inconsistently across contexts: Specifically, they perform particularly poorly on a critical subset of the optoelectronic database – our random sample of optoelectronics inventors – which is arguably the closest of our sub-samples to what might be expected of the majority of inventors in the USPTO (based on measurable disambiguation-relevant metrics). Here, these algorithms yield splitting rates from 10% (the Lai et al. (2014) semi-supervised approach) to over 20% (the (Fleming et al., 2007) rule- and threshold-based approach). In contrast, the supervised learning approach, using random forests (Breiman, 2001) trained on the OE dataset, consistently maintains error rates below 3% across all of our available samples, including the random sample of optoelectronic inventors.

Our results suggest it important for the TIE field to continue to pursue disambiguation approaches that are consistent across disambiguation contexts with varying features. We also show that to assess past theoretical work using the disambiguated results from the algorithms evaluated in this paper or other algorithms with similar approaches, it will be important to look at the suitability of the research contexts and questions to the chosen disambiguation approach's respective strengths and weaknesses. The performance of our algorithm on additional USPTO datasets (whether other industrial and institutional contexts or the full USPTO database) is inevitably limited by the features of the labeled USPTO inventor records to which we had access. Incorporating labeled records with useful features (including detailed information on non-matches) from alternative samples will likely improve our random forests algorithm's ability to disambiguate additional USPTO datasets, since this will allow samples of records with different features to be represented and accounted for in our models. To continue to improve inventor disambiguation in the USPTO and interpretation of research leveraging the disambiguation results of past disambiguation algorithms, it will be important to continue to evaluate existing and future approaches on other sets of labeled inventor records, both to identify additional areas of potential bias in existing models upon which past papers have been based and to evaluate and improve future supervised and semi-supervised learning models used for USPTO inventor disambiguation. It is also imperative that the field moves towards requiring authors to publish as part of their theoretical papers the disambiguation approach used to generate the data upon which the theory is built, including a discussion of where that disambiguation approach may have biases.

We make public (http://www.cmu.edu/epp/disambiguation) all code and labeled inventor records for our disambiguation process, for use by both the USPTO research community and the broader disambiguation and record linkage communities.[1] Our code allows users the flexibility to specify their own blocking criteria to support applying our algorithm to databases of different size, build supervised learning models on their own labeled training data representative of their target population for disambiguation, and adjust the disambiguation results depending on their desired prevalence of false positive and false negative matching errors (in accordance with their particular research question). In providing public access not only to our algorithm but also to our extensive

[1] Several past authors have also released software for record linkage and disambiguation, including Goiser and Christen (2006), Elfeky et al. (2003), and Christen (2008), among others.

labeled dataset, we seek to enable research on disambiguation and record linkage both within and beyond the USPTO context.

2. Background

Record linkage and disambiguation are key components of any study that involves linking information across multiple data sources or within a single data source. For example, government databases, such as those maintained by the U.S. Census Bureau, often have two or more records of the same individual that should be reduced (using record linkage) to a single record for measurement purposes (Winkler, 1988; Jaro, 1989).

Modern methods for record linkage and disambiguation fall into any one of three categories: unsupervised approaches, semi-supervised learning approaches, and supervised learning approaches. Each of these three groups is defined by the way the algorithms use labeled records or "labeled training data" to link records. In the context of record linkage and disambiguation, labeled training data (also known as "labeled records" or simply "training data") is typically defined as a subset of records (from the database to be disambiguated) for which the corresponding unique entities are known (and labeled). For example, in a bibliometric database (records of publications, titles, dates, authors, etc), each author may be assigned some unique identification number (ID). Suppose we had a set of bibliometric author records for which the author ID is included on each record, and is verified to be accurate. These IDs are considered "labels" that identify the underlying unique author, and the set of author records are considered to be a set of labeled records.[2]

In the field of TIE, record linkage and disambiguation are important areas of research. The USPTO maintains an online database of all patents issued in the United States. In addition to identifying information about the patent, the database contains each patents' list of inventors and "assignees," the companies, organizations, individuals, or government agencies to which the patent is assigned. Researchers in the field seek to study the patenting characteristics of these inventors and assignees in order to make informed decisions and draw conclusions about TIE in the US and internationally (e.g. Hall et al., 2001; Singh, 2005; Fleming et al., 2007; Marx et al., 2009; Fleming and Singh, 2010; Akinsanmi et al., 2014.) However, inventors and assignees in the USPTO database are not given unique identification numbers, making it difficult to track inventors and assignees across their patents or link their information to other data sources. As a result, methods for disambiguating inventors and assignees in the USPTO database are needed. Prior methods used for disambiguating the USPTO are summarized below in Table 1. For the remainder of this work, we focus on the problem of inventor disambiguation in the USPTO database. For the remainder of this paper, we call records which refer to the same unique entity "matches" and records which refer to different entities "non-matches."

2.1. Unsupervised (including rule- and threshold-based) approaches for disambiguation and record linkage

Unsupervised learning approaches leverage statistics or machine learning techniques to try to find hidden structure in unlabeled records. For the purpose of this paper, we also group approaches that use heuristics and decision rules created by human experts (used commonly to disambiguate the USPTO) under our category of "unsupervised" approaches. In contrast to "unsupervised learning" described above, these approaches do not leverage statistics or machine learning techniques to try to find hidden structure in unlabeled data. Rather, they rely on a set of decision rules, often involving ad hoc weights, thresholds, and heuristics, to determine which records should be linked. Throughout the paper, we refer to these latter type of unsupervised methods as "rule- and threshold-based" approaches.

Fellegi and Sunter proposed the first mathematical model for linking records across two databases, or "bipartite record linkage" (Fellegi and Sunter, 1969). Typically, bipartite record linkage assumes that all records within a single data source each correspond to unique entities, and that these records can be linked to no more than one record from a secondary data source. This model is a commonly used unsupervised approach to record linkage. The Fellegi and Sunter (1969) model is the mathematical formalization of the record linkage approach previously described (qualitatively, not mathematically) by Newcombe et al. (1959) and Newcombe and Kennedy (1962). Using Newcombe's ideas, Fellegi and Sunter introduce and prove a theorem for finding the optimal linkage rule and provide two corollaries that make the theorem a practical working tool for record linkage applications.

Within a decade of Fellegi and Sunter's mathematical formalization, computer implementations of their record linkage methodology became common, and authors began analyzing the linkage accuracy and effectiveness of computers versus humans. Jaro led the computerized record linkage movement, creating "UNIMATCH," a computer system for implementing the Fellegi and Sunter (1969) record linkage model under conditions of uncertainty for applications to the US Census Bureau (Jaro, 1978). Newcombe and Smith (1975) showed that purely computerized duplicate detection can more accurately identify duplicates than methods involving both computerized procedures and manual review by trained humans by using distributional information from the data (e.g. relative commonness or rarity of names or locations) that humans cannot easily compute. Winkler later showed that computerized record linkage procedures can significantly reduce the resources needed for identifying duplicates over primarily manual record linkage methods (Winkler, 1995).

In addition to these direct implementations, several authors have published advances to the Fellegi–Sunter methodology. Winkler demonstrated that the expectation maximization (EM) algorithm can improve the calculation of weights in the Fellegi–Sunter model (Winkler, 1988). Using a linear weighting approach for the Fellegi–Sunter decision rules, Jaro also used the EM algorithm for the efficient calculation of weight parameters, applying his work to the 1985 Census of Tampa, Florida (Jaro, 1978). Winkler (1990) introduced new string comparison metrics that allow for better handling of typographical errors across fields. More recently, Sadinle and Fienberg introduce a generalization of the Fellegi–Sunter model for linking multiple (three or more) data files and offer a theoretical framework for situations in which bipartite record linkage struggles due to the lack of transitivity of pairwise matches across databases (Sadinle and Fienberg, 2013). Larsen and Rubin (2001) use mixture models for automated record linkage of two files. Bhattacharya and Getoor (2004) use information across all records in tuple comparisons to enhance the accuracy of record linkage comparisons. Each of these influential record linkage papers uses an unsupervised approach (incorporating no labeled records) to record linkage.

Approaches building on the Fellegi–Sunter framework are not the only unsupervised approaches to record linkage and disambiguation. Steorts et al. (2014) present a parametric Bayesian model for simultaneous deduplication and record linkage of multiple databases, where using a flexible new data structure they are able to estimate the attributes of the unique observable people in the population, calculate k-way posterior probabilities of matches across records, and propagate the uncertainty of record linkage into later

[2] Alternatively, labeled training data can be defined, in record linkage and disambiguation contexts, as a set of record-pairs labeled as "match" or "non-match".

Table 1
Past inventor (USPTO and EPO) disambiguation methods.

Reference	Method	Description	Available? Evaluated?
NBER: Hall et al. (2001, 2007)	Rule &threshold	String matching (assignees)	Results online
Singh (2005) Jones (2005) Fleming et al. (2007)	Rule &threshold	Exact matching, if-else decision rules	None
Lai et al. (2009) Trajtenberg et al. (2006) Lissoni et al. (2006) Miguelez and Gomez-Miguelez (2011)	Rule &threshold	Fuzzy string matching similarity scores matching thresholds SoundEx (Traj.)	Results online (Lai) Eval: (Traj.) 6023 Israeli invent. Eval: (Lissoni) Eur academ. Eval: (M&G-M) 445 Fr. academ.
Raffo and Lhuillery (2009)	Rule &threshold	Compare heuristic error rates (R&L)	Eval: (R&R) 349 Fr. academ.
Carayol and Cassi (2009)	Unsupervised	Bayesian (C&C)	Eval: (C&C) 445 Fr. academ.
Lai et al. (2009)	Semi-supervised	Extension of Torvik and Smalheiser (2009) artificial labels	Results online Some code Eval: 95 star acad.

analyses. However, this Bayesian approach grows quickly in estimation difficulty with the number of records or number of lists. Sadinle (2014) present a different Bayesian framework for deduplication and record linkage which allows for inference on the resulting linked files. This approach, however, can only use categorical similarity measures between records, which may lead to false negative errors when presented with common record linkage issues (e.g. typographical errors, name variations, etc.).

In the context of the USPTO, several unsupervised approaches exist. The National Science Foundation has funded several projects involving linking patent data to other data sources (e.g. USPTO assignees to Compustat in Zucker et al., 2011). However, the record linkage and disambiguation methods applied in TIE typically do not apply the mathematical models used in the unsupervised approaches discussed above (e.g. Fellegi and Sunter, 1969). That is, the methods described below often involve ad hoc weighting schemes, decision rules, and/or thresholds for matching. Additionally, many of these methods have not been evaluated for accuracy or bias in the results, except on small hand-disambiguated sets of labeled records in inventor disambiguation. It is important to note that we do not discuss record linkage of USPTO inventors to other data sources, although this problem is discussed by several authors (Hall et al., 2007; Bessen, 2007, 2009; Thoma et al., 2010).

The first inventor disambiguation algorithms in TIE discussed in published work use unsupervised approaches that involve simple data-cleaning and exact matching techniques for disambiguation. (See Table 1, row 2.) Jasjit Singh uses an approach involving exact string matching on comparison fields (e.g. last name and location) and if-else decision-making to determine matching record-pairs (Singh, 2005). Benjamin F. Jones uses a similar approach (Jones, 2005). Magerman et al. (2006) discuss methods for patentee name harmonization, or inventor disambiguation for European patents. Other researchers have used their own inventor disambiguation algorithms (e.g. Lim, 2012), but their methodologies and results were not published. Finally, Fleming et al. (2007) use a similar approach, incorporating information about the assignee and location of the inventors. In all cases, these inventor disambiguation results were not posted for public use.

Inventor disambiguation approaches soon began to incorporate methods such as using similarity scores and matching thresholds to link records. (See Table 1, row 3.) Before discussing these, it is worth noting that Milojevic (2013), using a simulated bibliographic dataset where the identities of the true authors were known, finds that simple algorithm that takes into account just last name and first initial when matching (and thus similar to those discussed in the previous paragraph) can achieve a high level of accuracy in author-disambiguation contexts, and, indeed, can be more accurate than other algorithms that utilize more information. It is difficult, however, to know if this simulated dataset had similar typo, missing field, and mis-spelling issues as is common in the USPTO. In the context of the USPTO, Trajtenberg et al. (2006) use the SoundEx system to group names that are similar phonetically, then use similarity scores and matching thresholds to determine pairwise matches. Trajtenberg et al. (2006) also provided the first attempt to disambiguate inventors in the full USPTO database. To evaluate the results of their methods, the authors use a large set of labeled records of inventors based in Israel, called the “Benchmark Israeli Inventors Set,” or BIIS for short. The BIIS contains 15,306 inventor records from 9155 patents, corresponding to 6023 unique Israeli inventors. Notably, the authors do not use it to train semi-supervised or supervised learning models for disambiguation. Instead, they use it to tune weight parameters in their unsupervised approach to inventor disambiguation and evaluate the results of their methods. This work is a landmark study in that it introduced many advanced disambiguation and record linkage techniques to the TIE and USPTO disambiguation literature, setting the stage for future approaches. The labeled dataset, while not public, is also extraordinary in its depth and size. Around the same time, Lissoni et al. (2006) design a method that incorporates similarity scores and matching thresholds for determining pairwise matches for European patents. They evaluate their approach on the “Keins Database,” a set of labeled inventor records corresponding to academic inventors in France, Italy, and Sweden created by the authors. Here again, they do not use these to train semi-supervised or supervised learning models for inventor disambiguation, opting instead for an unsupervised approach. This work is focused on the European Patent Office (EPO), so the labeled Keins Database unfortunately is not immediately helpful for our USPTO inventor disambiguation problem, since the EPO and USPTO have different standards for recording inventor and patent information, and an effort to translate a European inventor’s EPO patents into that inventors list of USPTO patents (if any) would require significant additional data to be error-free.

Several efforts to further improve inventor disambiguation have emerged since the Trajtenberg et al. (2006) and Lissoni et al. (2006) studies. Raffo and Lhuillery (2009) examine several heuristics for identifying unique inventors in the USPTO database. One advantage to their work is that they evaluate each approach, showing how each heuristic influences error rates. Notably, however, their evaluation group consistent entirely of (French) academic inventors. (See Table 1, row 4.) Carayol and Cassi use a Bayesian approach for disambiguating inventors of European patents (Carayol and Cassi, 2009). Although this approach is still unsupervised, the authors contribute a significant advance over previous work in formulating the problem in the context of a probabilistic model, similar to Fellegi and Sunter (1969). Carayol and Cassi (2009) also use a large benchmark dataset (again, consisting of French academic inventors) to evaluate their approach, attempting to minimize a linear combination of false positive and false negative errors in the results. The authors use the benchmark dataset solely for evaluating their algorithm's disambiguation results and not for training statistical models. (See Table 1, row 5.) Miguelez and Gomez-Miguelez (2011) disambiguate inventors of EPO patents, breaking their approach into three stages: parsing, matching, and filtering. The matching step aims at reducing the number of false negative errors, while the filtering step aims at reducing the number of false positive matching errors. The authors evaluate their results with the same benchmark dataset as Carayol and Cassi (2009).

Fleming and his collaborators use an approach similar to Miguelez and Gomez-Miguelez (2011) and Trajtenberg et al. (2006), calculating linear combinations of the similarity scores of each comparison field and using thresholds to determine pairwise matches (Lai et al., 2009). While their disambiguation results came out before Miguelez and Gomez-Miguelez (2011), we discuss them here last as they were the first to post a version of the USPTO inventor-patent database with disambiguated inventors for use in the research community. Without Lai et al. (2009) having made their algorithm and results public, our paper would not be possible. The algorithm's linear weighting approach is similar to that of Winkler (1988), Winkler (1989), and Jaro (1989), although the Lai et al. (2009) algorithm does not use the EM algorithm for weight-calculation.

Each inventor disambiguation approach mentioned here is unsupervised, since they do not train their models on information from labeled records to aid the disambiguation process. The advantage to using unsupervised approaches to record linkage and disambiguation is that labeled records, which are often very costly and/or difficult to obtain, are not required. Additionally, some unsupervised approaches do not suffer from the computational challenges that many semi-supervised and supervised approaches have. The disadvantages of unsupervised approaches, however, often outweigh these advantages. First, without in addition collecting a representative set of labeled records, it can be difficult to assess the error in the disambiguation results and the extent to which these errors are important to the subsequent research. Second, unsupervised methods that use ad hoc decision rules and heuristics to determine which pairs (or groups) of records match often suffer from systematic errors in the disambiguation results due to these heuristics.

2.2. Semi-supervised learning for disambiguation and record linkage

Semi-supervised learning approaches to record linkage and disambiguation fall between unsupervised and supervised approaches. In statistics and machine learning, semi-supervised approaches often use a small amount of labeled training data with a large amount of unlabeled training data to estimate the probability that pairs (or groups) of records refer to the same unique entity. Criminisi et al. (2011) use this approach to build semi-supervised learning models called "random forests," for example. For the purposes of this paper, we also include in the category of semi-supervised learning algorithms, semi-supervised algorithms trained on statistically generated artificial labels, such as the semi-supervised algorithms and training data developed by (Torvik and Smalheiser, 2009) and (Lai et al., 2011). Here, the statistically generated artificial labels use combinations of statistical techniques and heuristics to define which pairs of records should be considered matches or non-matches. We focus on this latter type of semi-supervised learning here, as it is the one applied in the context of the USPTO.

Torvik and Smalheiser (2009) introduce several statistical concepts in their disambiguation of authors in MEDLINE, a database of medical journal articles. They use logistic regression within a Bayesian framework to calculate matching probabilities for pairs of MEDLINE author records. A key step in this algorithm involves generating a set of record-pairs that are either "very likely" to be matches or are known to be non-matches. They do this by splitting the comparison field space into two independent subsets of comparison fields, then defining conditions on each set to identify pairs of records that are "very likely" matches or known non-matches. Then, assuming independence between the two sets, they have multiple sets of training data. This process yields a set of statistically generated artificial labels on which they can train a classification model. They then use the classifier to predict the labels (match vs. non-match) of record-pairs in MEDLINE (Torvik and Smalheiser, 2009). This approach has the benefit of providing training data with a relatively high probability of accuracy without requiring (potentially costly or inaccessible) "true" labeled data. There are also limitations. Any errors, biases, or incorrect assumptions in the label approximations would, of course, be propagated in the classification model. Additionally, if the two sets of fields are not independent (e.g. when applied to new contexts outside the original MEDLINE datasets), the training datasets will also have biases.

Lai et al. (2014) implement an adaptation of the Torvik and Smalheiser (2009) approach for disambiguating authors in the USPTO database. The Lai et al. (2014) algorithm marked the first time that a semi-supervised learning approach was used in inventor disambiguation. As with Torvik and Smalheiser (2009), the algorithm is trained on statistically generated artificial labels that are "highly likely" to be correct according to the authors' set of predefined rules for matching and not matching. As with Lai et al. (2009), the authors post the results of the algorithm online for use within the research community (Lai et al., 2011). Lai et al. (2014) in addition use a set of 1169 labeled inventor records corresponding to 95 inventors to estimate error rates the results of their algorithm. These 95 inventors were eminent U.S. academics from engineering and biochemistry fields from a manually curated dataset developed by Gu et al. (2008). To date, this is the only semi-supervised learning approach for disambiguating inventors in the USPTO database. (See Table 1, row 6.)

Semi-supervised learning approaches for record linkage and disambiguation have several advantages. First, the use of labeled training data allows for easy evaluation of the accuracy of the results. The statistical models also often provide standard errors on the evaluation metrics. Second, learning algorithms that leverage information from labeled records are often able to achieve improved performance over unsupervised approaches. The performance of these semi-supervised (and supervised) models will, however, be limited by the accuracy, usefulness of features, and representativeness of the labels, whether those labels are "true" labeled data or statistically generated artificial labels. One final potential disadvantage of semi-supervised approaches is that, similar to supervised learning, the probability predictions can be computationally intensive, depending on the size of the database.

2.3. Supervised learning for disambiguation and record linkage

Torvik and Smalheiser (2009) tackle a difficult issue that has plagued the field of record linkage and disambiguation: How do you calibrate (train) and evaluate (test) a record linkage or disambiguation algorithm in the absence of a sufficient number of labeled records? Representative sets of labeled records provide the unique opportunity to both calibrate (train) and evaluate (test) new and existing disambiguation approaches. Supervised learning approaches for record linkage and disambiguation use information from labeled records to build models which can predict whether or not pairs (or groups) of records match. Typically, pairs of labeled records are compared, and a classification model is built on these pairwise comparisons of labeled records. Similar to semi-supervised learning, the resulting classifier can be used to predict the probability that any pair of records in the database matches.

Classification models, or statistical models for categorical response variables, are a subset of supervised learning approaches. In the context of disambiguation, the response variable is “match” vs. “non-match,” a binary response variable (two categories). There are a multitude of classification models commonly used in modern statistics, such as logistic regression, probit regression, classification trees, linear and quadratic discriminant analyses, support vector machines, and random forests. For an overview of these and other classification models, see Hastie et al. (2009).

Several past authors have used classification models for record linkage and disambiguation. Elfeky et al. (2003) allow users of their record linkage software to choose both supervised and unsupervised approaches. Han et al. (2004) compare two supervised learning approaches for disambiguating publication lists from researchers' websites and 300,000 Digital Bibliographic Library Database citations. Christen (2008) uses nearest neighbor and support vector machine (SVM) classification in automated record linkage methods. Treeratpituk and Giles (2009) use supervised learning methods for disambiguating authors in MEDLINE, a database of over 15 million medical journal articles. Finally, Martins (2011) presents a supervised learning approach for duplicate detection of over 1200 geospatial dictionary and digital gazetteer records. Importantly, both Torra et al. (2010) and Abril and Navarro-Arribas (2012) show that supervised learning approaches are more accurate than rule- and threshold-based approaches in disambiguation and record linkage applications.

The advantages of supervised learning for record linkage and disambiguation is that labeled records give insight into determining the features of record-pairs (e.g. similarity of first/last names, number of shared co-inventors, etc) that lead to matches or non-matches. Training classification models on pairwise comparisons of labeled records yields classifiers which can accurately predict the probability that any pair of records matches. Supervised learning for disambiguation also has disadvantages. First, a representative set of labeled records of sufficient size with useful features can be difficult, expensive, or even infeasible to create or obtain.[3] Second, supervised learning algorithms will invariably be limited by the extent to which the labeled records are representative of the broader population to be disambiguated. Although measures can be taken to help avoid overfitting to the features of the available labeled data (such as building the algorithm on a minimum set of features), if the labeled records are not representative of the larger population, the supervised learning algorithms can be biased towards the specific set of training data used to build the models. Finally, if the features of the set of labeled records do not help to distinguish matching from non-matching record-pairs, then the resulting classification models may not perform well for disambiguation.[4]

Because population-representative sets of labeled records are difficult and/or expensive to obtain, researchers are often unable to apply supervised learning techniques to record linkage and disambiguation problems. Recently, the University of California, Irvine (UCI) Machine Learning Repository released a dataset of labeled epidemiological records called the “Record Linkage Comparisons Patterns Data Set” (2012), which provides researchers one viable dataset on which to test different record linkage and disambiguation algorithms. This publicly available dataset contains more than 5 million pairwise comparisons of epidemiological records, built using 100,000 labeled epidemiological records. To date, no sets of labeled USPTO inventor records are publicly available.

3. Methods

In this section, we first describe the USPTO patent-inventor database and the two extensive sets of labeled inventor records that we use to build and evaluate inventor disambiguation algorithms. Next, we discuss the three existing unsupervised, rule- and threshold-based algorithms (Fleming et al., 2007; Lai et al., 2009) and semi-supervised algorithm (Lai et al., 2014) evaluated in this paper. Then, we discuss our supervised learning approach to inventor disambiguation. We detail the full disambiguation algorithm, which includes blocking to ensure computational tractability and hierarchical clustering to resolve intransitive sets of pairwise matches. Finally, we describe our evaluation metrics and strategies to assess each disambiguation method. Our research framework is shown in Fig. 1.

3.1. Data

The USPTO hosts unique webpages for all of its approximately 8 million patents, identified by unique patent identification numbers. Each patent webpage has related information, including the patent ID, inventor(s) and inventor location(s), assignee(s) and assignee location(s), file and issue dates, class(es) and subclass(es), title, and abstract. As described in Lai et al. (2009) and Hall et al. (2001), data can be collected from these patent websites into one centralized USPTO patent-inventor database. Several authors have posted disambiguated versions of this data online for researchers in TIE (e.g. Lim, 2012); arguably the most extensive, methodologically transparent, and accessible versions of a USPTO patent-inventor database have been created by Lai et al. (2009, 2014).

In Table 2, we compare inventor records in the full USPTO database to our two sets of labeled inventor records using several statistics relevant to disambiguation. The first set of labeled inventor records corresponds to a sample of 824 inventors in the optoelectronics (OE) industry, defined in the USPTO context by a set of classes and subclasses described in Appendix A. We manually disambiguate the 98,762 records in this dataset using information from these inventors' curricula vitae, collected in a case study on OE inventors (Akinsanmi et al., 2014). The second set corresponds

[3] In the context of disambiguation, the goal is to maximize the extent to which the labeled dataset contains information about the true relationships among features that determine whether or not two records are a match. Important components of achieving this goal includes usefulness and distribution of features in the labeled dataset as well as the amount of information available (e.g. size) in the training dataset.

[4] Features that do not contain information about a response variable (here, whether two records are a match) will not be useful in model estimation. For example, a model would not be able to estimate the effect of gender (feature) on height (response) in a population of all women. Similarly, we would not be able to estimate the association of class with whether or not two records are a match if all of the records are of the same class.

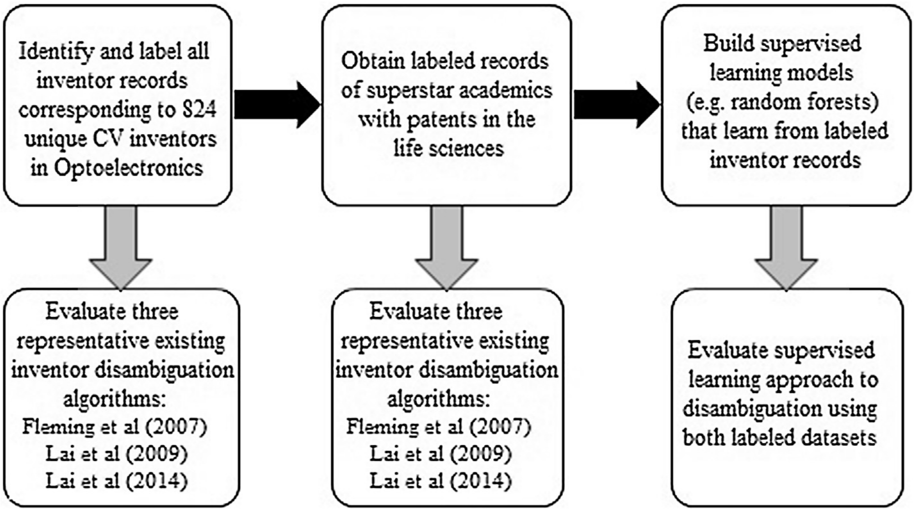

Identify and label all inventor records corresponding to 824 unique CV inventors in Optoelectronics
Obtain labeled records of superstar academics with patents in the life sciences
Build supervised learning models (e.g. random forests) that learn from labeled inventor records
Evaluate three representative existing inventor disambiguation algorithms: Fleming et al (2007) Lai et al (2009) Lai et al (2014)
Evaluate three representative existing inventor disambiguation algorithms: Fleming et al (2007) Lai et al (2009) Lai et al (2014)
Evaluate supervised learning approach to disambiguation using both labeled datasets

Fig. 1. Research framework using labeled inventor records.

to a set of 15,202 academics in the life sciences with one or more patents. This set of 53,378 labeled inventor records in the "academic life sciences" (ALS) was created as part of three separate undertakings – two published (Azoulay et al., 2007, 2012) and the third still underway – and generously provided to us by Pierre Azoulay. We describe the creation of the two labeled datasets in this paper (OE and ALS) in further detail in the following sub-sections. Our procedures are informed by the approaches for training dataset construction for disambiguation described in Bilenko and Mooney (2003a).

Each of these sets of labeled inventor records represents different samples of inventor records and patents with different feature distributions and disambiguation characteristics. For example, institutional contexts may have different standards or norms when reporting name information (e.g. inclusion of middle names/initials) and assignee information. Indeed, we find the middle names/initials field is left blank more often in our industry-dominated OE sample than in our academic life sciences sample. Depending on the industry and institutional context there may also be wide variation in the average number of co-authors. Additionally, patents in may be categorized into more technology classes and subclasses, depending on the industrial and institutional context. Each of these feature distributions and disambiguation characteristics can affect the way disambiguation algorithms behave when applied to disambiguate inventor records from these samples.

Looking at Table 2, many statistics relevant to the disambiguation process, such as average inventors per patent and average lengths of first and last names, are similar across the OE, ALS, and full USPTO datasets. Other statistics, however, vary across the three datasets. While OE has on average 17.62 patents per labeled inventor, ALS has only an average of 3.51 per labeled inventor. Note that the number of patents per inventor is unknown for the full USPTO database, since this information can only be attained via a set of labeled records. These results are fitting, in that when Akinsanmi et al. (2014) collect inventor CVs to create the labeled OE dataset, they three of their four samples are focused on prolific inventors of one form or another, while only one of their samples is a "random" sample of all OE inventors. In contrast, while (Azoulay et al., 2012) focus on "superstar" academics in the life sciences, only 10% of these superstars meet the criteria of being in the top 1% of patentors. These differences in average inventor patenting rates could lead to inconsistencies in the disambiguation results: A model trained on the ALS dataset might under-predict matches when applied to the complete OE dataset, since the average number of patents per ALS inventor is so small. Similarly, a model trained on the OE dataset might over-predict matches when applied to the ALS dataset. For percent of missing middle names and percent of U.S. inventors, the OE and full USPTO databases have similar statistics. The ALS dataset has a significantly lower percentage of missing middle names (approximately a quarter versus approximately half), and a significantly higher percent of U.S. inventors compared to the other two datasets (nearly 100% versus approximately half). This difference could likewise be influential in our results, since records with missing middle names are often more difficult to disambiguate. Finally, the percent of missing assignees and percent of last names in the Census top 200 differ across all three databases, with the OE's

Table 2
Inventor disambiguation statistics: optoelectronics, academic life sciences, and overall USPTO.

Disambiguation statistic	Optoelectronics	ALS	Overall USPTO
Number of records	98,762	53,378	9,358,182
Number of unique labeled inventors	824	15,202	NA
Inventors per patent (mean)	2.86	2.70	2.21
Classes per patent (mean)	1.87	2.08	NA
Subclasses per patent (mean)	4.33	5.38	NA
Patents per labeled inventor (mean)	17.62	3.51	NA
Assignees per labeled inventor (mean)	3.90	3.30	NA
Length of last name (mean)	5.45	6.55	6.48
Length of first name (mean)	6.09	5.72	5.84
Percent of missing middle names	48.80%	19.02%	51.10%
Percent of missing assignees	4.98%	0.00%	9.02%
Percent of United States inventors	54.30%	98.93%	50.36%
Percent of last names in census top 200	25.78%	10.56%	8.34%

Table 3
Description of CV inventor sub-samples. *Source:* Akinsanmi et al. (2014).

Sub-sample description	Sub-sample name	Population	CV sample	Response rate
Top 1.5% of OE inventors by patent total through 1999	Most	760	233	31% (73% of those reached)
Top 1.5% of OE inventors by patenting rate through 1999	Rate	680	229	34% (82% of those reached)
All OE inventors with at least one patent in 385/14, or "Integration"	Int	900	249	27% (95% of those reached)
Random sample of all OE inventors except those with at least one patent in 385/14	Rand	1250	169	14% (83% of those reached)

percent of missing assignees being slightly closer to the full USPTO's and the ALS's percent of last names in the census top 200 being quite a bit closer to the USPTO's (where both could be considered close to 10%, while the OE fraction is closer to a quarter).

3.1.1. *Labeled optoelectronics (OE) inventor records*

Our labeled OE inventor records come from a study on economic downturns, inventor mobility, and technology trajectories in OE (Akinsanmi et al., 2014). The authors collected four samples of resumes and curricula vitae (CVs) corresponding to 824 inventors in the OE industry. The target populations of the four sub-samples were as follows: top inventors by number of patents before 1999, top inventors by rate of patenting before 1999, all inventors with patents in a technological sub-field of OE corresponding to USPTO subclass 385/14, an emerging sub-field of OE called "integration" (on which Akinsanmi et al., 2014 were focused), and a random sample of all OE inventors with no patents in subclass 385/14. These sub-samples were chosen for the purposes of the research described in Akinsanmi et al. (2014), not for the purposes of inventor disambiguation. Akinsanmi et al. (2014) initially identified inventors fitting into each of their four sub-samples using disambiguation results from an adaptation of the Lai et al. (2009) algorithm. For the random sample of all OE inventors, Akinsanmi et al. (2014) ran a random number generator to select which inventors from the larger population would be contacted. They then worked with the three largest professional societies in optics to gain contact information for as many of the inventors from each population as possible. Table 3 gives basic descriptions of the four sub-samples and summarizes the response rates for inventors (reproduced from Akinsanmi et al. (2014)). Two final points are important to note regarding the labeled dataset. First, 34 of the 824 inventors that provided CVs ended up for one reason or another not fitting into the four target populations. While we include these inventor CVs in our full OE labeled dataset, these 38 CVs cannot logically be included in Table 3. We likewise follow the samples shown in Table 3 and do not include them when we later run analyses using the individual sub-samples. Second, individual inventors can fall into more than one of the four inventor populations. As such, there is some overlap between the CV samples reported in Table 3. In total, 132 of the 824 inventors were in two or more CV samples (most of which correspond to a large overlap between the two prolific inventor sub-samples). See Appendix B for the specific inventor overlaps across the four sub-samples.

When contacting inventors in these sub-samples, Akinsanmi et al. (2014) request (1) the inventor's CV and (2) a list of all patents belonging to the inventor. For inventors who could not be reached, Akinsanmi et al. (2014) also attempted to obtain their CVs from inventors' websites and LinkedIn profiles.[5] In addition to the information reported in Table 3, Akinsanmi et al. (2014) assessed potential biases in the sample of inventors who responded compared to the target population. Of those biases found, only the following is relevant in the context of inventor disambiguation: For the second sub-sample of prolific "Rate" inventors, those in our sample are more likely to be more mobile before 1999 than the broader target population. Mobile inventors may be more difficult to link across their patents, and thus to disambiguate, due to changes in their location and/or assignee information. A more detailed discussion of their sample bias assessment can be found in Akinsanmi et al. (2014).

Once the CVs and patent lists have been obtained, we create labeled inventor records in six steps: First, we manually parse and store information including each inventor's employment, location, and patenting history. Second, we generate a list of potentially matching inventor records for each CV inventor from Akinsanmi et al. (2014), or "potential matches," defined as any inventor record in the USPTO OE patent database that has a last name similarity score of at least 0.90 with the CV inventor's last name.[6] Third, we manually compare the parsed CV inventor's information to each of the potentially matching inventor records to create labels for matching and non-matching inventor records. When this is complete, each record has an identifier attached to it, indicating to which CV inventor it matched (labeled with that inventor's CV ID number) or indicating that it did not match any of the CV inventors (labeled with a "0"). [7] Fourth, we attempt to re-contact each CV inventor and ask them to verify the resulting lists of their patents. Inventors respond by indicating if there are any patents that we mistakenly assigned to them, or if there are any patents that we mistakenly did not assign to them. Fifth, in the event that the inventors are unreachable for this verification step, a random forests model is used to estimate the probability that each record is correctly labeled. Then, an independent research analyst uses these probabilities along with a combination of CV information and Internet searches to again verify each inventor's list. Finally, we remove duplicated, non-matching records and compile the resulting labeled inventor records into a single dataset.[8]

[5] The "percentage of those reached" reported in Table 3 includes only inventors reached by phone or email, and does not include CVs acquired through inventors' websites and LinkedIn profiles.

[6] The choice of 0.90 last name similarity is empirically motivated. In practice, we found no records matching to a CV inventor which did not have at least a 0.90 last name similarity score. In choosing this threshold, our goal was to maximize the probability that we had all possible matches while minimizing the number of non-matching records that needed to be verified via hand-matching against the CVs. See Appendix C for more details on similarity scores.

[7] During the labeling process, we labeled each inventor record as a match or a non-match to each labeled CV inventor. This labeling process contains no information on patents for whom we do not have labeled inventor CV data. This inability to tell whether the non-matches should be linked to each other does not affect the presented results. When we make the set of pairwise comparisons from the labeled records, we don't consider pairs where both records in the pair were not matched to one of the CV inventors. We thus do not train on pairs of non-matches. (We do, however, train on pairs where one record is a non-match to one of the 824 and the other is a match to one of the 824, so that our models have information about both matching and non-matching pairs.) Likewise, we ignore pairs of non-matches when evaluating the results of any disambiguation algorithm.

[8] Our approach to labeling inventor records – i.e., only considering records with similar last names as potential matches – ignores cases of inventors who changed their last name (e.g. after marriage). In our verification step, where we contact each inventor directly and ask them if we missed any of their patents, none of the CV

Table 4
Inventor disambiguation statistics: most, rate, integration, and random non-integration sub-samples.

Disambiguation statistic	Most	Rate	Integration	Random non-integration	USPTO
Number of records	40,380	37,972	23,056	16,407	9,358,182
Number of unique labeled inventors	564	555	480	399	NA
Inventors per patent (mean)	3.11	3.08	3.11	3.19	3.09
Classes per patent (mean)	1.85	1.85	1.87	1.88	1.87
Subclasses per patent (mean)	4.37	4.32	4.32	4.36	4.22
Patents per labeled inventor (mean)	35.87	24.09	16.09	3.67	NA
Assignees per labeled inventor (mean)	6.39	4.32	3.77	1.88	NA
Length of last name (mean)	5.43	5.59	5.21	5.72	6.48
Length of first name (mean)	6.13	6.11	6.10	6.06	5.84
Percent of missing middle names	50.84%	49.13%	43.57%	45.85%	51.10%
Percent of missing assignees	4.06%	4.93%	5.06%	4.98%	9.02%
Percent of United States inventors	57.64%	63.92%	65.60%	54.30%	50.36%
Percent of last names in census top 200	21.92%	19.27%	19.34%	25.78%	8.34%

Table 4 shows the same disambiguation statistics reported in Table 2 with the OE column broken out into each of the four sub-samples.[9] We find that the OE dataset's disambiguation statistics are remarkably consistent across each of these four sub-samples, except for the number of patents per labeled inventor. As expected, the Most and Rate sub-samples have the most average patents per inventor, with 35.87 and 24.09, respectively. Inventors patenting in the emerging technology sub-field of OE called "integration" also have a large average number of patents per inventor, with 16.09. The sub-sample of random OE inventors without patents in integration have by far the fewest average patents per inventor at 3.67, although this average is still ever so slightly higher than that of the ALS dataset (3.51), perhaps due to the ALS dataset's exclusive focus on academics in the life sciences, some of whom may have little interest in patenting.

The final hand-disambiguated dataset has 98,762 labeled inventor records; 14,520 of these records are matched to one of 824 unique CV inventors, and 84,242 fail to map to any of the 824 CV inventors. Note, however, that these 84,242 labeled non-matching records are very important, since they will help our algorithms determine the features that are associated with pairs of records failing to match. Also, note that these non-matching records were not chosen arbitrarily; they were chosen because they were similar enough to warrant examination (here, having a last name similarity score of 0.9). Only after this manual disambiguation step were they found to be non-matches.

For the purposes of model building and evaluation, the labeled OE dataset was split into two subsets – one for training (building models) and one for testing (evaluating results) the supervised learning approaches – denoted as OE_{train} and OE_{test} (with the full OE dataset being denoted as OE_{full}). There is no inventor or record overlap across the OE_{train} and OE_{test} subsets.[10] Additionally, we define sub-samples of the OE dataset, OE_{full}, as follows: the "Most" sub-sample (inventors with the most patents in OE through 1999, by number of patents), denoted as OE_{Most}; the "Rate" sub-sample (inventors with the most patents in OE through 1999, by rate of patenting), denoted as OE_{Rate}; the "Integration" sub-sample (inventors with at least one patent in the integration sub-field of OE), denoted as OE_{Int}; and the "Random Non-Integration" sub-sample (a random sample of OE inventors without any patents in integration), denoted as OE_{Rand}. Note that while some records overlap across the OE_{Most}, OE_{Rate}, OE_{Int}, and OE_{Rand} sub-samples, there is no inventor or record overlap between the test set of each of these four sub-samples ($\text{OE}_{Most\text{-}test}$, $\text{OE}_{Rate\text{-}test}$, $\text{OE}_{Int\text{-}test}$, $\text{OE}_{Rate\text{-}test}$) and OE_{train}, which is a stratified random sample of inventors from each of the four subgroups. Thus, for any inventors chosen to be in the OE_{train} group, we remove these from the test sets of the other four sub-samples for proper out-of-sample testing purposes.

Examples of labeled inventor records are shown in Fig. 2. The ID column indicates the CV inventor to which each record was matched (a 0 ID indicates no match). Of the 824 CV inventors in our sample, 216 have "common" last names according to the US Census (defined as any surname which appears in the list of the 1000 most common surnames from the 2000 US Census Bureau).

3.1.2. *Labeled records of academics in the life sciences with patents*

Although no sets of labeled USPTO inventor records are available publicly, our OE set is not the only one in existence. To help evaluate each inventor disambiguation approach, Pierre Azoulay kindly provided a set of 53,378 labeled USPTO records corresponding to a subset of 15,202 academics in the life sciences with patents (Azoulay et al., 2007, 2012).

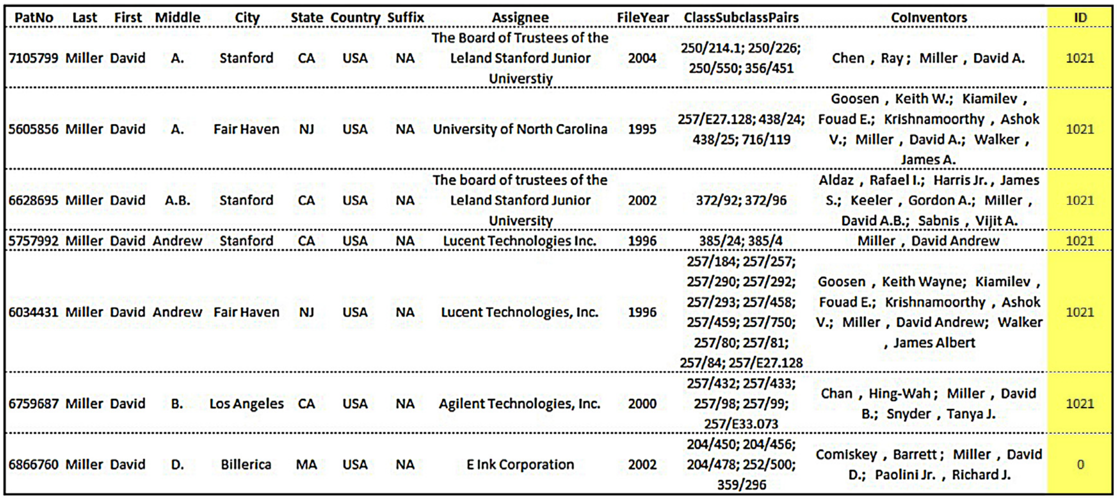
The database kindly provided by Pierre Azoulay is the compilation of three separate data collection efforts: a labeled dataset of all members of the Association of American Medical Colleges faculty who patent (Azoulay et al., 2007), a labeled dataset of elite academic life scientists (Azoulay et al., 2012), and any additional information available from a real-time data collection effort to update the information on each of these two populations to the current time. The data collection and labeling procedures for each of these datasets is described below.[11] Importantly, this labeled

inventors who responded indicated that we missed any of their patents due to last name changes. In fact, less than 0.1% of inventor records were mislabeled before our verification step.

[9] Recall that (1) these four sub-samples are not disjoint (that is, there is a substantial amount of overlap across these sub-samples) and (2) the union of these four sub-samples does not comprise the entire labeled OE dataset (that is, there are 34 inventors who did not fit into any of these four categories).

[10] To address computational scalability, we build our classification models on a random subset of all pairwise comparisons from the OE_{train} sample. We determine this size of this random subset empirically: We first examined the classification error rates of our approach when built using different-sized subsets of training data. With larger subsets of training data, the models were able to utilize more information, and the error rates were reduced. However, this effect was sub-linear in the number of training pairs. That is, in the case of OE, as long as the number of training pairs was approximately 150,000 and these pairs were a representative sampling of the larger target population to be disambiguated, the effect on error rates of adding more training pairs was minimal.

[11] While matched based on an extraordinary and careful effort that included garnering curriculum vitae information for many inventors, these datasets do not have resume or curriculum vitae information for every inventor. In contrast to



PatNo	Last	First	Middle	City	State	Country	Suffix	Assignee	FileYear	ClassSubclassPairs	CoInventors	ID
7105799	Miller	David	A.	Stanford	CA	USA	NA	The Board of Trustees of the Leland Stanford Junior Universtiy	2004	250/214.1; 250/226; 250/550; 356/451	Chen , Ray ; Miller , David A.	1021
5605856	Miller	David	A.	Fair Haven	NJ	USA	NA	University of North Carolina	1995	257/E27.128; 438/24; 438/25; 716/119	Goosen , Keith W.; Kiamilev , Fouad E.; Krishnamoorthy , Ashok V.; Miller , David A.; Walker , James A.	1021
6628695	Miller	David	A.B.	Stanford	CA	USA	NA	The board of trustees of the Leland Stanford Junior University	2002	372/92; 372/96	Aldaz , Rafael I.; Harris Jr. , James S.; Keeler , Gordon A.; Miller , David A.B.; Sabnis , Vijit A.	1021
5757992	Miller	David	Andrew	Stanford	CA	USA	NA	Lucent Technologies Inc.	1996	385/24; 385/4	Miller , David Andrew	1021
6034431	Miller	David	Andrew	Fair Haven	NJ	USA	NA	Lucent Technologies, Inc.	1996	257/184; 257/257; 257/290; 257/292; 257/293; 257/458; 257/459; 257/750; 257/80; 257/81; 257/84; 257/E27.128	Goosen , Keith Wayne; Kiamilev , Fouad E.; Krishnamoorthy , Ashok V.; Miller , David Andrew; Walker , James Albert	1021
6759687	Miller	David	B.	Los Angeles	CA	USA	NA	Agilent Technologies, Inc.	2000	257/432; 257/433; 257/98; 257/99; 257/E33.073	Chan , Hing-Wah ; Miller , David B.; Snyder , Tanya J.	1021
6866760	Miller	David	D.	Billerica	MA	USA	NA	E Ink Corporation	2002	204/450; 204/456; 204/478; 252/500; 359/296	Comiskey , Barrett ; Miller , David D.; Paolini Jr. , Richard J.	0

Fig. 2. Example of labeled inventor records.

dataset only includes information on the patents identified as belonging to individual inventors, and does not include information on patents with inventor information that could have feasibly been a match but that were determined to be non-matches, as with our labeled OE inventor records.

The primary source of the labeled academic life science data provided by Pierre Azoulay is 7874 members of the Association of American Medical Colleges faculty who hold patents (Azoulay et al., 2007). In putting together this labeled dataset, Azoulay et al. (2007) leveraged three data sources: the Association of American Medical Colleges (AAMC) Faculty Roster, which includes information on active full-time faculty from all medical schools during the period from 1981 through 2000; the NIH Consolidated Grant Application File (CGAF), which provides information on grants awarded, principal investigators and their institutions, and several project characteristics; and the database of patents issued by the USPTO during the period from 1976 through 2004. The AAMC Faculty Roster included 158,266 faculty members with an M.D. degree, a Ph.D. degree, or a joint M.D.-Ph.D. degree (7874 of whom patent during the target time period.) It also included demographic information such as the faculty members' sex, department, and experience (years since the last academic degree had been earned.) Azoulay et al. (2007) began by matching the AAMC Faculty Roster data to the USPTO data, using information on individual names, institutions, and the timing of patenting. To prevent a compromise of this matching process due to lags in reporting of affiliation changes (i.e. to avoid false negatives), they identified each faculty member whose name (or variant thereof) appeared in both the AAMC Faculty Roster and the patent database, but whose institutional information did not match. In these cases, Azoulay et al. (2007) used information from Web pages, publication history, and patents to determine whether the faculty member was in fact the same person. Likewise, to guard against false positive results, Azoulay et al. (2007) used such information to determine whether a person with a common name was in fact the same person. They made extra efforts for records containing a last name that was common to five or more faculty members in the AAMC Faculty Roster. On the basis of these matches, Azoulay et al. (2007) obtained counts of all awarded patents that each of the AAMC faculty members applied for during the period from 1981 through 2000 and were granted by 2004.

In addition, the labeled dataset provided by Pierre Azoulay includes a set of superstar academics in the life sciences who patent, collected for Azoulay et al. (2012). These superstar academics, for whom yet more information was garnered in the labeling process, are to a large extent a subset of the scientists identified for Azoulay et al. (2007). In identifying their target population, Azoulay et al. (2012) compiled a list of 12,829 elite academic life scientists. To fall into this list of superstars, the academic life scientists needed to match one of seven criteria for cumulative scientific achievement: (1) highly funded, (2) highly cited, (3) top patentors (17 patents or more by 2004 – the top 1%), (4) members of the National Academy of Sciences, (5) National Institute of Health MERIT awardees, (6) Howard Hughes Medical Investigators, or (7) early career prize winners. Of these 12,829 superstars, 3760 have one or more patent and thus become part of our labeled dataset. Of the 3760 with patents, 378 are top patentors (17 patents or more by 2004 – the top 1% by cumulative patents) and the remaining 3382 are in the data due to meeting one of the six remaining criteria for superstardom and happen to also have patented over the course of their career. As Azoulay et al. (2012) note, there are barriers to mobility in the academic life sciences, and, indeed, only 30% of their overall superstar sample (not just those that patent) transition between academic institutions between 1975 and 2004.

Azoulay et al. (2012) trace back the careers of their 12,829 superstars in the academic life sciences from the time they obtained their first position as independent investigators until 2006. They do this through a combination of curriculum vitae, National Institute of Health Biosketches, Who's Who profiles, accolades/obituaries in medical journals, National Academy of Sciences biographical memoirs, and Google searches. For each of these individuals, they record employment history, degree held, date of degree, gender, and up to three departmental affiliations. In the case of the 3760 superstars who patent, Azoulay et al. (2012) follow a series of steps to link the scientists with their patents and create the labeled records that we use here in this paper. First, they eliminated from the set of potential patents all patents issued in classes that appear unrelated to the life sciences. Next, they focused on the set of superstars with relatively rare names, and automated the match with the patent data by declaring as valid any link in which (i) the inventor's full name matches and (ii) at least one patent assignee matches with one of the scientist's employer, past or present. They then relaxed

Akinsanmi et al. (2014), who seek to collect curriculum vitae for representative samples of larger inventor populations, central to the theoretical contributions in Azoulay et al. (2007) and Azoulay et al. (2012) is having a complete population. Based on conversations with Pierre Azoulay, of the two datasets, slightly more information for labeling was available on the population of elite academic life scientists.

the constraints one at a time, examining potential matches by hand. Using knowledge about the research of the scientists from their biographical records, they then passed judgement on the validity of any more uncertain matches. They repeated the same procedure for the set of inventors with common names, with these records often requiring the inspection of each potential patent to ascertain whether they corresponded to legitimate matches.

Finally, recently Pierre Azoulay has been working to update and extend the information on both of the above populations to the current date. The labeled dataset provided by Pierre Azoulay for this project includes all additional labeled information he and his team had collected on these populations of academic life scientists as of May 7, 2014. This includes 3568 academic life scientists not in the previous datasets as well as additional information on the 11,634 academic life scientists from the previous data collection efforts.

The resulting set of 53,378 labeled inventor records (inventor-patent pairs) can be used to evaluate both our three existing unsupervised and semi-supervised approaches and our supervised learning approach to inventor disambiguation. In particular, this "academic life sciences" dataset (ALS) allows us to test our supervised learning models on a second dataset with different features than the OE dataset. For the purposes of evaluation, the ALS dataset was split into two subsets – one for training (building models) and one for testing (evaluating results) the supervised learning approaches – denoted as ALS_{train} and ALS_{test} (with the full "academic life sciences" dataset being denoted as ALS_{full}).[12]

3.1.3. *Pairwise comparisons of labeled inventor records*

In almost all record linkage and disambiguation algorithms (e.g. Fellegi and Sunter, 1969; Lai et al., 2009; Torvik and Smalheiser, 2009, etc), the operation of interest is the linking of two records, or a *pairwise comparison*. Each pairwise comparison describes the similarity of two records by a set of scores, one per shared field (see Appendix C for more detailed information on similarity scores), and if the records are labeled, an indicator of whether or not the pair corresponds to the same unique individual. Our final pairwise comparison dataset is comprised of both matches and non-matches, as to evaluate an algorithm's disambiguation results, it is equally important to understand when two records should not be linked together as when they should.

3.2. *Three examples of existing inventor disambiguation algorithms*

We evaluate three existing approaches to USPTO inventor disambiguation. First, we evaluate two rule- and threshold-based unsupervised approaches (Fleming et al., 2007; Lai et al., 2009). Next, we evaluate the only existing semi-supervised approach to USPTO inventor disambiguation (Lai et al., 2014). We choose to evaluate these algorithms for several reasons. As discussed, the algorithm used in Fleming et al. (2007) is similar to many other basic rule- and threshold-based algorithms first used by researchers in TIE. We also evaluate the more advanced Lai et al. (2009) algorithm. Lai et al. (2009) not only provide a clear description of their algorithm but also are the first to post their USPTO inventor disambiguation results online.[13] Since many TIE researchers use these disambiguation results in subsequent papers, we considered it valuable to understand what types of error, if any, are associated with them. We evaluate the Lai et al. (2014) algorithm for similar reasons, since the authors also post their disambiguation results for use by TIE researchers, and are arguably the public source for disambiguated inventor patents. The Lai et al. (2014) is also the most advanced statistical approach applied to-date to the USPTO.

The results of any disambiguation algorithm are dependent on the size of the dataset being disambiguated. The extent to which this effect exists depends on how an algorithm assigns pairwise probabilities of matching and how it handles sets of intransitive matches.

With respect to an algorithm's handling of pairwise probabilities of matching, the implications of changes in database size for disambiguation performance depend on the details of the algorithm. Depending on the rules and thresholds used, a rule- and threshold-based algorithm can be expected to have different ratios of false positives and false negatives when assigning pairwise probabilities on different sized datasets. In the case of supervised learning algorithms, random forest models typically (although not always, dependent on the training dataset and the dataset to which the algorithm is applied) yield more extreme probabilities (i.e. closer to 0 or 1) than logistic regression models. If a logistic regression algorithm matches the above characterization, the logistic regression's pairwise probabilities would lead to more false negatives and fewer false positives the larger the dataset. If a random forest algorithm matches the above characterization, the random forest algorithm's pairwise probabilities would lead to fewer false negatives and more false positives the larger the dataset.

With respect to an algorithm's handling of intransitive matches, the larger the database being disambiguated, the more information the algorithm has to link records. In large databases, there may be additional records that help link record-pairs that would otherwise be non-matches. For example, an algorithm might not link "David Miller, Fair Haven, NJ, Lucent Technologies" and "David Miller, Stanford, CA, Agilent Technologies." But in larger databases, there might be a third record, "David Miller, Stanford, CA, Lucent Technologies," that the disambiguation algorithm links to both of these records. Then, by transitivity, all three records would be linked. As such, when algorithms are run on larger databases, they may be able to better link records (i.e. avoid false negatives) than when they are run on smaller databases. In the full USPTO disambiguation, additional information from over 9 million inventor records is available, which may help algorithms avoid false negative errors that they might make if implemented on smaller subsets. This avoidance of false negative errors has the potential to increase the number of false positive errors.

We implement our versions of the Fleming et al. (2007) and Lai et al. (2009) inventor disambiguation algorithms on both the labeled OE and ALS datasets. Given that parts of the publicly posted code for the Lai et al. (2014) algorithm were not available at the time of this writing (in particular, their statistically generated artificial labels are not publicly available), we are unable to implement it on our two sets of labeled inventor records to assess its performance. We can, however, evaluate the accuracy of the Lai et al. (2014) posted results using our labeled OE and ALS inventor records. We also evaluate the accuracy of the Lai et al. (2009) posted results using our labeled OE and ALS inventor records. These posted results were generated by Lai et al. (2009) and Lai et al. (2014) running their respective algorithms on the full USPTO database – rather than on our smaller OE or ALS datasets. We evaluate the subset of posted

[12] Again, as with OE, to address computational scalability, we build our classification models on a random subset of all pairwise comparisons from the ALS_{train} sample. We again determine this size of this random subset empirically. In the case of ALS, as long as the number of training pairs was approximately 150,000 and these pairs were a representative sampling of the larger target population to be disambiguated, the effect on error rates of adding more training pairs was minimal.

[13] One component not clarified by Lai et al. (2009) is how they approach missing fields, and in particular, missing middle names. In our implementation, we assume that if the middle name fields for two records are both missing they do not match, and we redistribute the weight that would otherwise have been assigned to the middle name to the first and last name fields.

results corresponding to our labeled inventor records. As discussed earlier, the algorithms can be expected to perform differently when run on a larger dataset, such as the full USPTO. In the case of the Lai et al. (2014) algorithm, since it is a logistic regression we might expect the larger the dataset, the more false negatives and fewer false positives. It's difficult to predict how the heuristics driving the statistically generated artificial labels would affect the performance of the Lai et al. (2014) algorithm on datasets of different size. In the case of the Lai et al. (2009) algorithm, it is likewise challenging to predict how the weights and thresholds would change the performance of the algorithm on datasets of different size. There is, however, one aspect of the Lai et al. (2009) algorithm that will clearly drive differences when implementing the Lai et al. (2009) algorithm on industry- or technology-specific datasets (such as our OE and ALS labeled datasets) versus on the full USPTO: The Lai et al. (2009) algorithm puts significant weight on two records matching if the inventors have similar names and the patents include the same class. In industry- or technology-specific datasets, all of the patents, by definition, fall within a small set of classes. We would therefore expect the Lai et al. (2009) to perform better when run on the full USPTO than on these smaller industry- or technology-specific datasets. For the Lai et al. (2009) algorithm, we have the opportunity to run our implementation of the algorithm on our labeled datasets and compare its performance on these smaller datasets with the Lai et al. (2009) implementation of approximately the same algorithm on the larger dataset. When implementing Lai et al. (2009) on our labeled datasets, we run the algorithm both with the original classes rule used by Lai et al. (2009) as well as with the same rule instead applied to subclasses, to see if the latter may improve results for smaller, industry- or technology-specific datasets.

3.3. *Our supervised learning inventor disambiguation algorithm*

We execute our supervised learning approach for inventor disambiguation in two key steps: (1) For five different supervised learning (i.e. "classification") models, we predict the probability that each record-pair matches (i.e. "out-of-sample link predication"). (2) Using the pairwise probabilities of matching from the best-performing classification model from step 1, we use hierarchical clustering to identify groups of records that refer to the same unique individual. We discuss each of these steps, and then provide a detailed description of our disambiguation algorithm implementing them.

3.3.1. *Link prediction with random forests and other classification models*

In inventor disambiguation, whether or not a pair of records matches is binary (yes/no) and so can be modeled by training a supervised classification model on a set of labeled matches and non-matches. We can then use the resulting classifier to predict whether pairs of unlabeled inventor records match. There are several standard classification models that could be used for inventor disambiguation, including linear discriminant analysis, quadratic discriminant analysis, classification trees, logistic regression, and random forests (Hastie et al., 2009). Criminisi et al. (2011) present a unified framework for applying random forests to many statistical tasks, such as classification, regression, and density estimation, among others.[14] Partitioning our labeled records into training and testing subsets, we evaluate the performance each of the five standard classification models at predicting whether pairs of labeled inventor records match (i.e. out-of-sample link prediction).

Classification trees and random forests can be advantageous in the context of supervised learning for disambiguation since they enable a user to adjust important tuning parameters (such as the smallest allowed node size or restrictions on the within-node deviance) to help avoid overfitting to a particular set of training data (Hastie et al., 2009). Random forests combine results from an ensemble of "classification trees" to predict the class of a categorical outcome variable (here, a match or non-match). A classification tree builds a decision tree from the selected features by determining cut-points in the features that best separate matches from non-matches. Each classification tree in the random forest is built using a random set of features and returns a predicted class for each pairwise comparison. The predicted class from the random forest is the class that receives the majority of the votes of the individual trees (Breiman, 2001). We can obtain a predicted probability of matching from random forests by dividing the number of underlying trees that predict "match" by the total number of underlying trees. If this probability is greater than 1/2 (or a different specified threshold), the random forest predicts a pairwise match; otherwise, it predicts a pairwise non-match.

3.3.2. *Hierarchical linkage clustering to resolve transitivity violations*

Clustering is an approach used commonly in statistics and machine learning to find groups of similar observations within a dataset. Generally, clustering algorithms seek to place observations with high similarity (low dissimilarity or small distance) into the same group, or "cluster," while splitting observations with low similarity (high dissimilarity/large distance) into different clusters (Hartigan, 1975). In the context of deduplication, observations are the n records in the database, and the resulting clusters are groups of records corresponding to unique entities. We use a clustering approach called "Hierarchical Linkage Clustering" to determine – given each record-pair's probability of matching – which groups of records refer to the same unique individual (Hartigan, 1975).

Hierarchical linkage clustering relies on the existence of a distance matrix. Note that we use "distance" to represent either a distance or dissimilarity measure. Given a set of n observations, the distance matrix is a data structure containing the distances between all $\binom{n}{2}$ pairs of observations in the data. We denote the distance between observations x_i, x_j as $D_{[i,j]} = d_{ij} = dist(x_i, x_j)$, $\forall\, i, j \in \{1, 2, \ldots, n\}$ s.t. $1 < j < i < n$ (Hartigan, 1975). Note that in this context our distances are symmetric, $D_{[i,j]} = D_{[j,i]}$. For disambiguation, these pairwise distances are inverse transformations of the probability of matching for each pair of records.

The results of hierarchical linkage clustering are actually a set of clusterings described and visualized by a dendrogram. The dendrogram can be "cut" at a given distance level or height τ to identify a set of clusters; any pair of observations which are linked at a distance lower than τ are considered to be in the same cluster. As τ increases, the number of clusters decreases. (Note: choosing an appropriate threshold τ for hierarchical clustering is considered an open problem and is not the focus of this work.) In the disambiguation context, adjusting τ is equivalent to making the probability threshold for deciding whether a pair of records match more or less strict. We can then tune the disambiguation results to desired levels of false positive and false negative errors in the results.

Several TIE authors have used hierarchical linkage clustering or a mathematically equivalent approach to resolve pairwise

[14] Criminisi et al. (2011) also present a method for using random forests in a semi-supervised learning context. However, this characterization of semi-supervised learning is slightly different than that of Lai et al. (2014). That is, Criminisi et al. (2011) first learn from labeled data, then incorporate information from unlabeled data using their initial models.

transitivity violations in disambiguation (e.g. Tang and Walsh, 2010; Lai et al., 2014). Lai et al. (2014) calculate the pairwise matching probabilities for all record-pairs and choose a threshold for determining pairwise matches. They then enforce transitivity of pairwise matches to ensure that no intransitive triplets of records remain (Lai et al., 2014).[15]

3.3.3. Random forests inventor disambiguation algorithm

Given a random forests classifier trained on pairwise comparisons of our labeled inventor records, our algorithm for inventor disambiguation works as follows:

1. *Let X be the set of n original records to be disambiguated.*
2 *Determine a blocking rule and partition X into B blocks of records, X_b, $b = 1, \ldots, B$.* A blocking rule partitions the data into homogeneous groups of records which share some basic feature or set of features. In this application, we block on last name (exact match of first three letters). That is, only pairs of records which share the first three letters of their last names will be compared. To further reduce computational time, an additional level of blocking (blocking on the first three letters of the first name) is imposed on blocks with more than 1000 records. Preliminary analyses using our labeled OE dataset show that the number of disambiguation errors introduced by this blocking scheme is extremely small. For more details on blocking, see Appendix E.
3. *Within each block of records X_b:*
   (a) *Quantify the similarity of each pair of records in X_b.* We use several methods in this step to help reduce computation time. First, we use parallelization techniques to help calculate the independently calculated similarity scores (e.g., on different fields) more quickly. Second, we calculate the similarity score for each unique text string comparison only once, and then store and reference it for later use. For example, the comparison of first names “Dan” and “Daniel” occurs several times throughout the paper. Instead of re-calculating at every occurrence, we store the similarity after its first use and reference this stored value for subsequent uses, yielding substantial reductions in computation. For more details on the specific set of similarity scores used, see Appendix C.
   (b) *Calculate the predicted probability of matching, $\hat{p}_{ij}$, for all pairs of records x_i, x_j in X_b using the Random Forests classifier.* After this step, we now have predicted probabilities of matching for each pair of records in X_b.
   (c) *Convert each probability estimate, $\hat{p}_{ij}$, to an estimate of the distance/dissimilarity between each pair of records by letting $\hat{d}_{ij} = h(\hat{p}_{ij})$.* Here, h can be any smooth, continuous, and monotonically decreasing function. Some examples are $h(x) = 1 - x$, $h(x) = -\log(x)$, $h(x) = e^{-x}$, $h(x) = 1/(1 + x)$. For our purposes, since the probabilities are defined on [0,1], we use $h(x) = 1 - x$ so that the resulting distances are also defined on [0,1]. Using this transformation, a distance of 0 corresponds to a pairwise matching probability of 1, or a perfect match; a distance of 1 corresponds to a pairwise matching probability of 0, or a definite non-match.
   (d) *Calculate the single linkage hierarchical clustering solution corresponding to the distances from the previous step.* Again, we choose single linkage to enforce transitivity among record pairs. Clusters are determined by cutting the dendrogram tree at a level $\tau = 0.5$. We choose $\tau = 0.5$ because it corresponds to an intuitive probability threshold and, empirically, yielded (approximately) the best disambiguation results of all thresholds.
4. *Combine the clustering results across blocks to yield a final set of disambiguated inventor IDs.* After Step 3, records which are assigned to the same cluster in each block are considered “duplicates” belonging to the same unique entity (inventor) and are given identical inventor IDs. Records in different clusters in the same block or in different blocks are assigned to different unique entities (inventors).

3.4. Evaluation metrics

We evaluate the performance of each algorithm (Fleming et al., 2007; Lai et al., 2009, 2014) and our five supervised learning algorithms) with error metrics characterizing the numbers of false positive and false negative errors in the results. To conduct this evaluation, we develop a revised version of the evaluation metrics developed by Torvik and Smalheiser (2009) and used by Lai et al. (2014). Lai et al. (2014) use the Torvik and Smalheiser (2009) interpretation of the error metrics “splitting” and “lumping” to evaluate their algorithm’s disambiguation results. The terms “splitting” and “lumping” are intuitive terms describing possible errors. Lumping occurs when multiple unique inventors are given a single unique inventor ID (“lumped” into a single ID). Splitting occurs when a single unique inventor is given multiple inventor IDs (“split” across multiple IDs). The precise mathematical definitions are given below.

For each unique inventor in a set of labeled records, let the number of split records be defined as the number of records that the disambiguation algorithm fails to map to that inventor’s largest cluster of records. Then (Lai et al., 2014):

$$Splitting = \frac{Total\ \#\ of\ Split\ Records\ for\ All\ Inventors}{Total\ \#\ of\ Labeled\ Records} \qquad (1)$$

For each unique inventor in a set of labeled records, let the number of lumped records be defined as the number of records that the disambiguation algorithm incorrectly mapped to that inventor’s largest cluster of records. Then (Lai et al., 2014):

$$Lumping = \frac{Total\ \#\ of\ Lumped\ Records\ for\ All\ Inventors}{Total\ \#\ of\ Labeled\ Records} \qquad (2)$$

3.4.1. False negative and false positive error metrics

We use a revised version of the Lai et al. (2014) and Torvik and Smalheiser (2009) metrics in our evaluation for two reasons. First, the above metrics focus only on the largest cluster of records, ignoring the number and size of all the different clusters corresponding to a unique entity. For example, there may be another cluster of similar size for the same inventor, but these metrics do not take that cluster into account. Second, the above metric uses the number of incorrectly assigned inventor records as the unit of measure. We instead choose to evaluate all pairwise comparisons of inventor records made by the disambiguation algorithm rather than the assignment of the records themselves. We create a contingency table of the true pairwise labels (match or non-match) and the predicted pairwise labels and then evaluate our results in terms of false positive and false negative pairwise comparisons.

[15] In Appendix D, we show that this approach is equivalent to the hierarchical linkage clustering approach that we use in our random forest model – single linkage hierarchical clustering. Single linkage hierarchical clustering is faster computationally due to available clustering algorithms such as Prim’s for quickly finding the minimum spanning tree of a set of distances (Hartigan, 1975).

Thus, we define the following versions of splitting and lumping:

Splitting: A single unique inventor is “split” into multiple inventor IDs

$$\begin{aligned} Splitting &= \frac{\#\ of\ comparisons\ incorrectly\ labeled\ as\ non-matches\ across\ all\ inventors}{Total\ \#\ of\ pairwise\ true\ matches} \\ &= \frac{\#\ of\ False\ Negatives}{\#\ of\ True\ Positives + \#\ of\ False\ Negatives} \end{aligned} \tag{3}$$

Lumping: Multiple unique inventors are “lumped” into one inventor ID

$$\begin{aligned} Lumping &= \frac{\#\ of\ comparisons\ incorrectly\ labeled\ as\ matches\ across\ all\ inventors}{Total\ \#\ of\ pairwise\ true\ matches} \\ &= \frac{\#\ of\ False\ Positives}{\#\ of\ True\ Positives + \#\ of\ False\ Negatives} \end{aligned} \tag{4}$$

Thus, splitting is a measure of the prevalence of false negative matches, and lumping is a measure of the prevalence of false positive matches. We will use these metrics for all algorithms throughout the remainder of this paper.

For the purposes of the analyses presented in this paper, the goal is to simultaneously minimize both the splitting and lumping metrics, so as to avoid as many false positive and false negative errors as possible. As such, we prefer *low, balanced* splitting and lumping errors. Note that in some contexts, minimizing one particular type of error may be favorable. For example, in the context of the USPTO and the PubMed database, Fegley and Torvik (2013) examine the effect that splitting and lumping errors from disambiguation each have on co-authorship network metrics. They show that splitting (false negative error rate) has very little effect on network measures, but lumping (false positive error rate) can substantially change many important network measures. For the purposes of this paper, in the interest of producing the most widely useful output for public consumption, we set our goal at producing disambiguation results that are application- or research-context agnostic. As such, we evaluate three representative past algorithms and new supervised learning algorithms with the goal of minimizing and *balancing* both types of error.

3.4.2. Out-of-sample testing and other methods to help avoid overfitting for supervised learning approaches

We take several measures to help avoid overfitting our classification models to the labeled inventor records dataset (i.e. to improve the chances that our model will make accurate predictions on any dataset, not just this training data). First, we use only a small, basic set of explanatory variables to model the match versus non-match outcome of a pair of records: We use the similarity scores described in Appendix C for the last, first, middle, and suffix names; assignee name; city, state, and country locations; list of co-inventors; and lists of classes and subclasses.[16] Second, we use out-of-sample testing when calculating all classifier error metrics. Third, to support future researchers using our disambiguation code in deciding the value of collecting additional outside labeled data for their particular disambiguation context and the features that may be important for that labeled data, we evaluate how our supervised learning approach performs if trained on a labeled dataset with different features than the target population for disambiguation (specifically, trained on OE and applied to ALS, trained on ALS and applied to OE, and trained on mixes of OE and ALS and applied to record samples non-overlapping with the training data of each). Finally, we utilize feature differences across our OE sub-samples to evaluate how our supervised learning approach trained on a sample of records from the full OE dataset corresponding to a mix of each of the four sub-samples performs on inventors from different samples (i.e. a randomly generated sample of inventors, two forms of prolific inventors, and inventors in the field’s emerging technology). Here, our sample of CVs corresponding to a randomly generated sample of all OE inventors is potentially most valuable both with respect to potentially being most similar of our available labeled datasets to the distribution of disambiguation-relevant features in the USPTO and in helping reduce the likelihood that our methods would be biased towards disambiguating prolific OE inventors or OE inventors from specific institutional contexts (e.g. academia versus firms versus government).

Supervised learning models are by definition tailored to the data on which they are trained. To avoid overfitting to our training data, we split both sets of labeled inventor records into training and testing subsets. The training subset (e.g., OE_{train}) is used to build the supervised learning model, and the testing subset (e.g., OE_{test}) is used to evaluate the efficacy of the model. That is, we only evaluate how a model trained on one subset performs when applied to another subset. This method is known as “out-of-sample testing,” and is well-documented in statistics literature, and our out-of-sample testing approach is similar to cross-validation (Hastie et al., 2009).

For the labeled optoelectronics data set, nearly half of our sample consists of prolific inventors (according to one of two definitions of prolific: top 1.5% of OE inventors by total patents up through 1999 and top 1.5% of OE inventors by average patents per year up through 1999). This large proportion of prolific inventors could cause our algorithm to perform less well at disambiguating less prolific inventors. To address this issue, we assess the robustness of our outcomes if we instead train our algorithm on a set of records from a mix of the four OE sub-samples (OE_{train}) and evaluate our algorithm on sets of records from each of the four sub-samples ($OE_{Most\text{-}test}$, $OE_{Rate\text{-}test}$, $OE_{Int\text{-}test}$, and $OE_{Rand\text{-}test}$) that do not share any records or inventors with OE_{train}. We are particularly interested in the efficacy of each disambiguation approach when applied to the randomly chosen subset of inventors ($OE_{Rand\text{-}test}$), which are likely to be most representative of the full range of records in the full USPTO database.

Finally, one type of bias remains unresolved by the above out-of-sample testing procedure, bias towards the feature distributions or characteristics of the labeled data. In an ideal world, a supervised or semi-supervised model would be trained on a set of labeled data that both has useful disambiguation features (e.g. fields that help the algorithm what determines what is and what is not a match) and is representative of the broader population to be disambiguated. In practice, developing a representative labeled dataset with useful disambiguation features can be costly or impossible to

[16] Supervised algorithms are less prone to overfitting when trained using a small set of features. Generally, the more observations available in the training data the more features can be added without overfitting. It is also possible in supervised learning to choose a set of features by analyzing the importance of each feature in determining a match. We do not take this approach to help avoid overfitting our model to our labeled training data. Bramer (2007) provide a more detailed discussion of overfitting decision tree-based models (e.g. classification trees and random forests) to training data. Note that Lai et al. (2014) use a different, larger set of features in their semi-supervised logistic regression models, including interactions between and transformations of existing similarity scores.

achieve. Past research shows that different industries may, depending on the industry, have different features (Bound et al., 1984; Klevorick et al., 1995; Cohen et al., 2000). We expect, however, the largest differences, with respect to features relevant to disambiguation, will be across institutional contexts (e.g. academic versus firm versus government) and inventor characteristics (e.g. prolific versus less prolific, highly mobile versus less mobile) A model trained on labeled data with a particular feature distribution may perform poorly disambiguating a population for which those features are not representative. This bias can be present in any algorithm that learns information from training data whether semi-supervised learning algorithms such as Torvik and Smalheiser (2009) or Lai et al. (2011) or supervised learning algorithms such as the ones in this paper). One goal in developing disambiguation algorithms is to make the algorithm as robust as possible to alternative contexts. Regardless, when evaluating the costs and benefits of collecting additional labeled data for disambiguating a new context, it is important to understand what the algorithm's performance is when trained on a labeled dataset with different feature distributions or different feature importance in determining matches. To support future researchers using the code and labeled inventor records associated with our disambiguation approach, we assess in this paper the performance of our supervised learning approach when trained on a labeled data set from one of our two labeled datasets (e.g. optoelectronics) and then tested using the labeled data set from the other (academic life sciences). We also evaluate the performance of our supervised learning approach when trained on a labeled dataset with fewer useful disambiguation features (e.g. our academic life sciences labeled dataset where information was not recorded on records with similar inventor names that were non-matches, where inventors have lower likelihood of having missing middle name fields compared to the full USPTO, and where inventors have a relatively low likelihood of moving institutions compared to the full USPTO.) Finally, we evaluate the performance of our supervised learning approach when trained on mixes of the OE and ALS datasets and then applied to record samples non-overlapping with the training data of each.

4. Results

In the results that follow, we assess the accuracy of two examples of unsupervised, rule- and threshold-based approaches applied to the USPTO (Fleming et al., 2007; Lai et al., 2009), the semi-supervised learning algorithm trained on statistically generated artificial labeled data (Lai et al., 2014), and five supervised learning approaches for inventor disambiguation. In each case we evaluate the algorithms on our dataset of 98,762 labeled OE inventor records as well as on Azoulay's dataset of 53,378 labeled ALS inventor records.

4.1. *Evaluation of unsupervised, rule- and threshold-based algorithms for disambiguation*

Using our splitting and lumping error metrics, we evaluate the disambiguation results of Fleming et al. (2007), Lai et al. (2009) as two examples of existing unsupervised algorithms (here, rule- and threshold-based approaches) for USPTO inventor disambiguation. We re-implement these algorithms and evaluate their results using a combination of R and Python software. Our results are given in Table 5. These results reflect the disambiguation accuracy of our implementations of these algorithms on our labeled OE and ALS inventor records.[17] Note that for unsupervised and semi-supervised algorithms, "NA" appears in the "Training Dataset" column because these approaches do not use a set of labeled inventor records as training data in their respective disambiguation approaches.

Whether evaluated on the set of labeled OE inventor records or the set of labeled ALS inventor records, the Fleming et al. (2007) algorithm has a much higher splitting metric in comparison to its lumping metric, indicating that it is more susceptible to false negative errors than false positive errors for these datasets. In these results, some OE and ALS inventors are not getting credit for all of their patents, as they are being "split" into multiple inventor IDs. Consequently, lists of the most prolific OE or ALS inventors compiled using the algorithm's results will be incomplete and inaccurate. The algorithm also will overestimate the number of unique OE or ALS inventors. Finally, inventor mobility in the OE and ALS contexts will be underestimated in the Fleming et al. (2007) results. In particular, the false negative errors in the disambiguation results occur systematically due to one of the algorithm's decision rules, which requires inventors with matching common names to also share the same assignee or location. This requirement can split a mobile inventor with a common name into multiple inventor IDs. Importantly, the majority of past disambiguation approaches to the USPTO have been unsupervised approaches that, like Fleming et al. (2007), use heuristic, human-defined decision rules.

The results of the Lai et al. (2009) algorithm are more nuanced than those of Fleming et al. (2007), and offer insight both into the differences in feature characteristics between the OE and ALS datasets and into the implications of the Lai et al. (2009) algorithm's rules when the algorithm is implemented on datasets of different scale and different feature characteristics.

For our implementation of the Lai et al. (2009) algorithm on the OE and ALS labeled datasets, the splitting metric of Lai et al. (2009) is lower on both OE and ALS than the Fleming et al. (2007) algorithm run on the equivalent dataset, while the lumping metric is higher. Both the splitting and lumping metrics are still comparatively high, and especially so the splitting metric, which is still more than twice that of the lumping in the case of OE and 35% more than the lumping in the case of ALS. Recall that the lumping metric (rate of false positive errors) indicates that inventors sometimes receive credit for additional patents that do not belong to them. The false positive errors in the disambiguation results occur systematically due to a decision rule in the algorithm that allows inventor records with similar names to match if any of their assignees, locations, co-inventors, or classes (or in the case of our implementation, subclasses) match. This decision rule will lump records with inventors who have similar names and happen to share another characteristic into a single unique inventor ID. This lumping rule is particularly problematic for the Lai et al. (2009) algorithm implemented on the focused OE and ALS datasets, due to the records in these datasets, by definition, sharing a common set of classes and subclasses.[18]

The Lai et al. (2009) posted results, representing the Lai et al. (2009) algorithm run on the full USPTO have dramatically better lumping results than our implementation of the Lai et al. (2009) algorithm on OE and ALS. This improvement in lumping when Lai et al. (2009) is run on the full USPTO is likely due to the algorithm's blocking rules. In general, we would expect more records to increase lumping, keeping all other parts of the algorithm the

[17] Because the authors of these approaches did not post their disambiguation code publicly, we re-implemented these algorithms for these analyses. It is important to note that, while we followed the public descriptions of these algorithms as closely as we could, some features of the implementation may differ slightly from the authors' original disambiguation algorithms. In particular Lai et al. (2009) do not specify how their algorithm handles missing middle name fields.

[18] We also implemented the Lai et al. (2009) algorithm using the original decision rule with classes rather than subclasses. As expected, using classes instead of subclasses further increases the lumping error – specifically, to 9.46% instead of 6.79%.

Table 5
Evaluation of rule- and threshold-based unsupervised algorithms for inventor disambiguation.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Fleming et al. (2007)	Unsupervised	NA	OE_{full}	OE_{full}	13.50	0.68
Fleming et al. (2007)	Unsupervised	NA	ALS_{full}	ALS_{full}	19.82	0.00
Lai et al. (2009)	Unsupervised	NA	OE_{full}	OE_{full}	8.39	4.13
Lai et al. (2009)	Unsupervised	NA	ALS_{full}	ALS_{full}	9.16	6.79
Lai et al. (2009) posted results	Unsupervised	NA	$USPTO_{full}$	OE_{full}	9.18	0.76
Lai et al. (2009) posted results	Unsupervised	NA	$USPTO_{full}$	ALS_{full}	0.24	0.35

same. However, when Lai et al. (2009) ran their algorithm on the full USPTO database, they had to use a blocking scheme, which removes "unnecessary" comparisons to improve computational scalability. In removing these "unnecessary" comparisons, it is possible that they removed some would-be false positive errors. When implementing Lai et al. (2009) on our smaller labeled datasets, it was not computationally necessary to use blocking. Due to not using blocking, it's possible that we're introducing some false positive (lumping) errors that would not occur if we used blocking.[19] In the case of the ALS dataset, the Lai et al. (2009) posted results also have significantly less splitting than our implementation thereof. Again, it is possible that this reduction may be due to the algorithm's blocking rules removing "unnecessary" comparisons that if not removed would have been false negatives. For the OE labels, however, the splitting errors in the Lai et al. (2009) posted results remain high (indeed, slightly higher than in our implementation on just OE.

In a vacuum, the low, balanced splitting results of the posted Lai et al. (2009) algorithm evaluated on ALS might seem to reflect favorably upon the Lai et al. (2009) algorithm. Indeed, these results do reflect favorably on the algorithms accuracy at disambiguating academics in the life sciences with patents in the USPTO. A comparison of the Lai et al. (2009) algorithm's performance on the ALS dataset and on the OE dataset, however, reveals that the algorithm's performance is context-dependent. Recall that the ALS dataset is comprised of academics in the life sciences who patent. Among other characteristics, it is possible that academics are more likely to list their names correctly and without typographical errors (perhaps due to more control during the patent filing process and less mobility), making them easier to disambiguate. As shown in Table 2, inventors in this dataset are more likely to list a middle name, and to be based in the US. Thus, while the Lai et al. (2009) algorithm run on the full USPTO performs well on academics in the life sciences, it does not perform as well on the OE dataset, which has a higher percentage of missing fields and a lower percentage of U.S. inventors. This inconsistency in error rates across the OE and ALS datasets indicates that Lai et al. (2009) will not perform equally across the variety of contexts likely to be found in the USPTO.)

To assess past theoretical work using these disambiguated results or other disambiguation results based on algorithms with similar approaches, it will be necessary to look at the suitability of the research contexts to the chosen disambiguation approach's respective strengths and weaknesses. For example, Marx et al. (2009) state that they use an inventor disambiguation approach similar to the Fleming et al. (2007) algorithm. If this approach is also highly susceptible to false negative matching errors, they may underestimate inventor mobility. Without additional information, however, it is impossible to say whether the algorithm's challenges in disambiguating particular inventor types more than others influence their final results (such as that non-compete enforcement decreases mobility more sharply for inventors with firm-specific skills and for those who specialize in narrow technical fields). Singh (2005) also use an inventor disambiguation approach similar to the Fleming et al. (2007) algorithm. If their approach is similarly susceptible to false negative errors, they may underestimate the diffusion of knowledge across collaboration networks. Again, without additional information, it is impossible to say whether the algorithm's potential challenges in disambiguating particular inventor types more than others may at all influence their final results (such as that intra-regional and intra-firm knowledge flows are stronger than those across regional or firm boundaries). Fleming and Singh (2010) use an inventor disambiguation approach similar to the Lai et al. (2009) algorithm to suggest that lone inventors are more likely to produce poor outcomes and less likely to achieve breakthrough in comparison to projects outcomes achieved through collaboration. If their approach is similarly susceptible to false negative errors, they may fail to link inventors to their future patents, possibly yielding an underestimate of the effect that lone inventors have on breakthrough inventions.

Many current papers in the field do not make clear what, if any, disambiguation approach is used. Further, as discussed in the background section, hand-disambiguation based on "common sense" without labeled data will not necessarily perform better than simple disambiguation algorithms. We provide the examples above to highlight the potential implications of the above-discussed biases. In presenting these examples, we are not seeking to call-out these papers in a negative fashion, rather to highlight the importance of research in the field disclosing their disambiguation methods and discussing the implications of those methods for their specific context and results. The papers above should be lauded for being so rigorous as to present to the reader the disambiguation method upon which their results are built.

4.2. *Evaluation of a semi-supervised learning algorithm (trained on statistically generated artificial labeled data) for disambiguation*

Our evaluations of the posted disambiguation results of the Lai et al. (2014) semi-supervised inventor disambiguation algorithm using statistically generated artificial labels as training data are given in Table 6. These results reflect the disambiguation accuracy of the original authors' implementations of these algorithms on the full USPTO dataset. We compare these posted disambiguation results to our labeled OE and ALS inventor records.

We encourage the reader to compare the evaluation of the posted Lai et al. (2009) and Lai et al. (2014) results. For completeness, we also include the full evaluation results of the unsupervised inventor disambiguation algorithms from the previous section. Recall that these unsupervised algorithms were implemented on the OE and ALS subsets, while the semi-supervised Lai et al. (2014) was only run on the full USPTO database. We include our evaluation of both the Lai et al. (2009) posted results as well as of our implementation of Lai et al. (2009) algorithm on the OE and ALS dataset.

[19] It is also possible that our implementation may differ slightly from the authors' original algorithms. Specifically, Lai et al. (2009) do not specify whether their algorithm considers a missing field a match or a non-match. In our implementation of Lai et al. (2009) we assume a missing field is a non-match. If the missing field is a middle name, we redistribute the matching weight placed on the middle name to the remaining name fields.

Table 6
Evaluation of semi-supervised learning algorithms for inventor disambiguation.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Fleming et al. (2007)	Unsupervised	NA	OE_{full}	OE_{full}	13.50	0.68
Fleming et al. (2007)	Unsupervised	NA	ALS_{full}	ALS_{full}	19.82	0.00
Lai et al. (2009)	Unsupervised	NA	OE_{full}	OE_{full}	8.39	4.13
Lai et al. (2009)	Unsupervised	NA	ALS_{full}	ALS_{full}	9.16	6.79
Lai et al. (2009) posted results	Unsupervised	NA	USPTO_{full}	OE_{full}	9.18	0.76
Lai et al. (2009) posted results	Unsupervised	NA	USPTO_{full}	ALS_{full}	0.24	0.35
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO_{full}	OE_{full}	2.49	0.39
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO_{full}	ALS_{full}	0.35	0.04

Importantly, the Lai et al. (2014) algorithm can not be expected to present the same disambiguation differences with scale as the Lai et al. (2009) algorithm. Among other differences, the Lai et al. (2014) algorithm will not have the Lai et al. (2009) decision rule that allows inventor records with similar names to match if any of their assignees, locations, co-inventors, or classes match, unless this rule is embedded in the heuristics used by Lai et al. (2014) to generate their statistically generated artificial labeled data.

The Lai et al. (2014) posted results are better than those of Lai et al. (2009) based on a reduction in both error rates when evaluated on the set of labeled OE inventor records. Based on a balance of low splitting and low lumping, Lai et al. (2014) performs similarly to Lai et al. (2009) when evaluated on the set of labeled ALS inventor records, with the Lai et al. (2014) lumping results being slightly lower, but the Lai et al. (2009) results being slightly more balanced. Across the two labeled datasets we have available, the Lai et al. (2014) semi-supervised inventor disambiguation algorithm trained on statistically generated artificial labeled data provides a more robust (e.g. similar performance across our two datasets with different disambiguation features) and accurate set of disambiguation results than the rule- and threshold-based approaches.

4.3. Evaluation of out-of-sample link prediction for supervised learning models

To select a supervised learning approach to compare against the previously discussed unsupervised and semi-supervised approaches to inventor disambiguation, we first evaluate the effectiveness of five possible supervised learning models (or "classification models") for link prediction in USPTO inventor disambiguation. Rather than test each model's full disambiguation results, we focus on each model's ability to correctly predict pairwise links. Each model would use the same blocking scheme to reduce computational time and the same hierarchical clustering scheme to resolve pairwise transitivity violations, so it is not necessary to evaluate these steps. The only part of the disambiguation algorithm described in Section 3.3.3 where these models would differ is the pairwise link prediction step (step 3.b). As such, the results in the table below reflect pairwise match vs. non-match prediction accuracy only. The full disambiguation algorithm is evaluated in the next section.

Using pairwise comparisons of labeled inventor records, we build and evaluate five commonly used classification models.[20] The results shown in Table 7 are based on out-of-sample predictions of each classification method. Recall that we use this out-of-sample testing to ensure that the classification models do not overfit to the training data and will yield stable predictions on out-of-sample, unlabeled comparisons of inventor records. Discriminant analysis methods find a combination of features that best separates two or more classes of objects or events (e.g. match vs. non-match). Logistic regression was used in the Lai et al. (2014) semi-supervised learning approach and is one of the most well-known classification methods for binary responses. Classification trees and random forests are described in Section 3.3.1. Each of these methods is described in detail in Hastie et al. (2009).

Our goal is to have low, balanced splitting and lumping metrics. By limiting and balancing both types of errors, we hope to achieve results that yield more accurate lists of the most prolific inventors, will better approximate the total number of unique inventors, and will not be biased by the mobility of inventors. Note that low splitting and high lumping (or vice versa) could be preferable for some specific research questions. For example, suppose we wanted to approximate the number of unique inventors in the database, but for our particular application, it is better to underestimate than to overestimate this quantity. In this case, having low splitting and high lumping would be preferable, since this would inherently decrease the number of unique inventors. For the purposes of this paper, however, we want to balance low splitting and lumping results, with the goal of accurate disambiguation results regardless of the subsequent contextual application.

As can be seen in Table 7, logistic regression and random forests perform best, with random forests having the lowest splitting and lumping metrics of the five supervised learning methods. As shown in Appendix H, these results are robust to cross-validated standard errors on the pairwise link prediction results (Hastie et al., 2009). Random forests are known to be powerful classifiers, and are advantageous in our disambiguation context for several reasons. First, they are designed to work well with large training datasets. Second, because they are built using an ensemble of decision tree classifiers, they yield an estimate of the probability that any record-pair matches. Finally, the underlying decision tree classifiers provide an intuitive solution to the disambiguation problem. That is, the if-else structure of a decision tree is similar to the structure of many ad-hoc disambiguation approaches, except that these decision trees do not rely on human input – they learn the most accurate ways to separate matching from non-matching pairs from the training data. Because of the algorithm's performance and the advantages listed above, we choose to use it for the prediction step of our supervised learning approach to inventor disambiguation.

4.4. Evaluation of random forests algorithm for disambiguation

The evaluation metrics shown in Table 7, do not reflect the results of a full inventor disambiguation algorithm. The results only compare pairs of records. These pairwise links need to subsequently be consolidated into a set of IDs to resolve potential violations of transitivity of pairwise matches. In this section, we evaluate

[20] We also tried using support vector machines (*SVMs*; see Hastie et al. (2009) for more details) for the inventor disambiguation problem, but these yielded highly unstable results. In some cases, the SVM models could not be built on datasets of even moderate size (<100,000 pairwise comparisons) due to computational restrictions. As such, we opted to use classifiers which could take into account a larger set of pairwise comparisons so that the training data could be as representative of the disambiguation population as possible. Additionally, random forests yielded results that were as good as or better than SVMs when tested on smaller subsets of training data.

Table 7
Evaluation of out-of-sample link prediction for supervised learning models.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Linear discriminant analysis	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	8.48	1.66
Quadratic discriminant analysis	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	3.19	1.62
Classification trees	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	2.22	2.49
Logistic regression	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	1.68	1.64
Random forests	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	0.61	0.73

the results of our full supervised learning inventor disambiguation approach, which uses random forests to predict the probability that pairs of records match, hierarchical clustering to resolve pairwise transitivity violations, and blocking to reduce computational time. We evaluate the model twice – once on the OE subset and once on the ALS subset. In each case, we train the random forest model on one group of records from each dataset (OE$_{train}$ and ALS$_{train}$), then apply that model to disambiguate a different set of records from that dataset (OE$_{test}$ and ALS$_{test}$). This out-of-sample testing procedure helps reduce the probability that our models are overfit to their training data, within the context of applying the model to the same labeled dataset (i.e. OE or ALS). The results of this procedure are shown in Table 8. For comparison purposes, we also include the results of the previously discussed unsupervised and semi-supervised disambiguation approaches.

The random forest inventor disambiguation algorithm yields results similar to those of the semi-supervised approach of Lai et al. (2014) in that both maintain error rates below 3% across our two available disambiguation contexts. For the OE dataset, the random forest algorithm trained and run on OE performs approximately the same as Lai et al. (2014) trained on statistically generated artificial labels and run on the full USPTO for splitting (2.09% vs. 2.49%), but worse for lumping (1.26% vs. 0.39%). For the ALS dataset, the random forest algorithm trained on OE and run on ALS performs slightly better than Lai et al. (2014) on splitting (0.00% vs. 0.35%) and slightly worse on lumping (0.8% vs. 0.04%). It is notable that the random forest approach trained on OE has such low, balanced splitting and lumping when applied to a dataset from an entirely different industrial and institutional context (ALS). Overall, however, the semi-supervised learning approach trained on statistically generated artificial labels and run on the full USPTO slightly outperforms the supervised learning approach in this example, based on a balance of both low splitting and low lumping.

In Appendix G, we discuss sources of error and variation across disambiguation algorithms. In Appendix H, we discuss our approach for obtaining standard errors on the splitting and lumping results for our random forests disambiguation algorithm via cross-validation. The results of this procedure, shown in Table 17, suggest that the above-discussed results hold given our algorithm's standard errors. We are unable to assess the standard errors for the Lai et al. (2014) algorithm, or know where the posted results of Lai et al. (2014) sit with respect to those standard errors.

4.5. Random forests trained on labeled records with different feature distributions

A classification algorithm will perform best when trained on labeled data with useful features (in determining whether two records are or are not a match) and a feature distribution that matches the target population for disambiguation. Labeled data that meets these requirements, however, can be costly (both in terms of time and money) and difficult or even impossible to acquire. To help support future researchers using our disambiguation code in assessing the value of garnering additional outside labeled data for their context (compared to using the labeled OE data we are able to provide), we evaluate the performance of our supervised learning approach in a variety of training contexts, to the extent possible with our two labeled datasets. Our goal is to illustrate the consequences for the performance of our random forest algorithm of being trained on labeled data with different extents of useful features and different feature distributions than alone our labeled OE dataset. With respect to useful features for disambiguation, recall that one important difference between the OE and the ALS datasets is that the ALS dataset does not contain information on records with similar names that after evaluation with labeled data were found not to be matches. For differences in disambiguation features across the two datasets, refer to Table 2. The results of this procedure are shown in Table 9. To support comparison, in the last six rows of the table we re-include the results of the random forest trained on OE as well as the results of the Lai et al. (2009) and Lai et al. (2014) posted results presented in Table 8.

As can be seen in Table 9, in all cases, training the random forest algorithm on ALS data leads to reduced performance on the test datasets, compared to training on the OE data. Training on the ALS subset and evaluating on the ALS subset, the random forest algorithm yields significantly higher splitting (14.38% vs. 0.00%) and somewhat higher lumping (3.68% vs. 0.8%) than when trained on OE and run on ALS. Training on the ALS subset and evaluating on the OE dataset yields, from the perspective of low, balanced splitting and lumping, even worse results. While lumping is comparatively low (0.29% vs. 1.26%) when training on ALS and running on OE, splitting errors are extraordinarily high at 44.41% (compared to 2.09% for trained on OE and run and tested on OE.) Adding ALS training data into the OE training data also in all cases leads to lower performance than just training on OE. Training on a mix of the OE$_{train}$ and ALS$_{train}$

Table 8
Evaluation of random forests algorithm for inventor disambiguation.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random forests	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	2.09	1.26
Random forests	Supervised	OE$_{train}$	ALS$_{test}$	ALS$_{test}$	0.00	0.80
Fleming et al. (2007)	Unsupervised	NA	OE$_{full}$	OE$_{full}$	13.50	0.68
Fleming et al. (2007)	Unsupervised	NA	ALS$_{full}$	ALS$_{full}$	19.82	0.00
Lai et al. (2009)	Unsupervised	NA	OE$_{full}$	OE$_{full}$	8.39	4.13
Lai et al. (2009)	Unsupervised	NA	ALS$_{full}$	ALS$_{full}$	9.16	6.79
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{full}$	9.18	0.76
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	ALS$_{full}$	0.24	0.35
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{full}$	2.49	0.39
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	ALS$_{full}$	0.35	0.04

Table 9
Disambiguating a target population with random forests trained on different labeled records.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random forests	Supervised	ALS$_{train}$	ALS$_{test}$	ALS$_{test}$	14.36	3.68
Random forests	Supervised	ALS$_{train}$	OE$_{test}$	OE$_{test}$	44.41	0.29
Random forests	Supervised	OE$_{train}$ + ALS$_{train}$	OE$_{test}$	OE$_{test}$	11.19	0.72
Random forests	Supervised	OE$_{train}$ + ALS$_{train}$	ALS$_{test}$	ALS$_{test}$	0.12	3.75
Random forests	Supervised	OE$_{train}$ + ALS$_{train}$	OE$_{test}$ + ALS$_{test}$	OE$_{test}$ + ALS$_{test}$	7.70	1.34
Random forests	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	2.09	1.26
Random forests	Supervised	OE$_{train}$	ALS$_{test}$	ALS$_{test}$	0.00	0.80
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{full}$	9.18	0.76
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	ALS$_{full}$	0.24	0.35
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{full}$	2.49	0.39
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	ALS$_{full}$	0.35	0.04

subsets and evaluating on the OE$_{test}$ subset, reduces the splitting errors compared to training on ALS alone, but still significantly worse splitting than training just on OE. Likewise, and perhaps most surprisingly, training on a mix of the OE$_{train}$ and ALS$_{train}$ subsets and evaluating on the ALS$_{test}$ subset, reduces the splitting errors compared to training on ALS alone, but still significantly worse splitting than training just on OE and running on ALS. This reduced performance of the random forest algorithm when training on ALS is most likely driven by the ALS dataset's lack of information on records with similar inventor names where the records proved after being checked with labeled data not to be a match.

Additional factors reducing the usefulness of the ALS labeled dataset for training may include the slightly smaller size of the ALS dataset relative to the OE dataset, the relatively low percentage of missing fields in the ALS data, the reduced frequency of differences in how the same inventor reports their name, and the near-exclusive focus on U.S. inventors leading the random forest algorithm to lack sufficient training to perform well in alternative contexts (e.g. contexts that are not the norm in the ALS dataset) when trained on ALS. Additionally, as pointed out by Raffo and Lhuillery (2009), differences in parsing methods for the ALS and OE datasets could also lead to differences in matching, specifically if the features used in the random forest models have differing distributional characteristics across these two datasets.

To shed additional insight into the results in Table 9, we show in Table 10 scaled Gini importance statistics for the random forest model trained on each of the above datasets. The Gini importance statistics show how the labeled data on which a classifier is trained influences the relative importance of each disambiguation feature in determining whether or not two records are a match (Breiman, 2001). In Table 10, we scale the Gini importance statistics[21] to support comparing the Gini importance statistics across the three training datasets found in this paper: OE$_{train}$, ALS$_{train}$, and OE$_{train}$ + ALS$_{train}$. Note that the values of the Gini importance statistics by themselves do not hold meaning, rather they provide the relative importance of individual disambiguation features in determining a match versus non-match within a given model.

Table 10
Scaled Gini importance statistics for random forests trained on different samples of labeled inventor records. "feature$_j$" indicates a Jaro–Winkler comparison. "feature$_e$" indicates an exact matching comparison for that feature. "feature$_3$" indicates a comparison of the first three characters.

Feature name	OE$_{train}$	ALS$_{train}$	OE$_{train}$ + ALS$_{train}$
last_j	237.25	515.62	505.66
last_e	182.13	228.17	285.90
last_3	47.91	201.25	212.09
first_j	2574.16	563.16	550.52
first_e	2537.82	182.76	283.18
first_3	1379.38	382.86	357.12
mid_j	354.62	102.45	296.74
mid_e	257.23	100.19	337.66
city_j	804.40	658.19	873.33
city_e	887.55	514.86	629.78
state_e	359.56	1014.40	1024.90
country_e	1.41	4.42	130.83
suffix_e	0.62	12.69	2.09
assignee_j	226.90	3019.73	2657.69
assignee_e	149.05	2499.26	1852.52

The random forest algorithm trained on OE$_{train}$ places more emphasis on comparisons of first name and of middle name than the other two models. In contrast, assignee comparisons are not a significant factor in determining matches. For the random forest model trained on OE, last name comparisons remain important in determining matches, but not as influential as first name comparisons. This relative importance of the disambiguation features in determining matches is a direct result of the construction (e.g. nature and distribution of features) of our OE dataset: As discussed in Section 3.1.1, for each OE inventor for whom we had a CV, we generated a list of potential matches that included all inventor records in the USPTO with a last name similarity score of at least 0.9 with the last name on the inventor's CV. We then used the inventor's CV information and follow-up calls with the inventor to label each record as a match or non-match. Our resulting OE training dataset contained 14,520 records that matched one of the 824 CVs and 84,242 records that did not match to one of the 824 CVs. As a result, the training data supplied to the random forest model contains extensive information on non-matching pairs with similar last names, decreasing the importance of the last name comparisons in the model.

In contrast, assignee comparisons are by far the most significant factor in determining matches for the random forest trained on ALS$_{train}$. This influence of assignees in determining matches is likely the explanation for why these models do not perform well when tested on alternative datasets. Again, the significance of assignees is again a function of the feature distribution of the ALS dataset. For a significant fraction of the hand-labeled inventors in the ALS dataset, the matching pairs of inventor records always shared the same assignee. This feature may have been a consequence of the low probability of changing institutions of academics in the life sciences, as noted in Azoulay et al. (2012). Including the ALS data in a mixture of OE and ALS training data similarly leads the random forest model to place significant weight on matching pairs of records with the same assignee, leading to errors when matching inventors who change institutions.

In conclusion, the results of this section emphasize the importance of having training data that not only matches the feature

[21] We scale each model's Gini importance statistics by the total Gini importance of that model, so that Gini importance across models are comparable. That is, we divide each Gini importance by the sum of all Gini importance from its model, then multiply by a constant scaling factor for interpretability.

distribution of your target dataset for disambiguation, but also has useful features – such as on non-matches – from which the model can learn the best features from which to determine when inventor records should and should not match.

4.6. Evaluation of all algorithms on labeled inventor records from OE sub-samples

Finally, and perhaps most importantly, we leverage differences in feature distributions and disambiguation characteristics (likely attributable to how prolific each sample of inventors is) across the four sub-samples of our labeled OE inventor records dataset (OE_{Most}, OE_{Rate}, OE_{Int}, and OE_{Rand}) to evaluate all four disambiguation algorithms discussed in this paper. From the perspective of developing a generalizable approach robust to alternative disambiguation contexts likely to be found in the full USPTO, the ideal disambiguation approach would offer low, balanced error rates across each of these subsets, and perform consistently across all four subsets. Specifically, an algorithm that yields low, balanced error rates across all four samples would be preferable to an algorithm that yield the lowest, balanced error rates for the prolific inventor samples but high, unbalanced error rates for other samples. When evaluating the performance of an algorithm, the consistency of a disambiguation algorithm's performance is as important a consideration as its accuracy: if an algorithm does not have consistent performance across inventor records from different institutional, industrial, or individual contexts, the disambiguation results may introduce context-specific biases (corresponding to different features and characteristics of the underlying records) into the analyses of researchers using these results. For the random forests algorithm, we train again on OE_{train}, which is a stratified random sample of inventors from each of the four subgroups. OE_{train} does not share any records with any of the four test sub-samples. The results of this analysis are shown in Table 11.

When compared with Table 8, the results in Table 11 reveal that the unsupervised and semi-supervised approaches suffer from significant false negative errors when applied to our random sample of optoelectronic inventors. When evaluated on $OE_{Most\text{-}test}$, $OE_{Rate\text{-}test}$, and $OE_{Int\text{-}test}$, each disambiguation algorithm evaluated here performs about the same as it does on the full set of labeled OE inventor records. The two unsupervised approaches run on each of the labeled inventor subsets have slightly better performance on the $OE_{Most\text{-}test}$, $OE_{Rate\text{-}test}$, and $OE_{Int\text{-}test}$ samples than when the same algorithm is run on the full OE dataset. Breaking out the posted results of the Lai et al. (2009) algorithm run on the full USPTO by OE subset reveals that the algorithm has slightly better performance on the $OE_{Most\text{-}test}$ and $OE_{Int\text{-}test}$ subsets and approximately the same or slightly worse performance on the $OE_{Rate\text{-}test}$ subset compared to the performance shown earlier for the full OE dataset. Breaking out the posted results of the Lai et al. (2014) algorithm run on the full USPTO by OE subset reveals that the algorithm has slightly better performance on the $OE_{Rate\text{-}test}$ subset and approximately the same or slightly worse performance on the $OE_{Most\text{-}test}$ and $OE_{Int\text{-}test}$ subsets compared to the performance shown earlier for the full OE dataset.

Overall, the Lai et al. (2014) and random forests algorithms yield the most consistently low, balanced error rates on prolific inventors, with the Lai et al. (2014) having slightly better lumping rates throughout. Since multiple important papers in TIE focus on prolific inventors (or "stars") these consistently low, balanced error rates for Lai et al. (2014) and random forests are important; results from papers that focus on star inventors (Zucker and Darby, 1996; Zucker et al., 2011; Azoulay et al., 2010, 2012) will likely be minimally affected by our findings if they use these or similar disambiguation approaches. We discuss the performance of all of the approaches in this paper for prolific inventors in further detail in Appendix F. However, when evaluated on the random sample of OE inventors (OE_{Rand}), the unsupervised (Fleming et al., 2007; Lai et al., 2009) and semi-supervised approaches (Lai et al., 2014) perform quite poorly – with splitting rates from 10 to over 20%. These high splitting metrics indicate a high prevalence of false negative errors in the disambiguation results. Even Lai et al. (2014), which outperforms the random forests approach in earlier analyses, suffers this high rate of false negative errors, with a splitting metric of 10.54%. These results for Lai et al. (2014) align with those of (Ge et al., 2014), who find that Lai et al. (2014) over-estimate mobility on a sample of 13,181 individuals with online profiles. Note, that while Ge et al. (2014) attribute the error in tracking mobility of engineers and scientists to using patents, our results suggest that a significant proportion of this error may be coming from the disambiguation. Specifically, disambiguation algorithms with high splitting rates identify patents belonging to one inventor in two locations as belonging to two different inventors.

Similar to the discussion in Section 4.1, this high prevalence of false negative errors in the disambiguation results could have implications for past work done using algorithms similar to Fleming et al. (2007), Lai et al. (2009), and Lai et al. (2014). Indeed, Ge et al. (2014) find that out of six past findings on mobility and productivity based on inventors' patent records where the matching of inventors to patents was achieved through varying patent matching and disambiguation methods, only one holds when mobility is instead estimated based on the information in their linked in sample. In the case of our study, the high false negative error rates could also impact metrics involving lone inventors as sources of breakthrough inventions, e.g. Fleming and Singh (2010), inventor mobility, e.g. Marx et al. (2009), and knowledge diffusion across inventor collaboration networks, e.g. Singh (2005).

Random forests, in contrast, yields consistently low, balanced error rates across all four sub-samples of the labeled OE database. On the OE_{Rand} group, random forests has a lower splitting metric (1.74%) than it does for any of the other three subgroups, while maintaining a lumping metric of 2.48%. If the goal of a paper was disambiguating our OE dataset, our random forest model trained on the OE dataset is by far the best approach of those evaluated. Extrapolation from the results presented in this paper the value of our random forest model in a broader array of contexts is more challenging. Our supervised learning approach – using blocking, random forests, and hierarchical clustering – provides the most consistent low, balanced, error rates across the labeled data samples available to us in this study, each of which has unique feature distributions and disambiguation characteristics. The performance of our random forest model trained on OE and tested on ALS and vice versa (where error rates remain below 4%) and the performance of our random forest model on our random sample of OE inventors suggest that the our random forests algorithm may be less likely to introduce systematic, context-specific biases into subsequent research compared to the results of the unsupervised and semi-supervised approaches discussed above. We are, however, limited by the labeled datasets available to us in this study.

Future users will inevitably face the question of when to use the labeled data we provide and when the value (in terms of more accurate disambiguation) of obtaining additional labeled data will outweigh the time and costs. Indeed, our process (in conjunction with Akinsanmi et al. (2014) of collecting as carefully constructed a labeled dataset as presented here, including collecting inventor contact information from the top three professional societies in optoelectronics, tracking down the inventors making up the random OE sample, and re-checking patent lists with individual inventors took three years and efforts by more than 14 individuals (including 10 diligent and persuasive CMU undergraduates) to complete. In situations where other labeled data does not exist, our choice to build our supervised learning model on a minimum set of

Table 11
Evaluation of all algorithms on labeled inventor records from OE sub-samples.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random forests	Supervised	OE$_{train}$	OE$_{Most\text{-}test}$	OE$_{Most\text{-}test}$	2.26	1.54
Random forests	Supervised	OE$_{train}$	OE$_{Rate\text{-}test}$	OE$_{Rate\text{-}test}$	2.01	1.55
Random forests	Supervised	OE$_{train}$	OE$_{Int\text{-}test}$	OE$_{Int\text{-}test}$	2.61	0.67
Random forests	Supervised	OE$_{train}$	OE$_{Rand\text{-}test}$	OE$_{Rand\text{-}test}$	1.74	2.48
Fleming et al. (2007)	Unsupervised	NA	OE$_{Most\text{-}test}$	OE$_{Most\text{-}test}$	8.40	0.16
Fleming et al. (2007)	Unsupervised	NA	OE$_{Rate\text{-}test}$	OE$_{Rate\text{-}test}$	7.49	0.47
Fleming et al. (2007)	Unsupervised	NA	OE$_{Int\text{-}test}$	OE$_{Int\text{-}test}$	7.68	0.03
Fleming et al. (2007)	Unsupervised	NA	OE$_{Rand\text{-}test}$	OE$_{Rand\text{-}test}$	22.28	0.35
Lai et al. (2009)	Unsupervised	NA	OE$_{Most\text{-}test}$	OE$_{Most\text{-}test}$	8.17	1.10
Lai et al. (2009)	Unsupervised	NA	OE$_{Rate\text{-}test}$	OE$_{Rate\text{-}test}$	7.60	4.61
Lai et al. (2009)	Unsupervised	NA	OE$_{Int\text{-}test}$	OE$_{Int\text{-}test}$	7.71	0.36
Lai et al. (2009)	Unsupervised	NA	OE$_{Rand\text{-}test}$	OE$_{Rand\text{-}test}$	19.94	0.35
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{Most\text{-}test}$	8.34	0.37
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{Rate\text{-}test}$	9.19	0.85
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{Int\text{-}test}$	5.49	0.29
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{Rand\text{-}test}$	15.02	0.76
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{Most\text{-}test}$	2.50	0.33
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{Rate\text{-}test}$	1.64	0.23
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{Int\text{-}test}$	2.60	0.27
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{Rand\text{-}test}$	10.54	1.21

features combined with the results in Tables 8 and 11 suggest that our model trained on OE labeled data may perform with error rates below 3% in other contexts including those with non-star inventors. More labeled data would be necessary to explore this conjecture across a wider variety of contexts. In the case of industry- or other context-specific studies, given that TIE researchers often for this type of work make efforts to collect additional data on individuals, they may choose to leverage the labeled data from their context either in conjunction with or in place of the labeled data be provide from OE for training their disambiguation algorithms. Importantly, as shown in Table 9, when choosing to develop and leverage additional labeled data, it will be critical for researchers to include information in their labeled dataset on both when similar names are matches as well as when similar names are not matches. In the case of the full USPTO, ideally, a repository would be developed where researchers, as they completed their individual industry- or context-specific studies, would describe (e.g. the collection mechanisms and features of) and share their labeled data. This larger pool of labeled data could then be used by the TIE field and beyond to support better disambiguation.

5. Discussion

Modern methods for record linkage and disambiguation generally fall into one of three categories: unsupervised (including, for the purposes of this paper, "rule- and threshold-based") approaches, semi-supervised learning approaches (including, for the purposes of this paper those trained on statistically generated artificial labels), and supervised learning approaches. To date, the majority of disambiguation approaches in the field of TIE have been rule- and threshold-based, using heuristic, human-defined decision rules to determine which records should be linked. Examples include Trajtenberg et al. (2006), Fleming et al. (2007), and Lai et al. (2009), all of which disambiguate inventors in the United States Patent and Trademark Office database. More recently, the (Lai et al., 2014) apply a semi-supervised learning approach for USPTO inventor disambiguation, using statistically generated artificial training data (pairs of records that are highly likely to be either matches or non-matches) to build statistical models that determine which records should be linked. We introduce the first fully supervised learning approach for USPTO inventor disambiguation. Our approach leverages extensive sets of "labeled" (hand-matched based on information from resumes) inventor records to build "random forest" models (Breiman, 2001) to predict which pairs of records should be linked.

We evaluate two rule- and threshold-based unsupervised approaches for USPTO inventor disambiguation (Fleming et al., 2007; Lai et al., 2009), the only semi-supervised learning approach for USPTO inventor disambiguation (Lai et al., 2014), and our supervised learning approach (using random forests) against two extensive sets of labeled USPTO inventor records. The first corresponds to a set of 824 inventors with patents in the optoelectronic industry, which can be split into four sub-samples (one of which is a random sample of optoelectronics inventors) with varying characteristics (Akinsanmi et al., 2014). The second is a sample of academic inventors with patents in the life sciences (Azoulay et al., 2012). Our evaluation criteria is the extent to which these algorithms each to consistently achieve a balance of both low splitting errors and low lumping errors across the range of labeled sub-samples with different disambiguation features available to us. Here, consistent performance across contexts is equally important to balance, as a disambiguation algorithm that performs inconsistently across contexts would provide results that suggest differences across, for example, institutional or industrial contexts (or particular types of inventors) that are created by the disambiguation algorithm rather than being a reality in the original data.

We find that the random forests classification approach yields the most consistently low, balanced error rates across all samples, each of which has different feature distributions and disambiguation characteristics.

We find that the performance of the three past disambiguation algorithms that we evaluate have high splitting rates, and have performance that varies with the features of the dataset to be disambiguated. Fleming et al. (2007), which is similar to the majority of past disambiguation approaches to the USPTO including, for example, (Singh, 2005; Jones, 2005), has a relatively high splitting rate, regardless of dataset. Lai et al. (2009), which takes a more advanced approach and is similar to past approaches such as (Trajtenberg et al., 2006; Lissoni et al., 2006; Miguelez and Gomez-Miguelez, 2011), performs relatively well at disambiguating the set of academics in the life sciences with patents. In the academic life sciences dataset, inventors appear to submit relatively consistent information to the USPTO (something we hypothesize may be more likely for academics and non-mobile inventors), include their middle initial, and are primarily U.S. based. However, Lai et al. (2009) fairs less well on the optoelectronics dataset, where middle names and other fields are frequently missing, and the proportion of U.S. inventors is (as in the full USPTO) only approximately half of all inventors in the sample. The semi-supervised Lai et al. (2014) algorithm, in turn, at first appears to outperform all other inventor disambiguation algorithms (including our supervised learning approach) on both the full academic life sciences and the full optoelectronics datasets.

When we examine the performance of each algorithm on different subsets of inventors in optoelectronics; however, we find that the semi-supervised approach's performance is inconsistent across contexts and has significant splitting errors. Specifically, we find significant errors for the unsupervised and semi-supervised approaches on the random sample of optoelectronic inventors. The semi-supervised approach continues to perform well on highly prolific inventors, yielding substantially better error rates than the unsupervised approaches and slightly better error rates than our random forests approach. However, the unsupervised and the semi-supervised approaches yield false negative error rates ranging from 10% (for the semi-supervised approach) to more than 20% on the random sample of optoelectronic inventors. While biases in disambiguating any particular group would be concerning, the random sample of optoelectronic inventors is particularly significant in that it is the closest of our sub-samples to what might be expected of the majority of inventors in the USPTO (based on disambiguation-relevant comparison metrics). Our supervised learning approach, using random forests trained on OE, consistently maintains error rates below 3% across the four OE samples as well as the ALS data.

Our results suggest it important for the field to continue to pursue disambiguation approaches that are consistent across disambiguation contexts with varying features. Indeed, our analysis suggests that a substantial proportion of the error identified by Ge et al. (2014) in the tracking of mobility of engineers and scientists using patents may be attributable to the matching procedure or disambiguation algorithm itself. The performance of our algorithm on additional USPTO datasets (e.g. other contexts as well as the full USPTO database) is inevitably limited by the features of the labeled USPTO inventor records to which we had access. Incorporating into our training data labeled records with useful features from alternative samples will likely improve our random forests algorithm's ability to disambiguate additional USPTO datasets, since this will allow samples of records with different features to be represented and accounted for in our models. Importantly, it will be critical for this labeled training data to include information not only on matches, but also on the records with similar inventor names that are not matches. To continue to improve inventor disambiguation in the USPTO, as well as our interpretation of research leveraging the results of past disambiguation algorithms, it will also be important to continue to evaluate existing approaches on other sets of labeled inventor records, both to identify other areas of potential bias and to further improve learning models used for USPTO inventor disambiguation. It will similarly be essential to evaluate our algorithm when applied to new contexts or trained on new datasets as well as new learning algorithms on labeled data before using them to develop theory. Going forward, it is imperative that the field moves towards requiring authors to state as part of their theoretical papers the disambiguation approach used to generate the data upon which the theory is built, including a discussion of where that disambiguation approach may have biases.

To support researchers seeking to apply the supervised learning approach we describe in this paper, we make public (http://www.cmu.edu/epp/disambiguation) all code and labeled optoelectronics inventor records associated with our approach. In our code, we allow users to specify their desired susceptibility to false positive and false negative disambiguation errors in the results. We also release the set of labeled optoelectronics inventor records used in our analyses so that other researchers in the field can use them for evaluating their own disambiguation approaches and/or building their own supervised learning models. Additionally, we allow users to use their own training data, so that those interested in building on our work or in disambiguating specific sub-samples can do so more accurately (provided they have or are able to collect a representative training dataset for their disambiguation context with useful features). Finally, our disambiguation code works for disambiguation problems both within and outside of the USPTO context. That is, any researcher interested in disambiguating any database with any features can use our code for this purpose, provided that they have their own set of labeled training data.

Acknowledgements

We would like to thank first and foremost, Eyiwunmi Akinsanmi, for spearheading the collection of the 824 USPTO optoelectronics inventor resumes. Many thanks to SPIE in particular Eugene Arthurs and Krisinda Plenkovich and the Optical Society of America (the two largest professional societies in optics) for sharing with the Fuchs research group contact information for reaching the optoelectronics inventors. Thanks to Neha Nandakumar, for her fantastic help parsing the resume data and matching it to the patent data; Alex Lucci, Angela Ng, Carl Glazer, Willis Chang, Derek Lessard, Farjad Zaim, and Dan Murby for their extraordinary efforts tracking down individual inventors and their resumes; and Peter Pong for his preliminary work scraping and disambiguating the USPTO optoelectronics data.

Special thanks to Pierre Azoulay, who kindly shared his set of labeled USPTO inventor records from the academics in the life sciences, without which we could not show the performance of our algorithm in industrial and institutional contexts beyond optoelectronics. Many thanks to Lee Fleming for his early encouragement, and he and his groups openness sharing insights from their own efforts. Thanks also to Matthew Marx for his very helpful feedback as our discussant at the 2012 Roundtable on Engineering Entrepreneurship Research (REER). Thanks likewise to Vetle Torvik and the participants in the Graduate School of Library and Information Science University of Illinois at Urbana-Champaign Workshop on Disambiguation; Marie Thursby and the participants at the 2012 REER workshop; Carliss Baldwin, Rahul Kapoor, and the other attendees of our session at the 2012 Industry Studies Association Annual Conference; Stephen E. Fienberg, William F. Eddy, Rebecca C. Steorts, Mauricio Sadinle, and other members of the Carnegie Mellon University Department of Statistics Census research group; attendees of the 2012 Classification Society Annual Meeting; Kenneth Huang and the other participants in the 2012 Academy of Management Professional Development Workshop "Measuring Knowledge Flows: Patent and Non-Patent Data; and Scott Stern, Benjamin Jones, and Heidi Williams and the other participants at the NBER Summer Institute 2014 Innovation Workshop for their early feedback on this work. Finally, we are grateful to the National Science Foundation for funding this research through grants #1056955 (CAREER: Rethinking National Innovation Systems Economic Downturns, Offshoring, and the Global Evolution of Technology), #0830354, #25951-1-1121631, and #SES-1130706 (NCRNMN: Data Integration, Online Data Collection and Privacy Protection for Census 2020).

Appendix A. Optoelectronics classes and subclasses

The following classes and subclasses have been designated as belonging to the optoelectronics industry (format: Class ID/Subclass ID; * denotes all subclasses within this class; – denotes a range of subclasses within this class):

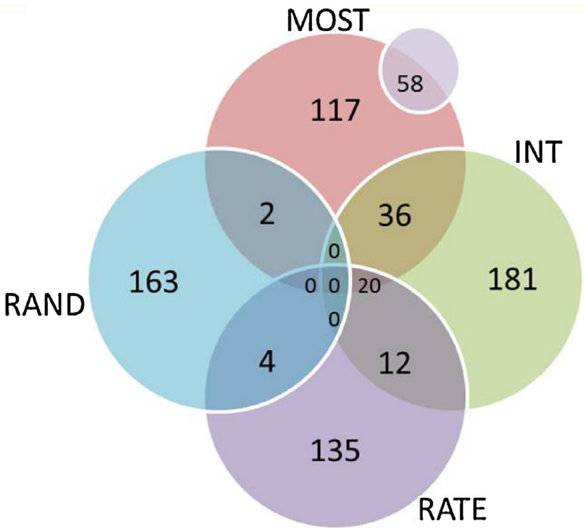


Fig. 3. Overlap between curricula vitae groups/samples.

720/∗, 356/∗, 372/∗, 385/∗, 359/∗, 398/∗, 250/200–339,
250/551, 438/24, 438/25, 438/27, 353/∗, 257/13, 257/21,
257/53–56, 257/59, 257/79–103, 257/113–118,
257/184–189, 257/225–234, 257/257–258, 257/290–294,
257/431–466, G9B/7

Definitions for these classes and subclasses can be found at: http://www.uspto.gov/web/patents/classification/.

Appendix B. Overlap between groups of curricula vitae samples

Akinsanmi et al. (2014) provide the following graphic (Fig. 3), which details the overlaps between the different CV groups described in Section 3.1.1.

The most relevant group to our results and findings in this paper is the "Rand" group, described in Section 3.1.1. Note that this group only has six of 169 unique inventors overlapping with the other groups (four with the "Rate" group, and two with the "Most" group). This indicates that the "Rand" group is not biased towards prolific inventors, which we expect since it is a random sample of OE inventors.

For more information, please refer to Akinsanmi et al. (2014).

Appendix C. Comparing two inventor records

In disambiguation, we compare pairs of inventor records and determine if each pair is a match (the same unique inventor) or a non-match (two non-unique inventors). Several authors analyze the best approaches to comparing different types of fields (names, companies, locations, etc) for record linkage and disambiguation purposes. We direct interested readers to the works of Elmagarmid et al. (2007), Bilenko and Mooney (2003b), Bilenko et al. (2003), and Christen (2006) for further details on these discussions.

In order to make the match vs. non-match decision, we need to know how similar the pair of inventor records is. To do this, we describe the similarity of each field with a numerical value indicating how closely two records match. For the purposes of this paper, we define all of these "similarity scores" as γ_{ijk}, which represents the similarity score of records x_i and x_j according to field k, where $i, j \in \{1, 2, \ldots, n\}$, $k \in \{1, 2, \ldots, K\}$, K = the number of unique fields being compared, and n = the number of records in the database

There are three different types of fields that are compared: Long strings, short strings, and lists. In the next sections, we define similarity scores for each field in our dataset and discuss the computational issues for large numbers of comparisons. Our choices of similarity scores are motivated by previous authors' work in name disambiguation (Winkler, 1990; Bilenko and Mooney, 2003b; Lai et al., 2009), and it should be noted that changing these similarity scores could affect the disambiguation results. We also tried including marginal effects for other features, such as commonness of first or last name, in our models. However, these marginal effects were insignificant in the resulting models, so we do not show them here.

C.1. Long text strings: inventor, city, and assignee names

Long strings, such as assignee and inventor names, are susceptible to typographical errors and name variations. For example, none of "Sony Corporation," "Sony Corporatoin," and "Sony Corp." will match using simple exact matching. Similarly, "David" vs. "Dave" would not match. It is clear that more advanced string comparison methods are necessary for long strings.

The Jaro–Winkler string comparison (JW) method takes two strings as input and compares the characters and positions of matching characters across two strings (Winkler, 1990). The Jaro–Winkler string comparator provides a simple solution for quantifying the similarity of a pair of text strings, and its results have intuitive properties. The result is a score between 0 and 1 (inclusive) that indicates how similar two strings are to each other; if two strings are an exact match, their JW score will be 1. The mathematical details of the calculation of JW scores are given in Winkler (1990). Briefly, given two long strings X_{ik} and X_{jk} for inventors i and j and field k, $\gamma_{ijk} = 1$ if $X_{ik} = X_{jk}$, and $\gamma_{ijk} = 0$ if none of the characters in X_{ik} are also in X_{jk}. For our dataset, the long string fields are first name, last name, middle name, assignee name, and inventor city.

This approach is also used by other researchers, including Lai et al. (2009) and Lai et al. (2014). However, note that our supervised learning approach is flexible and does not depend on a specific type of string comparison. In the models used in this paper, we also use SoundEx to compare name attributes. SoundEx is described in further detail in Bilenko and Mooney (2003b), and we direct interested readers here. We note that Jaro–Winkler similarity scores are more important in our random forests models than SoundEx similarity scores, according to Gini importance index for each of our models.

Finally, other string comparison metrics such as the Levenshtein distance were tried (Levenshtein, 1966), but did not improve our models or results. Several other string comparison metrics exist, such as Token-based similarities, Metaphone, N-grams, and still several others. We choose to use a small set of string similarity metrics so as to avoid overfitting our models to the training data, and to mimic the comparisons done by earlier disambiguation approaches (e.g. Lai et al. (2009)). We direct interested readers to the works of Elmagarmid et al. (2007), Bilenko and Mooney (2003b), Bilenko et al. (2003), and Christen (2006) for more details on string similarity metrics.

C.2. Short text strings: state, country, and name suffix

If field k is a short string, we define the similarity score as follows. Given two short strings X_{ik} and X_{jk} for inventors i and j and field k, $\gamma_{ijk} = 1$ if $X_{ik} = X_{jk}$, and $\gamma_{ijk} = 0$ if $X_{ik} \neq X_{jk}$. That is, we check pairs of short strings for exact matches only.

Short string fields include the inventor name suffix, inventor state, and inventor country. We use exact matching for these fields because they are generally not susceptible to typos, and we do not want to give non-identical strings with similar characters a non-zero weight, such as the state abbreviations "MA" and "MN". Finally, note that many existing disambiguation approaches, such as the Lai et al. (2009) algorithm, also use exact matching for short strings.

C.3. *Lists: co-inventors, classes, and subclasses*

Each inventor record has two lists associated with it: (1) the list of co-inventors and (2) the list of class-subclass pairs for the corresponding patent. There are several different ways to quantify the similarity of two lists of co-inventors or class-subclass pairs. For the purposes of this paper, we use the following approach when comparing lists of co-inventors or class-subclass pairs. Given two lists X_{ik} and X_{jk} for inventors i and j and field k:

$$\gamma_{ijk} = \frac{|X_{ik} \cap X_{jk}|}{|X_{ik} \cup X_{jk}|}$$

That is, list similarity scores find the ratio of shared elements to unique elements across the two lists. Again, note that other list similarity scores could be substituted here.

Appendix D. Hierarchical clustering

The following discussion of the relationship between enforcing transitivity in the context of deduplication and applying single-linkage hierarchical clustering to identify clusters of records that refer to the same unique entity is taken from Ventura et al. (2014).

Mathematically, enforcing transitivity of pairwise matches in deduplication is equivalent to using single linkage hierarchical clustering with distance matrix D to identify unique entities, where $D[i, j] = \hat{d}_{ij} = h(\hat{p}_{ij})$, $\hat{p}_{ij}$ is the estimated probability of x_i, x_j matching, and h is some monotonically decreasing function. When transitivity is enforced, pairwise match/non-match decisions are made by comparing $\hat{p}_{ij}$ to some threshold. Let $0 < a \leq 1$ be the threshold, let $\tau = h(a)$, and (without loss of generality) let $h(x) = 1 - x$. Then, all pairs of records with $\hat{p}_{ij} \geq a$ are considered pairwise matches. Consequently, all pairs of records with $\hat{d}_{ij} = h(\hat{p}_{ij}) = 1 - \hat{p}_{ij} \leq 1 - a = \tau$ are considered pairwise matches. Thus, cutting the single linkage hierarchical clustering tree at the level τ is equivalent to linking all pairs of records with $\hat{p}_{ij} \geq a = h^{-1}(\tau)$ and enforcing transitivity amongst those pairwise links. Hartigan (1981) shows that the single linkage hierarchical clustering solution is consistent for high-density clusters, a useful property for deduplication applications with identical records (or "exact matches").

Appendix E. Blocking, computational complexity, and parallelization

Computational issues can make it challenging to apply our supervised learning approach to large-scale disambiguation problems. If a database has n records, then estimating all $\binom{n}{2}$ probabilities/distances is an $O(n^2)$ operation. As such, we use a technique called "blocking" to reduce computational complexity. A blocking rule partitions the data into homogeneous groups of records that share some basic feature or set of features. Blocking is done for two reasons: reducing the number of false positive errors and preventing the algorithm from making unnecessary/computationally taxing comparisons. In particular, we partition the n records into B "blocks," so that similar records are placed into the same block. Then, we only calculate estimated probabilities/distances for pairs of records within blocks; records that are not in the same block are not compared, substantially reducing the comparison space. Blocking is a commonly used tool for record linkage, with approaches dating back as far as the original Fellegi–Sunter model (Fellegi and Sunter, 1969) and continuing with more modern approaches (Kelley, 1984).

Blocking results in a set of B blocks, each of size n_b, with $b = 1, \ldots, B$ with $\sum_{b=1}^{B} n_b = n$. As a result, blocking reduces the total number of comparisons (or pairwise probability/distance estimates) from $\binom{n}{2}$ to $\sum_{b=1}^{B} \binom{n_b}{2}$. The worst-case computational complexity of our approach, after blocking, is now $O(Bn_{(B)}^2)$, where $n_{(B)}$ is the size of the largest block. (Note that $n_{(B)} \ll n$, so that the number of comparisons within each block b is feasible.) Assuming that $n_{(B)}$ is roughly constant with respect to n and that B increases linearly with n, then the run-time of the algorithm is now linear in n.

Furthermore, the operations performed within each block are independent of each other. That is, comparisons in block b_i do not have any effect on comparisons in block b_j, $i \neq j$. As a result, our record linkage procedure is trivially parallelizable across the B blocks. If we run this procedure on a machine with C processors, then the computational complexity is now reduced to (approximately) $O(Bn_{(B)}^2/C)$.

Appendix F. Identification of prolific inventors

Table 11 shows that our random forests approach yields the most consistently low, balanced false positive and false negative error rates across each of the OE sub-samples. Both the Lai et al. (2014) semi-supervised learning algorithm trained on statistically generated artificial labels and our random forests supervised learning algorithm perform well when disambiguating prolific inventors from the labeled OE inventor records dataset. In contrast, the two unsupervised, rule- and threshold-based algorithms yield higher false negative error rates (above 7%). Since disambiguating prolific inventors is an important task, we analyze each algorithm's disambiguation results for prolific inventors in more detail. In particular, we are interested in the number of top prolific inventors missed and added by each algorithm. We are also interested in the number of patents missed and added per top inventor, as well as the number of assignees missed and added per top inventor. In the following table, we compare these metrics for prolific inventors (defined here to be inventors with at least 20 patents) for each of the inventor disambiguation algorithms evaluated in this paper (Table 12).

Random forests and the Lai et al. (2014) posted disambiguation results outperform the Fleming et al. (2007) and Lai et al. (2009) unsupervised, rule- and threshold-based algorithms on most disambiguation metrics for prolific inventors. The number of inventors they mistakenly miss or add to a list of the top inventors is lower than the other algorithms. Similarly, the number of patents missed or added per top inventor is comparatively low for both algorithms. Since inventor mobility (across firms/assignees) is important in many research contexts (e.g. Marx et al., 2009), we also examined the number of assignees missed and added per prolific inventor for each disambiguation algorithm. Although most disambiguation approaches perform well here, note that random forests has the lowest average number of assignees missed, while Lai et al. (2014) has the lowest average number of assignees added per prolific inventor. Both of these algorithms perform well when disambiguating prolific inventors, supporting the evidence provided in Table 11.

F.1. *Evaluation of all algorithms on "Top 1%" versus "Rest" subsets of the ALS labeled data*

For completeness, to mirror Table 11, we partition the labeled ALS dataset into two subsets: ALS$_{top}$ and ALS$_{rest}$. Based on Azoulay et al. (2012), these two subsets should correspond approximately to the top 1% and remaining 99% of inventors in the USPTO, as defined

Table 12
Post-disambiguation results for prolific OE inventors (20+ patents through 2010).

Algorithm	Top inventors missed	Top inventors added	Patents missed per top inventor	Patents added per top inventor	Assignees missed per top inventor	Assignees added per top inventor
Random forests	6	4	0.78	0.44	0.09	0.25
Fleming et al. (2007)	13	9	2.60	0.10	0.37	0.06
Lai et al. (2009)	12	9	2.39	3.91	0.39	1.34
Lai et al. (2009) posted results	6	6	2.56	0.27	0.35	0.09
Lai et al. (2014) posted results	6	3	1.60	0.16	0.26	0.06

Table 13
Evaluation of all algorithms on labeled inventor records from ALS subsets.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random forests	Supervised	OE_{train}	ALS_{top}	ALS_{top}	0.00	0.00
Random forests	Supervised	OE_{train}	ALS_{rest}	ALS_{rest}	0.00	2.10
Fleming et al. (2007)	Unsupervised	NA	ALS_{top}	ALS_{top}	20.74	0.22
Fleming et al. (2007)	Unsupervised	NA	ALS_{rest}	ALS_{rest}	17.64	0.00
Lai et al. (2009)	Unsupervised	NA	ALS_{top}	ALS_{top}	9.16	1.73
Lai et al. (2009)	Unsupervised	NA	ALS_{rest}	ALS_{rest}	9.20	10.94
Lai et al. (2009) posted results	Unsupervised	NA	USPTO_{full}	ALS_{top}	0.12	0.00
Lai et al. (2009) posted results	Unsupervised	NA	USPTO_{full}	ALS_{rest}	0.52	0.74
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO_{full}	ALS_{top}	0.13	0.00
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO_{full}	ALS_{rest}	0.86	0.08

by the cumulative number of patents per inventor as of 2004. Inventors with 17 or more patents are in the top 1%, while inventors with fewer than 17 patents are in the remaining 99%. Results are shown in Table 13.

As in Table 8 where the algorithms were run on ALS without the subsets, here again random forests, Lai et al. (2009), and Lai et al. (2014) all perform well on the ALS datasets – whether the subset of inventors with more than 17 patents, or the remaining inventors with fewer patents. All three algorithms perform slightly worse on the subset of inventors with less than 17 patents than on the subset of inventors with more than 17 patents. As in Table 8, the Fleming et al. (2007) algorithm continues to have high splitting errors on both subsets, with slightly higher splitting errors on the top inventors with more than 17 patents than on the remaining 99%.

F.2. Evaluation of all algorithms on labeled inventor records from superstars subset of ALS

As noted in Section 3.1.2, the ALS database is the compilation of three data collection efforts: a labeled dataset of all members of the Association of American Medical Colleges who patent (Azoulay et al., 2007),[22] a labeled dataset of elite academic life scientists who patent (Azoulay et al., 2012), and any additional information available from a real-time data collection effort to update the information on each of these two populations to the current time. While in none of the three cases do Azoulay and co-authors have a curriculum vitae for every inventor, the effort with the most information for labeling inventors with their patents, and thus arguably the most accurate labels, is the labeled dataset of elite academic life scientists (Azoulay et al., 2012). In Tables 14 and 15 below, we look at this subset of the ALS dataset associated with elite academic life sciences who patent, and for which, of the ALS inventors, we arguably have the best labels. We show the disambiguation features of this subset of the labeled ALS data in Table 14. We then evaluate how each of the algorithms performs on this subset of elite academic life scientists within the ALS labeled data. Recall from Section 3.1.2, that although these are elite academic life scientists according to a set of criteria defined by Azoulay et al. (2012), not all of the elite academic life scientists are prolific patentors.

As can be seen in Table 14, when comparing the subset of elite academic life scientists who patent to the full set of inventors in the ALS dataset only two disambiguation features are strikingly different: on average the elite academic life scientists patent more (nearly twice as much), and move less (on average they only have one institutional location over the time span covered) than the broader set of inventors in the full ALS dataset.

As can be seen in Table 15, when compared to the disambiguation results on the full ALS dataset shown in Table 8, the random forest algorithm, Lai et al. (2009) algorithm, and Lai et al. (2014) algorithm all perform slightly better on the elite academic life scientists subset than they did on the full ALS dataset. This slight improvement is likely due to the subset of elite academic life scientists who patent being relatively easy to disambiguate due to their low mobility, and not due in any way to greater potential accuracy in the labeling process itself. There is no change in the performance Fleming et al. (2007) algorithm when implemented and evaluated on the full ALS data versus just the ALS stars subset.

Appendix G. Sources of error and variation in disambiguation algorithms

All disambiguation algorithms are subject to different types of error, but not all are subject to variation (e.g. changes in matches depending on the broader set of data on which the algorithm is run). We summarize differences in the types of error and possibilities for variation for each approach below.

Rule- and threshold-based approaches to disambiguation are subject to systematic error (and thus bias) resulting from the specific rules that they use. For example, many rule- and threshold-based approaches to disambiguation use “expert knowledge” of the underlying data or subject to create decision rules used for disambiguation. However, this “expert knowledge” can be incorrect (e.g. biased) or, alternatively, appropriate for particular contexts but not for others. Rule- and threshold-based approaches are deterministic, meaning that they will make the same decision for any pair of records regardless of the larger dataset within which those pairs are embedded. Thus, the overall splitting and lumping values might

[22] Note that Pierre Azoulay only provided us with access to the inventor labels. We had no access to the raw AAMC data.

Table 14
Inventor disambiguation statistics: optoelectronics, academic life sciences, academic life sciences superstars, and overall USPTO.

Disambiguation statistic	Optoelectronics	ALS	ALS stars	Overall USPTO
Number of records	98,762	53,378	23,894	9,358,182
Number of unique labeled inventors	824	15,202	3760	NA
Inventors per patent (mean)	2.86	2.70	2.91	2.21
Classes per patent (mean)	1.87	2.08	2.12	NA
Subclasses per patent (mean)	4.33	5.38	5.62	NA
Patents per labeled inventor (mean)	17.62	3.51	6.35	NA
Assignees per labeled inventor (mean)	3.90	3.30	1.00	NA
Length of last name (mean)	5.45	6.55	6.56	6.48
Length of first name (mean)	6.09	5.72	5.71	5.84
Percent of missing middle names	48.80%	19.02%	16.75%	51.10%
Percent of missing assignees	4.98%	0.00%	0.00%	9.02%
Percent of United States inventors	54.30%	98.93%	99.63%	50.36%
Percent of last names in census top 200	25.78%	10.56%	10.92%	8.34%

change when you run a rule- or threshold-based algorithm on a bigger or smaller datasets due to having different pairs within the dataset, but the decisions made about each of the record pairs will always be the same (match vs. non-match). As such, we do not provide "standard errors" on the splitting and lumping error rates for rule- and threshold-based approaches, since they will provide the same result each time when run on the data we have available.

Semi-supervised and supervised learning approaches to disambiguation are subject to different types of error in different steps of the algorithm. Both the Lai et al. (2014) semi-supervised algorithm and the random forests supervised algorithm described in Section 3.3.3 have the following three parts, each which may lead to specific types of error in the disambiguation results. We summarize these parts and their potential sources of error below.

G.1. Blocking

Both algorithms' first step is to partition the records into groups of loosely similar records, or "blocks." Comparison of records is only performed within blocks, and never across blocks; records that are not in the same block are assumed to be non-matches. This step will induce false negative errors if record-pairs that should be matched are split across blocks. Blocking schemes should be chosen to reduce this type of error and thus the rate at which these false negative errors occur. Note that blocking can also be part of the rules used in rule- and threshold-based approaches. Blocking is deterministic, and will group the same record-pairs together in a block regardless of changes in the overall dataset. That said, as a dataset grows, more pairs may be grouped with that pair in the block, which could lead to variation in the results in future stages of the algorithm (e.g. the clustering stage).

G.2. Link prediction – estimating pairwise probabilities of matching

The algorithm's second step is to estimate the probability of matching for all record-pairs in all blocks. To predict these pairwise probabilities of matching, Lai et al. (2014) use a logistic regression classifier, while we use a random forest classifier. The rates of false positive and false negative errors in this link prediction step of semi-supervised and supervised learning models depend on the representativeness, size, and usefulness of features in data on which they are trained. This step is not deterministic. If these algorithms are trained on different data, they will predict different pairwise probabilities of matching.

Given the variation depending on training data, an out-of-sample testing procedure, such as the one we use to help avoid overfitting to our training data is also a source of variation. The method we choose to split our overall labeled data into training and testing subsets can lead to variation in our final evaluation results. Similarly, to avoid over-dependence on choice of features, the random forest classifier is designed to repeatedly take random samples from the features and training data when building the classifier. This procedure, while beneficial to overall results, can also lead to additional variation.

Since the Lai et al. (2014) algorithm's training data uses a set of rules (determined by human/expert knowledge) to determine which record-pairs in the artificially labeled training data are matches and non-matches, we expect this algorithm to behave slightly differently than a semi-supervised algorithm trained on labeled data. Similar to rule- and threshold-based approaches, any bias in these rules could bias the chosen training data, which could in turn lead to biased models and prediction errors. We would not expect variation based on differences in the training data, since the training data here is rules-based. The exception would be if they changed their training data, such as by changing the rules or sampling from their artificial training data.

G.3. Hierarchical linkage clustering

Both algorithms' final step is to cluster the records based on their pairwise dissimilarities (calculated from the pairwise probabilities of matching). In this step, records with low dissimilarity (high match probability) are linked, and record-pairs with high

Table 15
Evaluation of all algorithms on labeled inventor records from superstars subset of ALS.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random forests	Supervised	OE_{train}	Stars_{test}	Stars_{test}	0.00	0.18
Fleming et al. (2007)	Unsupervised	NA	Stars_{full}	Stars_{full}	19.82	0.00
Lai et al. (2009)	Unsupervised	NA	Stars_{full}	Stars_{full}	10.47	3.15
Lai et al. (2009) posted results	Unsupervised	NA	USPTO_{full}	Stars_{full}	0.13	0.18
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO_{full}	Stars_{full}	0.27	0.00

Table 16
Evaluation of out-of-sample link prediction for supervised learning models (with standard errors).

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting % (standard error)	Lumping % (standard error)
Linear discriminant analysis	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	8.48 (0.17)	1.66 (0.14)
Quadratic discriminant analysis	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	3.19 (0.19)	1.62 (0.12)
Classification trees	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	2.22 (0.12)	2.49 (0.19)
Logistic regression	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	1.68 (0.08)	1.64 (0.11)
Random forests	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	0.61 (0.05)	0.73 (0.06)

dissimilarity (low match probability) are not linked, subject to some dissimilarity threshold. If the chosen threshold is too high, this may induce false positive errors, since a high threshold will match record-pairs at higher dissimilarity levels. Similarly, if the chosen threshold is too low, this may induce false negative errors, since a low threshold will impose a stricter dissimilarity level for matching. In the case of our random forest algorithm, we choose our threshold empirically based on out-of-sample testing, such that it will minimize the total number of false positive and false negative errors induced by the clustering step.

The clustering approach will always yield the same results if run on the same dataset. However, if the size of the dataset changes, then there could be variation in the clustering results. For example, when there is more data, additional information on new pairs can enable the hierarchical clustering algorithm to link previously unliked records into the same cluster. In hierarchical clustering, the way the results may change depends on the linkage method being used (e.g. single, average, complete, etc). We discuss this in Section 3.2. Both Lai et al. (2014) and our random forest algorithm use single linkage clustering.

Since Lai et al. (2014) do not provide the full code for their algorithm, we are unable to assess the standard errors associated with changes in the training data or with the dataset on which it is run for disambiguation. We also are unable to assess where the posted results' error rates are (e.g. mean, median, top of the range, bottom of the range) within the range of results associated with this algorithm's standard error.

We assess the standard error in our full supervised random forest algorithm empirically using cross-validation (Hastie et al., 2009). These standard errors represent the variations described above in the link prediction and clustering steps. The approach is discussed and results shown in Appendix H.

Appendix H. Standard errors on pairwise link prediction results for supervised learning approaches

Below, we include a version of Table 7 that includes standard errors on the splitting and lumping results, obtained via a version of cross-validation. As described in Hastie et al. (2009), cross-validation randomly partitions the data into training and testing subsets many times and obtains a statistical measure on the testing dataset (often the out-of-sample prediction error rates). In our case, the measures we obtain are the out-of-sample splitting and lumping rates. After obtaining the splitting and lumping results many times, we calculate an empirical cross-validated standard error on these metrics, shown in Table 16.

When randomly partitioning the data into training and testing subsets, we do not sample from the records directly, since for any given CV inventor, there are many near-duplicate inventor records. That is, we do not want these near-duplicates to be split across training and testing datasets, since including similar records in both subsets would artificially improve the accuracy of our models. Instead, we sample from the CV inventor IDs themselves, so that we have records from one random set of inventors used for training and records from another set of inventors used for testing. By doing so, we ensure that our error metrics are representative of what would occur if our disambiguation algorithm was used on another dataset in practice.

As Table 16 shows the standard errors on the prediction results of the different supervised learning methods are small. The random forest classifier was consistently better at pairwise link prediction than the other supervised learning approaches according to our splitting and lumping metrics.

Next, we obtain standard errors on the entire disambiguation procedure, using random forests as our classifier. The specific cross-validation procedure we use to obtain the standard errors is given below:

For *i* between 1 and 1000:

1. Randomly partition the data into training and testing subsets according to the procedure set forth in Section 3.1.1. We call these sets OE$_{train_i}$ and OE$_{test_i}$.
2. Train a classifier on pairwise comparisons from the OE$_{train_i}$ dataset.
3. Apply the classifier to the OE$_{test_i}$ dataset using the procedure described in Section 3.3.3. Note that for the results shown in Table 16, we only perform the pairwise link prediction step, ignoring the blocking and clustering steps, since our goal with this table is only to evaluate the performance of each classifier. In Table 17, we perform the entire disambiguation procedure since the goal is to evaluate the error of the entire procedure as a whole.

Table 17
Evaluation of random forests algorithm for inventor disambiguation (with standard errors).

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random forests	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	2.09 (0.78)	1.26 (0.41)
Random forests	Supervised	OE$_{train}$	ALS$_{test}$	ALS$_{test}$	0.00 (0.00)	0.80 (0.23)
Fleming et al. (2007)	Unsupervised	NA	OE$_{full}$	OE$_{full}$	13.50	0.68
Fleming et al. (2007)	Unsupervised	NA	ALS$_{full}$	ALS$_{full}$	19.82	0.00
Lai et al. (2009)	Unsupervised	NA	OE$_{full}$	OE$_{full}$	8.39	4.13
Lai et al. (2009)	Unsupervised	NA	ALS$_{full}$	ALS$_{full}$	9.16	6.79
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{full}$	9.18	0.76
Lai et al. (2009) posted results	Unsupervised	NA	USPTO$_{full}$	ALS$_{full}$	0.24	0.35
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{full}$	2.49	0.39
Lai et al. (2014) posted results	Semi-supervised	NA	USPTO$_{full}$	ALS$_{full}$	0.35	0.04

4. Evaluate the results with the splitting and lumping metrics described in Section 3.4.
5. Go back to step 1 until all 1000 iterations of this procedure have been completed.

After applying this procedure to the OE and ALS datasets, we are able to calculate empirical standard errors on the splitting and lumping results for our random forests disambiguation algorithm, shown in Table 17.

As we might expect, the standard errors for the ALS dataset are very low, since this dataset is relatively easy to disambiguate, meaning the algorithm consistently yields low error rates regardless in almost all iterations of the cross-validation procedure. For the OE dataset, the standard errors are higher, but not unexpectedly so. Since the OE dataset contains more difficult-to-disambiguate inventors, iterations of the cross-validation procedure sometimes randomly produce difficult-to-disambiguate OE_{test_i} datasets. This can lead to abnormally high splitting and lumping error rates for some iterations of the procedure, which can skew the standard errors.

Appendix I. Disambiguating the full USPTO database

Disambiguating the full USPTO is a complex and long-term research challenge for the field, and not one that can be overcome in a single paper alone. The collective solution to the USPTO is very hard for a number of reasons, including creating or obtaining appropriate training data as well as evaluating the outcomes of the algorithm on further labeled data. Disambiguating the full USPTO will require extensive resources. Google, for example, has the entire Google Scholar team trying to solve a similar problem, with the leverage and visibility of Google to get many inventors and authors to manually curate (e.g. self-identify) their own entries (e.g. patents and publications). (See https://medium.com/backchannel/the-gentleman-who-made-scholar-d71289d9a82d.)

While it is possible to run our random forest algorithm trained on OE on the full USPTO, we do not recommend using the disambiguated results that our current algorithm trained on OE would produce if run on the full USPTO for a number of reasons:

1. The labeled OE and ALS datasets were not collected with the intention of being training datasets for disambiguating the full USPTO. The labeled datasets were collected for theoretical studies within those specific industrial or institutional contexts. In both cases, the datasets only match the inventors to their patents within the field (OE patents or life sciences patents, respectively), and do not have information on patents the same inventors may have filed outside of the specific technical field. These datasets provide very good information on how to match inventors to their patents in the same field, but not very good information on how to match inventors to any patents they may have in other fields outside the industrial or institutional context of focus.
2. We have not focused on optimizing a blocking mechanism to minimize errors when disambiguating the full USPTO, as that was not the focus or intent of our original paper. (See the discussion of sources of error in Appendix G.) Other authors (e.g. Lai et al. (2014)) have proposed solutions to this problem, however.
3. Without additional labeled data, we are only able to evaluate the algorithm on how well it performed at disambiguating our labeled OE and ALS records. As many studies in our field are of specific industrial or institutional contexts (e.g. pharmaceuticals, semiconductors, optoelectronics, academics in the life sciences, etc.), disambiguation of smaller subsets is an important capability, especially in the context where disambiguation of larger datasets such as the full USPTO may have limitations.

The best long-term research approach to disambiguating the full USPTO is likely to be three-fold: (1) Increasing incentives for scholars contributing to disambiguation to make their full code and training data public so others can build on their work; (2) Developing funding mechanisms for a project dedicated to collecting representative, appropriately sized training data with useful features for disambiguation of the full USPTO (with the requirement that that training data subsequently be made public as is done in this project); Finally (3), in parallel, developing a repository where researchers, as they completed their individual industry- or context-specific studies would describe (e.g. the collection mechanisms and features of) and share their labeled data. This larger pool of labeled data could then be used by the TIE field and beyond to support better disambiguation.

While we do not have a labeled dataset collected with the intent of disambiguating the full USPTO and we have not optimized a blocking scheme for disambiguating the full USPTO, we do hope our paper points the way for further research leveraging labeled training data and machine learning techniques to disambiguate both smaller datasets as well as the full USPTO. In the case of smaller datasets, authors may choose to train on labeled data available to them personally and specific to their industrial or institutional context. In the case of larger datasets (including the full USPTO), future research may build on the framework presented in our paper, including both further development of machine learning algorithms (including evaluating their performance in different contexts and on different data set sizes against our own) and a focused effort on collecting labeled data representative of the full USPTO.

Finally, it is important to note when pursuing this future research agenda that the most important metric to users of the disambiguated data is the accuracy and sources of bias of the final IDs (e.g. matches) that they need to use to pursue their research question. From this user perspective (in contrast to the perspective of evaluating the algorithm), it does not matter which algorithm was used or on what size data it was run – only how accurate the final results are in the specific category from which results will be used. E.g., the user studying the optoelectronics industry only cares how good the final IDs are for inventors in optoelectronics. In this context, all comparisons evaluating final disambiguated results are relevant, regardless of the underlying means of achieving them (e.g. which algorithm on which size dataset).

Appendix J. Disambiguating the full USPTO database

We implemented our random forests inventor disambiguation algorithm on the full USPTO database[23] to further compare with the full USPTO implementations of the Lai et al. (2009) and Lai et al. (2014) algorithms. As discussed in Section 3.3.3, the third step of our disambiguation algorithm quantifies the similarity of

[23] Rather than scrape the full USPTO, we use the scraped and parsed USPTO data provided on by Lai et al. (2011) without using the IDs they create for each inventor. In running our algorithm on this dataset, approximately 5% of the records in our labeled optoelectronics dataset could not be identified in the Lai et al. (2011) database, likely due to differences in formatting standards and/or differences in the dates at which we originally scraped and parsed our optoelectronics data from the USPTO (as used in the rest of this study) versus when Lai et al. (2011) scraped and parsed the full USPTO data. For example, the Lai et al. (2011) database combines first and middle names into a single field, while our database splits these. Differences in the dates at which the databases were downloaded from the USPTO could affect which patent information is included in the database, since patent classes and other labelschange in the USPTO across time.

Table 18
Evaluation of all algorithms on labeled inventor records from OE sub-samples.

Algorithm	Type	Training dataset	Disambiguation dataset	Evaluation dataset	Splitting (%)	Lumping (%)
Random Forests	Supervised	OE$_{train}$	OE$_{test}$	OE$_{test}$	2.09	1.26
Random Forests	Supervised	OE$_{train}$	USPTO$_{full}$	OE$_{test}$	2.31	1.64
Lai et al. (2009) Posted results	Unsupervised	NA	USPTO$_{full}$	OE$_{full}$	9.18	0.76
Lai et al. (2014) Posted results	Semi-supervised	NA	USPTO$_{full}$	OE$_{full}$	2.49	0.39

each pair of records *within a given block* (i.e. never across blocks). Given that we only have labeled evaluation data available to us for optoelectronics, once we block the records in the full USPTO using the algorithm, and identify which of those blocks contain optoelectronics inventors, we only need to run the third step of our algorithm (quantifying the similarity of each pair) on those blocks with optoelectronic inventor records. Analyzing other blocks is unnecessary as they contain no optoelectronic records, and without additional labeled data, we have no way of verifying matches for records that are not in optoelectronics. We find that our random forest algorithm run on the full USPTO, maintains error rates within the standard errors reported in Table 18 for the same algorithm implemented strictly on the OE subset. The results for our algorithm implemented on the full USPTO is given in Table 18.

References

Abril, D., Navarro-Arribas, G., 2012. Improving record linkage with supervised learning for disclosure risk assessment. Inf. Fus. 13 (4), 274–284.

Akinsanmi, E., Reagans, R., Fuchs, E., 2014. Economic Downturns, Technology Trajectories, and the Careers of Scientists. Carnegie Mellon University Working Paper.

Azoulay, P., Michigan, R., Sampat, B.N., 2007. The anatomy of medical school patenting. N. Engl. J. Med. 357 (20), 2049–2056, http://dx.doi.org/10.1056/NEJMsa067417.

Azoulay, P., Zivin, J.S.G., Wang, J., 2010. Superstar extinction. Q. J. Econ. 125 (2), 549–589.

Azoulay, P., Zivin, J.S.G., Sampat, B.N., 2012. The diffusion of scientific knowledge across time and space: evidence from professional transitions for the superstars of medicine. In: Lerner, J., Stern, S. (Eds.), The Rate & Direction of Inventive Activity Revisited. University of Chicago Press, Chicago, IL, pp. 107–155.

Bessen, J., 2007. Programmer Documentation on PTO Assignee: Compustat Matching. National Bureau of Economic Research http://users.nber.org/jbessen/progdoc.pdf

Bessen, J., 2009. NBER PDP Project – User Documentation: Matching Patent Data to Compustat Firms. National Bureau of Economic Research http://users.nber.org/jbessen/matchdoc.pdf

Bhattacharya, I., Getoor, L., 2004. Iterative record linkage for cleaning and integration. In: 9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, http://dl.acm.org/citation.cfm?id=1008697

Bilenko, M., Mooney, R., 2003a. On evaluation and training-set construction for duplicate detection. In: Proceedings of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification, Washington, DC http://www.cs.utexas.edu/~ml/papers/marlin-kdd-wkshp-03.pdf

Bilenko, M., Mooney, R., 2003b. Adaptive duplicate detection using learnable string similarity metrics. In: Proceedings of ACM Conference on Knowledge Discovery and Data Mining, Washington, DC, pp. 39–48.

Bilenko, M., Mooney, R., Cohen, W., Ravikumar, P., Fienberg, S.E., 2003. Adaptive name matching in information integration. IEEE Intell. Syst. 18 (50), 16–23.

Bound, J., Cummins, C., Griliches, Z., Hall, B.H., Jaffe, A.B., 1984. Who does R&D and who patents? In: Griliches, Z. (Ed.), R&D, Patents, and Productivity. University of Chicago Press, pp. 21–54 http://www.nber.org/chapters/c10043

Bramer, M., 2007. Principles of Data Mining. Springer-Verlag Limited, London.

Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.

Carayol, N., Cassi, L., 2009. Who's Who in Patents: A Bayesian approach. http://www.researchgate.net/publication/228425987_Who's_Who_in_Patents._A_Bayesian_approach/file/d912f50a3c414cd487.pdf

Christen, P., 2006. A comparison of personal name matching: techniques and practical issues. In: Proceedings of the Workshop on Mining Complex Data (MCD), IEEE International Conference on Data Mining (ICDM).

Christen, P., 2008. Automatic record linkage using seeded nearest neighbour and support vector machine classification. In: Proceedings of the ACM SIGKDD 2008 Conference.

Cohen, W.M., Nelson, R.R., Walsh, J.P., 2000. Protecting Their Intellectual Assets: Appropriability Conditions and Why U.S. Manufacturing Firms Patent (or Not). National Bureau of Economic Research (Working Paper No. 7552), http://nber15.nber.org/papers/W7552

Criminisi, A., Shotton, J., Konukoglu, E., 2011. Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found. Trends Comput. Graph. Vis. 7 (2–3), 81–227.

Elfeky, M.G., Ghanem, T.M., Verykios, V.S., Huwait, A.R., Elmagarmid, A.K., 2003. Record Linkage: A Machine Learning Approach, A Toolbox, and A Digital Government Web Service. Purdue University e-Pubs – Computer Science Technical Reports http://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2572&context=cstech

Elmagarmid, A.K., Ipeirotis, P.G., Verykios, V.S., 2007. Duplicate record detection: a survey. IEEE Trans. Knowl. Data Eng. 19 (1), 1–16.

Fegley, B.D., Torvik, V.I., 2013. Has large-scale named-entity network analysis been resting on a flawed assumption? PLOS ONE 8 (7), e70299, http://dx.doi.org/10.1371/journal.pone.0070299.

Fellegi, I.P., Sunter, A.B., 1969. A theory for record linkage. J. Am. Stat. Assoc. 64 (328).

Fleming, L., King III, C., Juda, A., 2007. Small world and regional innovation. Organ. Sci. 18 (6).

Fleming, L., Singh, J., 2010. Lone inventors as sources of breakthroughs: myth or reality? Manag. Sci. 56 (1), 41–56.

Ge, C., Huang, K., Png, I., 2014. Engineer/Scientist Careers: Patents, Online Profiles, and Misclassification Bias. SSRN Working Paper 2531477. Revised, November. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2531477

Goiser, K., Christen, P., 2006. Towards automated record linkage. In: Proceedings of the Fifth Australasian Data Mining Conference (AusDM2006).

Gu, G., Lee, S., Kim, J., Marschke, G., 2008. Matching accuracy of the Lee-Kim-Marschke computer matching program. SUNY Albany Working Paper.

Hall, B., Jaffe, A., Trajtenberg, A., 2001. The NBER Patent Citations Data File: Lessons Insights and Methodological Tools.

Hall, B., Thoma, G., Torrisi, S., 2007. The Market Value of Patents and R&D: Evidence from European Firms. University of Camerino and CESPRI Bocconi University and Salvatore Torrisi, Bologna University and CESPRI Bocconi University.

Han, H., Giles, L., Zha, H., Li, C., Tsioutsiouliklis, K., 2004. Two supervised learning approaches for name disambiguation in author citations. In: Joint Conference on Digital Libraries 2004.

Hartigan, J.A., 1975. Clustering Algorithms. John Wiley & Sons, New York.

Hartigan, J.A., 1981. Consistency of single linkage for high-density clusters. J. Am. Stat. Assoc. 76 (374).

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. Springer-Verlag.

Jaro, M.A., 1978. UNIMATCH: A Record Linkage System, User's Manual. U.S. Bureau of the Census.

Jaro, M.A., 1989. Advances in record-linkage methodology as applied to matching the 1985 Census of Tampa, Florida. J. Am. Stat. Assoc. 84 (406), 414–420.

Jones, B.F., 2005. The Burden of Knowledge and the 'Death of the Renaissance Man': Is Innovation Getting Harder? National Bureau of Economic Research (Working Paper No. 11360), http://www.nber.org/papers/w11360

Kelley, R.P.,1984. Blocking considerations for record linkage under conditions of uncertainty. In: Proceedings of the Social Statistics Section. American Statistical Association, pp. 602–605.

Klevorick, A., Levin, R., Nelson, R., Winter, S., 1995. On the sources and significance of interindustry differences in technological opportunities. Res. Policy 24, 602–605.

Lai, R., D'Amour, A., Fleming, L., 2009. The careers and co-authorship networks of U.S. patent-holders, since 1975. Harvard Dataverse Network http://thedata.harvard.edu/dvn/dv/patent/faces/study/StudyPage.xhtml?studyId=38083&tab=catalog

Lai, R., D'Amour, A., Yu, A., Sun, Y., Fleming, L., 2011. Disambiguation and Co-authorship Networks of the U.S. Patent Inventor Database (1975–2010). http://hdl.handle.net/1902.1/15705UNF:5:RqsI3LsQEYLHkkg5jG/jRg== V3[Version]

Lai, R., D'Amour, A., Yu, A., Sun, Y., Fleming, L., 2014. Disambiguation and co-authorship networks of the U.S. Patent Inventor Database (1975–2010). Res. Policy 43 (6), 941–955.

Larsen, M.D., Rubin, D.B., 2001. Iterative automated record linkage using mixture models. J. Am. Stat. Assoc. 96 (453).

Levenshtein, V.I., 1966. Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Dokl. 10 (8), 707–710.

Lim, K., 2012. NUS-MBS Patent Database. http://kwanghui.com/patents/index.html

Lissoni, F., Sanditov, B., Tarasconi, G., 2006. The Keins Database on Academic Inventors: Methodology and Contents. Cespri – Universitá Bocconi (Working paper No. 181), http://www.francescolissoni.com/rp_g000004.pdf

Magerman, T., Van Looy, B., Song, X., 2006. Data production methods for harmonized patent statistics: patentee name harmonization. K.U. Leuven FETEW MSI Research Report 0605 http://epp.eurostat.ec.europa.eu/cache/ITY_OFFPUB/KS-AV-06-002/EN/KS-AV-06-002-EN.PDF

Martins, B., 2011. A supervised machine learning approach for duplicate detection for gazetteer records. Lect. Notes Comput. Sci. 6631, 34–51.

Marx, M., Strumsky, D., Fleming, L., 2009. Mobility, skills, and the Michigan non-compete experiment. Manag. Sci. 55 (6), 875–889.

Miguelez, E., Gomez-Miguelez, I., 2011. Singling out individual inventors from patent data. Research Institute of Applied Economics Working Paper.

Milojevic, S., 2013. Accuracy of simple, initials-based methods for author name disambiguation. J. Informetr. http://arxiv.org/abs/1308.0749

Newcombe, H.B., Kennedy, J.M., 1962. Record linkage: making maximum use of the discriminating power of identifying information. Comm. ACM. 5, 563–567.

Newcombe, H.B., Kennedy, J.M., Axford, S.J., James, A.P., 1959. Automatic linkage of vital records. Science 130, 954–959.

Newcombe, H.B., Smith, M.E., 1975. Methods for computer linkage of hospital admission separation records into cumulative health histories. Methods Inf. Med. 14 (3), 118–125.

Raffo, J., Lhuillery, S., 2009. How to play the "Names Game": patent retrieval comparing different heuristics. Res. Policy 38 (10), 1617–1627.

Sadinle, M., 2014. Detecting Duplicates in a Homicide Registry using a Bayesian partitioning approach. Ann. Appl. Stat. 8 (4), 2404–2434.

Sadinle, M., Fienberg, S.E., 2013. A generalized Fellegi–Sunter framework for multiple record linkage with application to homicide record-systems. J. Am. Stat. Assoc., http://dx.doi.org/10.1080/01621459.2012.757231.

Singh, J., 2005. Collaborative networks as determinants of knowledge diffusion patterns. Manag. Sci. 51 (5), 756–770.

Steorts, R.C., Hall, R., Fienberg, S.E., 2014. SMERED: a Bayesian approach to graphical record linkage and de-duplication. JMLR W&CP 33, 922–930.

Tang, L., Walsh, J.P., 2010. Bibliometric Fingerprints: Name Disambiguation Based on Approximate Structure Equivalence of Cognitive Maps. Springer – Scientometrics.

Thoma, G., Torrisi, S., Gambardella, A., Guellec, D., Hall, B.H., Harhoff, D., 2010. Harmonizing and Combining Large Datasets – An Application to Firm-Level Patent and Accounting Data. National Bureau of Economic Research (Working Paper No. 15851).

Torra, V., Navarro-Arribas, G., Abril, D., 2010. Supervised learning for record linkage through weighted means and OWA operators. Control Cybern. 39 (4) http://matwbn.icm.edu.pl/ksiazki/cc/cc39/cc3946.pdf

Torvik, V., Smalheiser, N., 2009. Author name disambiguation in MEDLINE. ACM Trans. Knowl. Discov. Data 3 (3), Article 11.

Trajtenberg, M., Shiff, G., Melamed, R., 2006. The Names Game: Harnessing Inventors' Patent Data for Economic Research. National Bureau of Economic Research (Working Paper No. 12479).

Treeratpituk, P., Giles, C.L., 2009. Disambiguating authors in academic publications using random forests. In: Joint Conference on Digital Libraries.

The United States Patent and Trademark Office, 2006. USPTO Assignee Harmonization. http://www.uspto.gov/web/offices/ac/ido/oeip/taf/data/misc/data_cd.doc/assignee_harmonization/_read_me_assignees_69_10Nov05.txt

2012. The United States Patent and Trademark Office. www.uspto.gov

Ventura, S.L., Nugent, R., Fuchs, E., 2014. Large-scale clustering methods with applications to record linkage. In: Proceedings of the Privacy in Statistical Databases Conference, https://dl.dropboxusercontent.com/u/18473260/PSD14%20(2).pdf

Winkler, W.E.,1988. Using the EM algorithm for weight computation in the Fellegi–Sunter model of record linkage. In: Proceedings of the Section on Survey Research Methods. American Statistical Association, pp. 667–671.

Winkler, W.E., 1989. Near automatic weight computation in the Fellegi–Sunter model of record linkage. In: Proceedings of the Fifth Census Bureau Annual Research Conference, pp. 145–155.

Winkler, W.E., 1990. String comparator metrics and enhanced decision rules in the Felligi–Sunter model of record linkage. In: Proceedings of the American Statistical Association, http://www.amstat.org/sections/srms/Proceedings/papers/1990_056.pdf

Winkler, W.E., 1995. Matching and record linkage. In: Cox, B.G., et al. (Eds.), Business Survey Methods. John Wiley, New York, pp. 355–384.

Zucker, L., Darby, M., 1996. Star scientists and institutional transformation: patterns of invention an innovation in the formation of the biotechnology industry. Proc. Natl. Acad. Sci. U. S. A. 95 (23), 12709–12716.

Zucker, L., Darby, M., Fong, J., 2011. Community Wide Database Designs for Tracking Innovation Impact: COMETS, STARS, and Nanobank. National Bureau of Economic Research (Working Paper No. 17404), http://www.nber.org/papers/w17404