Submitted to *Management Science*
manuscript MS-0001-2013.??

Methods Matter: Rethinking Inventor Disambiguation Algorithms with Classification Models and Labeled Inventor Records

Samuel L. Ventura
Department of Statistics, Carnegie Mellon University University, Pittsburgh, PA 15213, sventura@stat.cmu.edu
Rebecca Nugent
Department of Statistics, Carnegie Mellon University University, Pittsburgh, PA 15213, rnugent@stat.cmu.edu
Erica R.H. Fuchs
Department of Engineering & Public Policy, Carnegie Mellon University University, Pittsburgh, PA 15213, erhf@andrew.cmu.edu

Existing record linkage methods often use ad hoc weights, thresholds, and decision rules that may produce systematic errors in the results. We use a unique set of 98,762 labeled inventor records (records for which we know the true identification of the individual or entity) obtained from 824 United States Patent and Trademark Office (USPTO) optoelectronics inventors to evaluate the accuracy of these existing methods. We then develop and evaluate a new algorithm based on the Random Forests classification method to predict whether or not pairs of USPTO inventor records match. Our new Conditional Forest of Random Forests approach reduces false positive and false negative error rates by 84.5% and 92.7% respectively, over existing algorithms. Additionally, our errors do not occur systematically. This substantial reduction in disambiguation errors and their systematic occurrence suggests that research using the results of existing disambiguation methods should be revisited, as systematic bias in the errors may affect the conclusions of these and subsequent studies.

Key words: Inventor, Disambiguation, Patents, Classification, Random Forests

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; 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 inventors in the United States Patent and Trademark Office (USPTO)

database. However, current USPTO approaches fail to take advantage of recent methodological advancements in statistics, such as adaptations of the Fellegi-Sunter approach for record linkage. More importantly, many existing USPTO inventor disambiguation algorithms often use ad hoc weights, thresholds, and decision rules to determine matching record pairs instead of incorporating information from labeled records during disambiguation. Building models without using information from "labeled inventor records," or USPTO inventor records for which the true identify of the inventor is known, may cause prevalent and systematic errors in the disambiguation results.

Using a set of 98,762 labeled inventor records, we (1) evaluate existing inventor disambiguation algorithms to determine the rates of false positive and false negative errors in the disambiguation results, and (2) build and evaluate statistical classification models for inventor disambiguation using information from the labeled inventor records to inform the algorithm. We first evaluate the disambiguation efficacy for several different standard classification models. We then develop and implement a new variant of the most effective classification algorithm to further improve the disambiguation results.

Our results from evaluating existing inventor disambiguation algorithms show that even one of the most sophisticated USPTO inventor disambiguation algorithms (and one of the few that publicly posts both the algorithm and results for broader use) may be susceptible to high rates of false positive or false negative errors. We find that these errors systematically occur due to features of the existing algorithms, such as not accounting for typographical errors and using ad hoc matching thresholds which may allow too many or too few matches. Consequently, the algorithms systematically overestimate (in the first case) or underestimate (in the second case) important metrics such as inventor mobility (when an inventor moves to a new organization and/or location) or the total number of unique inventors in the USPTO database.

We next evaluate the results of our alternative classification approaches. We find that the Random Forests classification model, introduced by Leo Breiman in 2001, performs best according to our evaluation metrics. We then show that our variant method, "Conditional Forest of Random Forests," further reduces and balances the rates of false positive and false negative errors. In particular, our method reduces false positive and false negative error rates by 84.5% and 92.7% , respectively, over the publicly available inventor disambiguation algorithm described in Lai et al (2009).

Given (1) the potential issues with the existing inventor disambiguation algorithms and (2) the improvements that classification models that incorporate information from labeled inventor records yield, we recommend that research conclusions derived from disambiguation results of existing algorithms be reexamined to ensure that they have not been confounded or overly influenced by systematic errors inherent in these algorithms. Additionally, our results suggest that future record

linkage and disambiguation algorithms used in TIE, statistics, and other contexts should make greater efforts to incorporate information from labeled records during the linkage or disambiguation process to improve the accuracy of the results. Finally, it is imperative that TIE papers leveraging disambiguated data publish the algorithm used to create their data, and discuss the potential biases introduced therewith for the theory derived from their results.

2. Background

To review, record linkage refers to the process of linking records of unique entities across two or more data sources, while disambiguation refers to the process of linking records within a single database. Fellegi and Sunter proposed the first mathematical model for linking records across two databases, or "bipartite record linkage" (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.

Record linkage and disambiguation are key components of any research 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 US 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). Mauricio Sadinle and Stephen Fienberg extends Fellegi and Sunter's record linkage methodology to link information on crimes in Colombia across three criminal records databases (2013). Disambiguation methods have been applied to bibliographic databases such as MEDLINE (Torvik and Smalheiser 2009) to determine which papers and articles belong to which unique authors. Hall et al (2002) disambiguates assignee names in the USPTO database, and Zucker et al (2011) extends this by linking records of a disambiguated database of USPTO assignees to sources outside of the USPTO, such as Compustat. Finally, Lai et al (2009; 2011a) use disambiguation algorithms to create databases of inventors of USPTO patents.

Methodologically, there are some key differences between disambiguation and record linkage. Since disambiguation involves the use of only one database, all records being compared share the same fields. In record linkage, this may not be true – one data source might list an individual's first name and birthday, while another data source might list an individual's full name and address. Additionally, the distributions of values in each field may differ across data sources for record linkage. In record linkage, it is often assumed that each record in one data source can be linked to no more than one record from another data source. In disambiguation, this such assumption does not exist – there can be one, two, ten, or any other number of occurrences of each unique individual in the single data source. As such, the Fellegi-Sunter model for bipartite record linkage may not be applicable to many disambiguation problems. In both record linkage and disambiguation, the

key operations are comparisons of two or more records. The Fellegi-Sunter model, for example, compares pairs of records from each data source. Typically, when pairwise comparisons are made, the result of interest is whether each pair of records belongs to the same unique individual ("match") or to two different individuals ("non-match").

One feature that existing USPTO disambiguation algorithms have in common is that they usually do not incorporate information from *labeled* records during the linkage or disambiguation process. The approach we introduce here utilizes an extensive database of labeled records during the actual disambiguation process. Our algorithm uses information from these labeled records to identify which features of the record-pairs (e.g. the similarity of last names are across two records) best determine a match, rather than using subjective and potentially incorrect decision rules to determine which record-pairs match. In summary, we build statistical classification models that can predict whether pairs of inventor records match each other using a database of labeled inventor records.

2.1. Previous Work: Statistical Record Linkage

The Fellegi-Sunter model is the mathematical formalization of the record linkage approach previously described (qualitatively, not mathematically) by Howard Newcombe (1959, 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 (Fellegi and Sunter 1969).

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. Matthew Jaro led the computerized record linkage movement, creating "UNIMATCH," a computer system for implementing the Fellegi-Sunter 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. William 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 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 (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 (1989). Winkler introduced new string comparison metrics which allow for better handling of typographical errors across fields (1990). More recently, Rob Hall and Stephen Fienberg introduced probability models for record linkage which allow for valid statistical inference and enable the use of confidence intervals on the linkage results (Hall and Fienberg 2011). Finally, Sadinle and Fienberg introduce an extension of the Fellegi-Sunter model for linking multiple (three or more) data files and offer a theoretical framework for solving situations in which bipartite record linkage struggles due to the lack of transitivity of pairwise matches across databases (2013).

The Fellegi-Sunter approach for record linkage and its many extensions also have several weaknesses. First, it is difficult to apply to problems with three or more data sources due to violations of transitivity of matching (e.g., if the algorithm says that record A matches record B, record B matches record C, but record C does not match record A, transitivity of matching is violated). Sadinle and Fienberg's proposed extension of the Fellegi-Sunter model to situations with three or more databases addresses this (2013), but may be computationally unstable in a large database disambiguation scenario. Similarly, the standard Fellegi-Sunter model may not be appropriate to use in disambiguation problems since it assumes that each record from one data source can be linked to at most one record from another data source. Alternative statistical methods are needed to sove these issues. Additionally, it should be noted that the Fellegi-Sunter model does not explicitly use labeled records during the record linkage process.

As mentioned above, the Fellegi-Sunter model may not be appropriate for disambiguation applications. Specifically, since there can be any number of occurrences of each unique individual in the single data source used for disambiguation, the Fellegi-Sunter model is not applicable since it is designed specifically for situations where there can be at most one match across the compared data sources. Vetle Torvik and Neil Smalheiser deviate from the Fellegi-Sunter model while introducing several statistical concepts in their disambiguation of authors in MEDLINE, a database of medical journal articles (2009). In particular, they use logistic regression within a Bayesian framework to calculate pairwise matching probabilities, a weighted least squares algorithm to correct transitivity violations, and a maximum likelihood approach for detecting clusters of author-records that represent unique MEDLINE authors. One important step in this algorithm involves generating sets of pairwise comparisons of author-records 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, using their assumption of independence between the two sets, they have multiple sets of (assumed) unbiased training data. Note that this method approximates labeled data. Torvik and Smalheiser use these training sets of pseudo-labeled author

records to implement a supervised learning classification algorithm, which they then use to predict the labels (match vs. non-match) of record-pairs in MEDLINE. While this approach is innovative, there are some potential flaws. First, the independence of the two sets of fields and the unbiasedness of the training data sets needs further evaluation, as the assumption was not validated. Errors in the label approximation may be propagated in the classification model, if the model is trained on erroneously labeled records.

2.2. Previous Work: Record Linkage and Disambiguation in TIE

As discussed, record linkage and disambiguation are crucial in TIE. The USPTO maintains an online database of all patents issued in the US. 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. 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. 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, as discussed below in Sections 2.2.1 and 2.2.2, the record linkage and disambiguation methods applied in TIE are often different from the statistical record linkage methods discussed in Section 2.1. That is, the methods described below often involve ad hoc weighting schemes, thresholds, or decision rules. 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 (Lai et al 2011a). We discuss this issue in further detail in Sections 5 and 4.5.

2.2.1. Ad Hoc Assignee Disambiguation and Record Linkage. One topic of interest for the TIE literature is record linkage related to USPTO assignees. Assignee record linkage allows researchers to more accurately examine the patenting and innovation trends and behaviors of public and private firms, organizations, and government agencies. Assignee record linkage methods to-date have two main branches: (1) record linkage of assignees within the USPTO database, or "assignee disambiguation," and (2) record linkage of assignees to other data sources (i.e. linking USPTO assignees to non-USPTO data), or "cross-database assignee record linkage."

The most notable work in USPTO assignee disambiguation is that of Bronwyn Hall et al (2002) for the National Bureau of Economic Research (NBER), who published a database of standardized USPTO assignee names for use within the TIE and other research communities. Bessen (2007)

formalizes the disambiguation and matching procedures used in this work, documents the algorithm, and posts all code and related files. To perform the disambiguation, the authors designed an algorithm that used computerized routines (name cleaning and standardization operations, a word frequency algorithm, similarity score assignment, etc) combined with manual review of potentially matching assignee records. Like most assignee disambiguation algorithms, the approach does not incorporate any Fellegi-Sunter like approaches, modern statistical methods, or labeled comparisons of assignee records during the disambiguation process.

Other widely-used USPTO assignee disambiguation algorithms involve an extensive amount of (both computerized and by-hand) cleaning operations rather than advanced statistical record linkage methods. The USPTO itself released a set of files created to partially disambiguate assignee names in its database, in an effort to correct typographical and data scanning errors (2005). Similarly, only the results (and not the methods used) were published. The USPTO acknowledged that even with its own data-cleaning, many issues still were present: "No attempt has been made to combine data based on subsidiary relationships ... While every effort is made to accurately identify all organizational entities and associate patents with a single harmonized organizational name, achievement of a totally clean record is not expected, particularly in view of the many variations that may occur in corporate identifications (USPTO 2005)." Additionally, the Organisation for Economic Co-operation and Development (OECD) used more than 4000 cleaning operations to create a "harmonised" (i.e. cleaned or disambiguated) database of European patents (2000). The OECD post this database for public use, and they match the cleaned patent data with other data sources (e.g. patent assignee data to business registers). Note that the OECD methods involve both assignee disambiguation (within the USPTO database) and assignee record linkage (outside of the USPTO database). The record linkage methods used here are not published. This failure to publish the disambiguation methods used is common to many databases in the field (e.g. Lim 2012). Additionally, because the methods are unknown, it is impossible to determine how errors in the disambiguation results propagate throughout subsequent research based on these results. Finally, while some cleaning operations are necessary to assignee disambiguation and record linkage due to the systematic changes in assignee names across records or data sources, cleaning operations alone will not solve all issues related to assignee disambiguation. Given that the choice of disambiguation methods can dramatically influence the accuracy of the disambiguation results and likely biases therein, the lack of method documentation for smaller and author-specific databases is likewise concerning for accurate interpretation of subsequent research results. In particular, if a disambiguation algorithm is susceptible to any type of systematic errors, the conclusions of studies using these results may not be accurate.

Record linkage of USPTO assignees to other data sources is central to several areas of TIE research. Bronwyn Hall, Adam Jaffe, and Manuel Trajtenberg developed and made available a database of US patents and citations. The authors used a record linkage procedure to match the assignee information to Compustat, a database of all firms traded on the US stock market (2001). Similarly, Zucker et al linked assignee information across several sources of data, including US patents (granted and applications), grants, Thomson Reuters Web of Knowledge, and US doctoral dissertations. In the same work, assignee information from US patents was linked to ticker codes for public firms and other major databases with information about public firms. This linkage method primarily involves several data cleaning operations (e.g. correcting for misspellings, name variations, location changes) specific to this task (Zucker et al 2011) and does not use any of the statistical approaches discussed in Section 2.1. Similarly, this method does not use labeled assignee records during the disambiguation process.

In general, published databases of assignee disambiguation or record linkage results (and research using these results) have at least one (and often both) of the following two characteristics: (1) the disambiguation and/or record linkage methods used are not published, making it impossible to tell how potential disambiguation errors propagate in subsequent research, and (2) the disambiguation and/or record linkage methods used do not incorporate the modern statistical approaches (e.g. the Fellegi-Sunter model, probability-based linkage, supervised learning on labeled records) discussed previously.

2.2.2. Inventor Disambiguation. Our focus in this paper is on inventor disambiguation. In large-scale applications, disambiguation is usually done at least partially with an automated algorithm; the full USPTO database contains more than 8 million patents, each of which can have multiple inventors. The primary goal of any inventor disambiguation algorithm is to identify all records of all unique inventors in the database with a minimal number of non-systematically occurring errors[1].

USPTO inventor disambiguation algorithms in TIE currently follow three different branches: (1) simple data cleaning operations, exact string matching, and if-else decision-making to determine if record-pairs match; (2) thresholding linear combinations of comparison field similarity scores to determine if record-pairs match; and (3) implementation of the Torvik-Smalheiser (2009) approach described in Section 2.1. Although parts of these algorithms (e.g. linear combinations of similarity scores) are used in statistical record linkage, most of the previous inventor disambiguation work includes ad hoc methods for linking inventors across their patents.

[1] Many existing algorithms have errors that occur systematically. For example, any time an inventor changes organizations and/or locations (inventor mobility) many existing algorithms will systematically create two separate inventor IDs, one for each organization/location at which the inventor worked.

The first branch involves simple data-cleaning and exact matching techniques for disambiguation. Fleming et al's 2007 algorithm (Fleming, King III, Juda 2007) falls under the first branch of inventor disambiguation algorithms, as does the inventor disambiguation work done by the OECD, to harmonize inventors in Europe. The Fleming et al (2007) algorithm is described in detail in Section 3.3.1. 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 (2005). Benjamin F. Jones uses a similar approach (2005). Other researchers have used their own inventor disambiguation algorithms (e.g. Azoulay 2009 and Lim 2012), but their methodologies and results were not published. Finally, Fleming et al use a similar approach, incorporating information about the assingee and location of the inventors (Fleming, King III, Juda 2007). In all cases, these inventor disambiguation results were not posted for public use.

The second branch involves incorporating methods such as similarity scores and matching thresholds into disambiguation. In 2006, Francesco Lissoni, Bulat Sanditov, and Gianluca Tarasconi designed a method that incorporated more advanced disambiguation methods, such as similarity scores and thresholds for determining pairwise matches (Lissoni et al 2006). Trajtenberg et al use the SoundEx system to group names that are similar phonetically, then use similarity scores and matching thresholds to determine pairwise matches (2006). Fleming and his collaborators used a similar approach, calculating linear combinations of the similarity scores of each comparison field and using thresholds to determine pairwise matches (Lai et al 2009). These authors were the first to post a version of the USPTO inventor-patent database with disambiguated inventors for use in the research community. It is important to note that without Lai et al 2009 having made their algorithm and results public, our paper would not be possible. The Lai et al (2009) inventor disambiguation algorithm is described in further detail in Section 3.3.1.The linear weighting approach is similar to that of Winkler (1988, 1989a) and Jaro (1989), although the Lai et al (2009) algorithm does not use the EM algorithm for weight-calculation. In Section 4.2, we show that small changes to these hand-chosen weights and thresholds can substantially change the disambiguation results.

The third branch involves statistical approaches and the Torvik and Smalheiser approach (Section 2.1). Most recently, Lai et al (2011a) implemented an adaptation of the Torvik & Smalheiser approach for disambiguating authors in the MEDLINE database (2009). The Lai et al (2011a) algorithm marked the first time that an approach similar to supervised learning was used in inventor disambiguation, although it should be noted that the algorithm does not use actual labeled inventor records in the disambiguation process, only generated labels that are "highly likely" to be correct (see Section 2.1 for more details on the Torvik & Smalheiser approach). Lai et al (2011a) do not post the code and data for generating these labels, which would be necessary to mimic their algorithm and compare it against other algorithms. As with Lai et al (2009), the authors do post

the results of the Lai et al (2011a) algorithm online for use within the research community (Lai et al 2011b). We thus expect these results again to become widely used in current TIE research. Lai et al (2011a) use a set of labeled inventor records corresponding to 95 US-based academic inventors to test the results of their algorithm. We give our own evaluation of the results of this algorithm (along with those of Fleming et al 2007 and Lai et al 2009) in Section 4.1. Nicolas Carayol and Lorenzi Cassi use a Bayesian approach for disambiguating inventors of European patents (2009). They use a large benchmark dataset to minimize a linear combination of false positive and false negative errors in the results. Interestingly, the authors choose to use the benchmark dataset solely for evaluating their algorithm's disambiguation results instead of for building statistical models that could be used to minimize the error in the disambiguation results, as we do in this paper.

The improvements that the Lai et al (2009; 2011a) algorithms make in disambiguation, such as using statistical record linkage approaches like linear combinations of similarity scores, probability-based matching, and semi-supervised learning, cannot be understated. These algorithms set a new standard for disambiguation in the field of TIE, and the posted disambiguation results have helped further the frontiers of TIE research on inventors and innovation. Still, these algorithms do not use information from labeled records during the disambiguation process. Lai et al (2009) depends on ad hoc weights, thresholds, and decision rules, while Lai et al (2011a) uses only approximations of labeled records to train semi-supervised learning models. Thus, possibly most importantly, Lai et al's public posting of their algorithms (2009) and results (2009; 2011a) has enabled other researchers in the field to evaluate their approach and results and make improvements thereon.

2.3. Supervised Learning in Record Linkage

Torvik and Smalheiser (2009) tackle one difficult issue that plagues almost all record linkage and disambiguation methods: How do you calibrate (train) and evaluate (test) a record linkage model or disambiguation algorithm in the absence of a sufficient number of labeled records? Most current record linkage methodologies suffer from the lack of sufficient amounts of training data and cannot be accurately calibrated or evaluated in terms of the prevalence of false positive and false negative errors in the results. Existing methods sometimes compare disambiguation results against a small set of labeled records, but they do not incorporate the information from these labeled records during the record linkage or disambiguation process.

The method we adopt in this paper, using supervised learning techniques for inventor disambiguation, is a well-documented topic in statistics (e.g. Hastie et al 2009). Typically, supervised learning algorithms use labeled data to build statistical models for some outcome of interest (or "response variable"). Linear regression is an example of a supervised learning algorithm used for modeling continuous response variables, such as age, height, or income. In inventor disambiguation, the response variable we seek to model, whether or not a pair of records matches, is binary

(i.e. it has two possible outcomes, match or non-match). Supervised learning algorithms used for modeling binary or categorical response variables are called classification models. Similar to how a linear regression model can be used to predict a numerically continuous outcome for new data, the final classification model can be used to predict the categorical outcome of unlabeled data. In our case, we build classification models that determine whether or not record-pairs match using labeled inventor records. We then can use the resulting classifier to predict whether or not pairs of unlabeled inventor records match.

In addition to being able to predict the labels of (or classify) unlabeled record-pairs, classification models can identify features of the inventor records that are most useful in determining which record-pairs match. For example, classification models might tell us that an inventor's last name is more important in determining matching record-pairs than his/her location. Of the existing supervised learning methods used in modern statistics and machine learning, some commonly known classification models include logistic regression, classification trees, random forests, linear discriminant analysis, quadratic discriminant analysis, support vector machines, K nearest neighbors (Hastie et al 2009). The appropriateness of each method depends on the application. For example, logistic regression is often useful when information about the association between the features and the response is needed; estimated coefficients can be interpreted as the change in the log odds of the response falling into a particular category. Each of the classification models above is designed to yield optimized results by minimizing some statistical measure of error over a set of labeled data. Consequently, each classification method could yield different measures of error, depending on the application.

One issue preventing researchers from applying supervised learning techniques in record linkage and disambiguation is the lack of labeled data that can be used to train these models. Bilenko et al introduce automated methods for record linkage in situations when training data is available (2003). As mentioned in Section 2.2.2, Torvik and Smalheiser partially, but creatively, solve this issue by using artificial sets of training data. Hui Han et al use two supervised learning approaches – a naive Bayes probability model and the support vector machines classifier – to develop a model for determining pairwise matches for name disambiguation (2004). Pucktada Treeratpituk and C. Lee Giles extend this work and show that the use of the random forests classifier can substantially improve the disambiguation results (2009). However, these methods suffer from a common problem: It is difficult to obtain a large set of accurately labeled records on which to train. 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.

Given the potential benefits of a supervised learning approach, we choose to use a classification model to disambiguate USPTO inventors using a large set of labeled inventor records. To obtain enough data on which to build our classification models, we hand-label a large set of inventor records using information about inventors' employment, patenting, and location histories using their curricula vitae (CVs) and lists of their patents (Section 3.2.1). We convert these 98,762 labeled inventor records into a set of more than 20 million labeled pairwise comparisons of inventor records, or measures of how similar pairs of records are according to different fields, such as name, assignee, and location. Then, we build classification models on these labeled pairwise comparisons and use the resulting classifier to predict whether or not pairs of unlabeled records match (Section 3.3.3). We assess the methods' performance using cross-validated error metrics. Our large number of labeled records is a key advantage to our methodology and yields notable improvements when compared to other inventor disambiguation algorithms in the TIE field.

3. Methods

3.1. Research Framework

Our disambiguation methods utilize a set of 98,762 labeled inventor records corresponding to data collected for a case study of 824 inventors in the field of Optoelectronics (OE) by Eyiwunmi Akinsanmi et al (2012). First, we identify and label all inventor records corresponding to these 824 inventors and use these labeled records to evaluate existing inventor disambiguation algorithms. Second, we build and evaluate standard classification models for inventor disambiguation that learn from these labeled inventor records. Third, we design new classification models that can leverage distributional differences across patents and that are computationally tractable for large datasets. This process is shown in Figure 1.

We first describe the USPTO patent-inventor database, the subset of patents corresponding to the OE industry, and the extensive set of labeled inventor records from inventor CVs in Section 3.2. Then, we give an overview of all disambiguation methods evaluated in this work, including existing inventor disambiguation algorithms designed by Fleming et al (2007) and Lai et al (2009; 2011a) (Section 3.3.1); existing classification models applied to inventor disambiguation (Section 3.3.2); and new variants of one particular classification model empirically shown to yield the best results in this context (Section 3.3.3). We also describe the evaluation metrics used to test the results of each particular disambiguation method.

For the existing inventor disambiguation algorithms (Fleming et al 2007, Lai et al 2009, Lai et al 2011a), we evaluate the results over the entire dataset of labeled inventor records. For the classification models, we use out-of-sample testing and cross-validation to evaluate the disambiguation

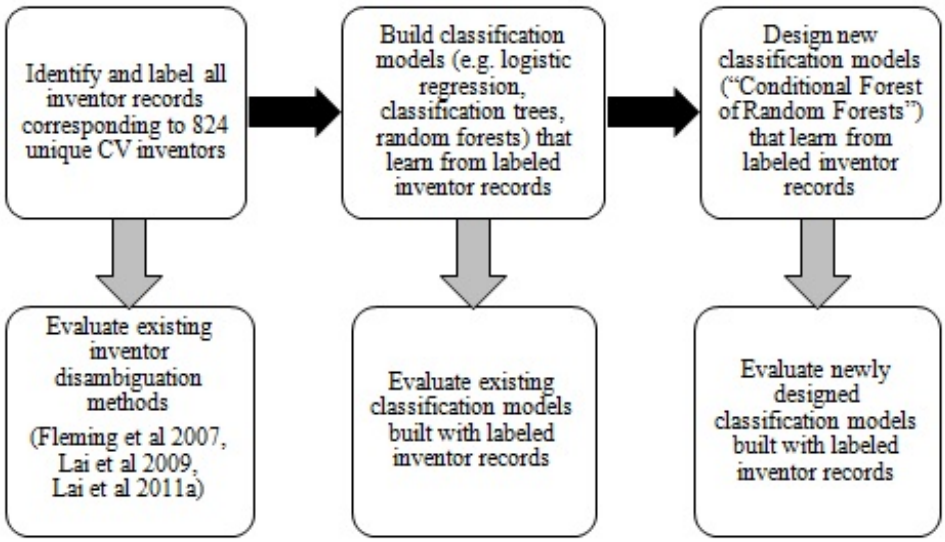

Figure 1 Research Framework Using Labeled Inventor Records

results. That is, to ensure that our classification models are not trained too specifically to one set of labeled records, we build our models using a subset of labeled inventor records, then evaluate using different subsets of labeled inventor records.

3.2. Data

TIE research often involves the analysis of information about patents, inventors, and assignees from the USPTO. 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), 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 were created by Lai et al (2009; 2011b). We utilize both versions of the database when evaluating inventor disambiguation algorithms.

We have access to extensive inventor resume and CV data in the field of OE, as discussed in further detail in Section 3.2.1 (Akinsanmi et al 2012). As such, we focus on a subset of the full USPTO database corresponding to the OE industry. Patents each have multiple classes and subclasses indicating the industry and technology categories that characterize the patent. These classes and subclasses are denoted with one to three character alpha-numeric codes that designate, among other information, the industry into which the patent's technology is classified. Using a list of classes and subclasses designated by the third author that correspond to the OE technical field,

we initially build a database of all USPTO OE patent information and a corresponding database of OE inventor records of all OE patents. This data is subsequently used to identify the sub-sample for the inventor CVs collected by Akinsanmi et al (2012). Table 1 compares some statistics relevant to inventor disambiguation for both the full USPTO database and the OE subset.

Table 1 Inventor Disambiguation Statistics: Optoelectronics vs. Full USPTO

Disambiguation Statistic	Optoelectronics	Full USPTO
Inventors per Patent (average)	2.33	2.21
Length of Last Name (average)	6.32	6.48
Length of First Name (average)	6.20	5.84
Percent of Missing Middle Names	62.00%	51.10%
Percent of Missing Assignees	4.08%	9.02%
Percent of United States Inventors	4.42%	5.04%
Percent of Last Names in Census Top 1000	20.40%	20.30%

Most statistics important to the disambiguation process, such as the average lengths of first and last names, are similar across the two datasets. Note that the percent of missing middle names is much higher in the OE subset than in the full USPTO database. This could be important to our results, since records with missing middle names are often more difficult to disambiguate.

3.2.1. Labeled Optoelectronics Inventor Records. As indicated earlier in this section, our labeled data comes from a study on economic downturns, inventor mobility, and technology trajectories by Akinsanmi et al, who collect a sample of resumes and CVs corresponding to 824 inventors in the OE industry (2012). Inventors in this sample come from one of four groups: 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 (on which Akinsanmi et al (2012) were focused for their study) and a random sample of inventors with no patents in subclass 385/14 (Table 2). For the groups corresponding to prolific inventors, the authors identify the top 1.5% of inventors using disambiguation results from an adaptation of the Lai et al (2009) algorithm. Note that these groups were chosen for the purposes of the research described in Akinsanmi et al (2012), not for the purposes of inventor disambiguation. As a given inventor can fall under more than one of each of the four inventor groups, there is some overlap between the populations, and likewise the CV samples. In total, 132 inventors were in two or more CV samples (most of which corresponded to a large overlap between the two prolific inventor groups). For the purposes of this paper, the overlaps of greatest relevance are overlaps with the group of random inventors without patents in class 385/14, since we use this group to evaluate sample bias in Section 4.6. There was very little overlap between this group and the other inventor groups: Only four inventors overlapped with the group of top inventors by rate of

patenting before 1999, and only two inventors overlapped with the group of top inventors by total patents before 1999. Finally, since nearly half of CV collection was focused on individuals from the two prolific inventor groups our sample is potentially biased towards prolific inventors, which could subsequently affect our models. We evaluate the potential consequences of biases in our sample in Section 4.6.

Table 2 Description of CV Inventor Groups

Group Description	Population	CV Sample	Response Rate
Top 1.5% of OE inventors by patent total through 1999	760	233	31% (73% of those reached)
Top 1.5% of OE inventors by patenting rate through 1999	680	229	34% (82% of those reached)
All OE inventors with at least one patent in 385/14	900	249	27% (95% of those reached)
Random sample of all OE inventors except those with at least one patent in 385/14	1250	169	14% (83% of those reached)

When contacting inventors in these groups, Akinsanmi et al request (1) the inventor's CV and (2) a list of all patents belonging to the inventor. After the CVs and patent lists are obtained, we create labeled inventor records in four steps. First, we manually parse and store information including the each inventor's employment, location, and patenting history. Second, we automatically generate a list of potentially matching inventor records for each of Akinsanmi et al's CV inventors, 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 (see Appendix B for more details on similarity scores). Third, we manually compare the parsed CV inventor's information to each of the potential matches to create labels for matching and non-matching inventor records. Finally, we review these match vs. non-match decisions to ensure label accuracy and, after removing duplicate records, compile the resulting labeled inventor records into a single dataset.

After the labeled inventor records were compiled, Akinsanmi et al (2012) assessed potential biases in the sample of inventors who responded. Of those biases found, only the following is relevant in the context of inventor disambiguation. For the second group, inventors in our sample are more likely to be more mobile before 1999. Mobile inventors may be more difficult to link across their patents due to changes in their location and/or assignee information. (We further evaluate the effect of potential biases in these samples in Section 4.6.)

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

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

Figure 2 Example of Labeled Inventor Records

inventors. Examples of labeled inventor records are shown in Figure 2. The ID column indicates the CV inventor to which each record was matched (a 0 ID indicates that the record did not match any of the CV inventors). 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).

We use our labeled inventor records to (1) evaluate the results of disambiguation algorithms (Sections 3.4 and 4), and (2) build classification models for inventor disambiguation to predict whether pairs of unlabeled inventor records match (Sections 3.3.2 and 3.3.3).

3.2.2. Pairwise Comparisons of Labeled Inventor Records. In almost all record linkage and disambiguation algorithms (e.g. Fellegi-Sunter, Lai et al 2009, Torvik-Smalheiser, 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 B 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. To evaluate an algorithm's disambiguation results, it is just as important to understand when two records should not be linked together as when they should.

3.3. Algorithms

Below, we provide greater detail on each of the disambiguation algorithms that we run on our OE dataset. First, we briefly review the three inventor disambiguation algorithms designed by Fleming et al (2007) and Lai et al (2009, 2011a), currently used widely in the field. Then, we introduce our

choice of inventor disambiguation methods that employ classification models using labeled inventor records.

3.3.1. Existing Inventor Disambiguation Algorithms. As discussed earlier, the first USPTO inventor disambiguation algorithm to be described publicly is that in Fleming et al (2007). The algorithm employs basic exact string matching and if-else decision making and allows for extremely fast computation (albeit with some systematic errors). An updated inventor disambiguation algorithm was created in 2009 by Fleming's group, and is described in full detail in Lai et al (2009). The Lai et al (2009) algorithm addresses the possibility of typographical errors and name variations. It uses disambiguation techniques such as similarity scores, field weights, and matching thresholds. In Section 4.2, we discuss the implications of slight changes to these weights and thresholds for the algorithm's results. Finally, the Lai et al (2011a) inventor disambiguation algorithm is an adaptation of the MEDLINE author disambiguation algorithm described by Torvik and Smalheiser (2009), with slight changes in the similarity scores and smoothing methods. These adaptations are necessary due to the different types of information available for author name data in MEDLINE versus inventor record data from the USPTO; they do not reflect any substantial deviation from the general Torvik and Smalheiser disambiguation procedure. None of the three existing algorithms described here use information from labeled inventor records during the disambiguation process.

We implement our own versions of the Fleming et al (2007) and Lai et al (2009) inventor disambiguation algorithms on our OE dataset and evaluate their results against our set of labeled OE inventor records in Section 4.1. For the Lai et al (2011a) algorithm, while most code files associated with the algorithm are available online, code for choosing the training datasets and training their semi-supervised logistic regression model – essential steps in the algorithm – are not publicly available. Thus, for (2011a) we are unable to implement our own version of the Lai et al (2011a) inventor disambiguation algorithm on our OE dataset and evaluate its results. We therefore also evaluate the posted results for implementing both the Lai et al (2009) and Lai et al (2011a) algorithms on the full USPTO data against our set of labeled OE inventor records. In order to achieve a legitimate comparison, it was important that we evaluated both the Lai et al (2009 and (2011a) algorithms implemented in the full USPTO, as any of the algorithms in this paper would have different results when implemented on a different scale database. Based on our direct communication with the authors, Lai et al are still finalizing the implementation of their (2011a) algorithm and examining its disambiguation properties. As such, our evaluations of the posted results associated with running the Lai et al (2011a) algorithm on the full USPTO against our labeled OE inventor records, which we give in Section 4.1, may change if Lai et al subsequently post new results.

3.3.2. 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 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 logistic regression, classification trees, random forests, linear discriminant analysis, and quadratic discriminant analysis (Hastie et al 2009). To avoid overfitting our models to the labeled inventor records dataset (so that our model will make accurate predictions on any dataset, not just this training data), we build each classification model using only a small, basic set of features: the similarity scores for each field (e.g. first and last name). Lai et al (2011a) use a different, larger set of features in their semi-supervised logistic regression models. Note that we could similarly choose a set of features by analyzing the importance of each feature in determining a match.

In Section 4.3, we evaluate each of these classification models for our labeled OE inventor records dataset. In doing so, we find empirically that the random forests classification model yields the most accurate results by a relatively wide margin over other models, in terms of false positive and false negative error rates. 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 decision tree from the selected features by determining cutpoints 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 receives the majority of the votes of the individual trees (Breiman 2001).

Despite the empirical success of random forests in disambiguating subsets of our labeled inventor records (Section 4.3), there are still some potential areas for improvement. First, random forests is a computationally intensive procedure. For example, if we compare all pairs of our labeled inventor records, we have more than 20 million pairwise comparisons. In practice, calculating a random forests classifier using 20 million observations is computationally infeasible. As discussed in the next section, using and combining multiple smaller random forests classifiers may provide a more computationally feasible solution. Second, we may be able to further improve random forest disambiguation results by conditioning on an informative feature of the records or comparisons and subsequently building separate random forests for each of the conditional subsets. We discuss these variants in the next section.

3.3.3. Conditional Forest of Random Forests. To alleviate the computational concerns of applying the random forests classification algorithm to our large set of pairwise comparisons of

labeled inventor records, we implement a variant called "Forest of Random Forests (FoRF) with Random Subsets." The FoRF algorithm partitions the pairwise comparisons into multiple smaller disjoint subsets. It then builds a standard random forests classification model on each of these subsets and aggregates the random forest classifiers during prediction. This method is implemented using the procedure described in Appendix C.1. By using random subsets of roughly equal size of the training data to build each individual random forest, we limit potential biases across the forests. Since this approach is suggested only for situations where the use of a standard random forest classifier is computationally intractable for large datasets, each individual random forest will be built on a dataset of sufficient size to yield stable predictions.

Aside from computational issues, there are additional changes we might incorporate to improve the accuracy of the disambiguation results. For example, a comparison of two US inventor records may have different matching properties than a comparison of two foreign inventor records. That is, different fields could be more or less important in determining whether two inventor records match depending on the inventor's country (due to name lengths, commonness of first and/or last names, etc). Instead of building one random forest classifier on all of the labeled inventor records, it may be helpful to build several random forests classifiers, each conditioned on a specific type of comparison of inventor records. By allowing the random forests to vary in determining feature importance, we may be able to enhance the accuracy of our disambiguation results.

For these situations, we implement a variant called "Conditional Forest of Random Forests." First, we determine a feature of the pairwise comparisons of labeled records on which to condition. For example, we might choose presence vs. absence ("missingness") of a middle name as our conditioning feature. Different disambiguation algorithms handle missing fields (i.e. NAs) in different ways. Instead of using ad hoc rules for handling missing data, we instead could train a classifier on each possible missing data scenario. For example, if we conditioned on missingness in middle name, there would be three conditional subsets: both middle names missing, one middle name missing, and neither middle name missing. In this example, the type of missingness would be known for unlabeled pairwise comparisons as well. Thus, after building the Conditional FoRF classifiers, we can predict the match vs. non-match labels of any pair of unlabeled inventor records by determining the appropriate conditional subset using the corresponding classifier. This procedure can allow the disambiguation algorithm to give more accurate predictions by using individualized decision rules for each conditional subset of pairwise comparisons of inventor records. This method is described formally in Appendix C.2. We evaluate the disambiguation results of Conditional FoRF in Section 4.4 and compare them to the standard random forest method.

3.4. Evaluation of Disambiguation Algorithms

Using our hand-disambiguated labeled OE inventor records for verification, we evaluate all described disambiguation algorithms. There are several possible ways to quantify the accuracy of each disambiguation algorithm. The evaluation metrics we use are described in Section 3.4.1. The out-of-sample evaluation methods we use for the classification-based approaches are described in Section 3.4.2.

3.4.1. Evaluation Metrics. We focus our evaluation on error metrics based on the numbers of false positive and false negative errors in the results. Lai et al (2011a) use Torvik and Smalheiser's (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 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 2011a):

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

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 2011a):

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

We use a revised version of these metrics in our evaluation for several reasons. First, the above metrics focus only on the largest cluster of records, ignoring thenumber and size of all the different clusters. For example, there may be another cluster of similar size for the same inventor. Additionally, this metric uses the number of incorrectly assigned inventor records as the unit of measure, instead of the number of incorrect matches or non-matches that the algorithm makes. 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 labels (match or non-match) and the predicted labels and then evaluate our results in terms of false positive and false negative pairwise comparisons.

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

Splitting: A single unique inventor is "split" into multiple inventor IDs

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

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 (or rate) of false positive matches. We will use these metrics throughout the remainder of this paper.

3.4.2. Out-of-Sample Testing for Classification Approaches. Classification models are by definition tailored to the data on which they are trained. To avoid overfitting to our training data, we split our set of pairwise comparisons of labeled inventor records into muliple smaller subsets. We then 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 (Hastie et al 2009). We describe our out-of-sample testing procedure, which is similar to cross-validation (Hastie et al 2009), below (results in Sections 4.3 and 4.4).

Our out-of-sample testing procedure consists of six steps. First, to construct our training data we calculate all pairwise comparisons of labeled inventor records. This is a very computationally intensive process, as discussed in Section 6. Second, we split the pairwise comparisons into 10 subsets of equal size, uniformly at random. Each of these subsets has more than 150,000 pairwise comparisons, about 14% of which are true matches and about 86% of which are true non-matches. This ratio is important, since many classification methods will not converge or will yield unstable results if the ratio of matches to non-matches in the training data is too small. Third, we train each classification method (e.g. logistic regression, classification trees, random forests, Conditional FoRF) on each pairwise comparison subset. Fourth, for each subset, we use the trained classifier to predict the match vs. non-match results for each of the nine other subsets of pairwise comparisons. Fifth, we evaluate each set of predictions using the splitting and lumping metrics described in Section 3.4.1. Finally, for each classification method, we average all 90 of its splitting metrics and all 90 of its lumping metrics to calculate the overall splitting and lumping results shown for each method in Tables 5 and 6.

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, in Section 4.6, we assess the robustness of our outcomes if we instead train and evaluate our algorithms only on the randomly chosen subset of inventors, which are, notably, also likely more representative of records in the full USPTO database. Specifically, we train and evaluate the performance our classification models on the subset of labeled inventor records corresponding to the group of random OE inventors with no patents in subclass 385/14. We use the same out-of-sample testing procedure described above but use only pairwise comparisons corresponding to the above-described random OE labeled inventor records.

4. Results

In the results that follow, we assess the accuracy of three different categories of disambiguation algorithms – existing inventor disambiguation algoritms, existing classification algorithms, and our new variant classification algorithm. In each case we run and then evaluating the algorithms on our dataset of 98,762 labeled inventor records. First, we evaluate existing, publically available USPTO inventor disambiguation algorithms (Fleming et al (2007) and Lai et al (2009)) in Section 4.1. Second, we analyze the sensitivity of the Lai et al (2009) disambiguation results to the algorithm's weight and threshold parameters (Section 4.2). Third, we evaluate disambiguation results of several common classification models built using our pairwise comparisons of labeled inventor records (Section 4.3). Fourth, we evaluate the performance of our new Conditional FoRF classification method for inventor disambiguation (Section 4.4).

Given that the code for the Lai et al (2011a) algorithm is not publically available, assessing its accuracy is largely outside the scope of this paper. Nonetheless, we wanted to shed what insights possible, given publically available information. The evaluation results of all disambiguation algorithms are dependent on the dataset being evaluated and the dataset of labeled records used for evaluation. We are unable to implement the Lai et al (2011a) algorithm (whose code is not fully available) on our OE dataset. It is likewise in the short term computationally infeasible to run our methods on the full USPTO dataset. We can, however, evaluate the accuracy of the Lai et al (2011a) posted results using our labeled OE inventor records (Section 4.5). For comparison purposes, we do the same for the Lai et al (2009) posted results. (These posted results correspond to running the Lai et al 2009 and 2011a algorithms on the full USPTO database rather than on our smaller OE dataset.) We then compare the Lai et al (2009 and 2011a) posted results to Conditional FoRF using these accuracy evaluations based on our labeled OE inventor records (Section 4.5).

Finally, we assess the influence of sample selection on the performance of our disambiguation approaches (Section 4.6).

4.1. Evaluation of Existing USPTO Inventor Disambiguation Algorithms

Using the splitting and lumping metrics described in Section 3.4.1, we evaluate the disambiguation results of two existing USPTO inventor disambiguation algorithms. In particular, we evaluate our implementations of the Fleming et al (2007) and Lai et al (2009) algorithms. Recall that there are 98,762 labeled inventor records in our dataset.

Table 3 Evaluation of Existing Inventor Disambiguation Algorithms

Existing Algorithm	Splitting (%)	Lumping (%)
Fleming et al (2007)	13.50	0.68
Lai et al (2009)	8.39	4.13

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. In these results, some 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 inventors compiled using the algorithm's results may be incomplete and inaccurate. The algorithm also likely overestimates the number of unique inventors. Finally, inventor mobility may be underestimated in the Fleming et al (2007) results. In particular, the false negative errors in the disambiguation results often 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.

The Lai et al (2009) algorithm results are a bit more complicated, since both the splitting and lumping metrics are high, indicating a prevalence of both false negative and false positive errors. 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. Unfortunately, these two types of errors do not simply cancel each other out. Instead, they bias the disambiguation results in both ways. Consequently, lists of the most prolific inventors compiled using the algorithm's results are likely incorrect, as is an estimate of the total number of unique inventors based on these results. Finally, inventor mobility is poorly estimated using the Lai et al (2009) results. In particular, the false positive errors in the disambiguation results can 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 match. This decision rule will lump groups of non-mobile inventors who have similar names and happen to share another characteristic into a single unique inventor ID. Similarly, the false negative errors in the disambiguation results can occur systematically if the thresholds for determining pairwise matches are too strict.

Note that these results are based on our set of labeled OE inventor records. As discussed in Section 4.6, these results may not reflect the disambiguation of non-OE USPTO inventor records.

4.2. Lai et al (2009) Sensitivity Analysis

Many disambiguation approaches risk overfitting to the labeled datasets on which they are evaluated. Disambiguation approaches using linear weighting schemes, for example, may tune the weights to minimize errors on a specific dataset; however, these weights may not be optimal for other datasets. As discussed in Appendix 3.3.1, the Lai et al (2009) algorithm depends on a set of weights and thresholds for its linear weighting scheme of similarity scores. In our implementation of the Lai et al (2009) algorithm, we explore how changes to these weights and thresholds can affect the disambiguation results. We run eight different versions (described in Table 4) of the algorithm on our labeled OE inventor records. The results are summarized in Figure 3.

Table 4 Lai et al (2009) weights and thresholds used for sensitivity analysis

Version #	Description	Weights	Thresholds
Default	Default Lai et al (2009)	Last name = 0.40 First name = 0.40 Middle name = 0.10 Suffix = 0.05 Other = 0.05 City = 0.60 State/Country = 0.40 Assignee Name = 0.90	Name = 0.90 Location = 0.95 Assignee = 0.85
Version #	**Description of Changes (From Default)**	**Weight Changes (From Default)**	**Threshold Changes (From Default)**
1	Shift first name weight to last name	First = 0.20 Last = 0.60	NA
2	Shift last name weight to first name	First = 0.60 Last = 0.20	NA
3	Stricter name threshold	NA	Name = 0.975
4	Less strict name threshold	NA	Name = 0.875
5	Less strict location and assignee thresholds	NA	Location = 0.80 Assignee = 0.75
6	Stricter location and assignee thresholds	NA	Location = 0.975 Assignee = 0.95
Optimal Thresholds	Less strict name, location, and assignee thresholds	NA	Location = 0.812 Assignee = 0.861 Name = 0.912

Although the default version of the Lai et al (2009) algorithm performs well in comparison to other versions, minor changes in the weights and thresholds of the algorithm yield substantial changes in the splitting and lumping results. Interestingly, changing the location and assignee matching thresholds did not affect the results much. However, changing the name weights and thresholds increased the number of false positive or false negative errors substantially in most

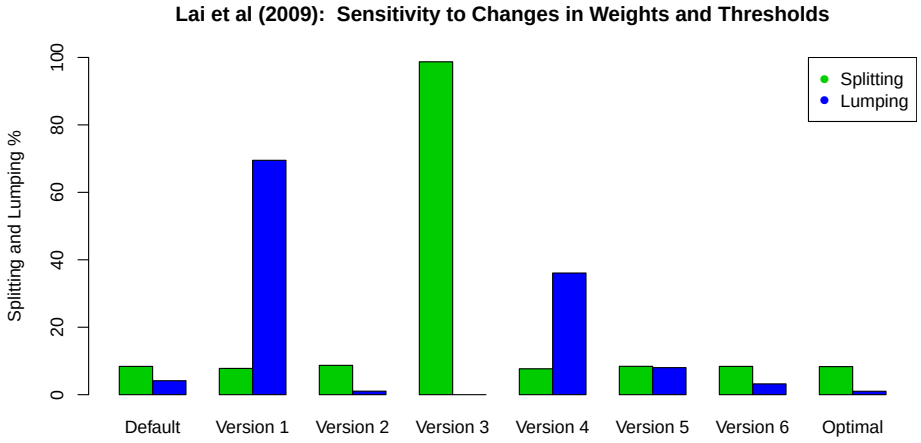

Figure 3 Splitting and Lumping Results for Lai et al (2009) Sensitivity Analysis

cases. Version 1, in which some weight was shifted from last name to first name similarity, yielded a large increase in the rate of false positive errors (lumping), indicating, as expected, that last name similarity should not be down-weighted. Versions 3 and 4 of the algorithm yielded substantial changes in splitting and lumping, respectively. However, these versions involved only minor changes to the overall name threshold: 0.95 to 0.975 in Version 3, and 0.95 to 0.90 in Version 4. Ideally, disambiguation algorithms using weighting schemes would be insensitive to such small changes in parameter values. If they are sensitive to these changes, a more robust disambiguation approach may be necessary.

We also calculated the "optimal" thresholds (i.e. the set of thresholds which best lowers the resulting splitting and lumping metrics) for Lai et al (2009). These optimal thresholds were calculated by automatically determining the threshold in each field's similarity score at which it is more likely for a pair of records to be a match than a non-match. For example, we found that the optimal location threshold should be approximately 0.812, where the location similarity score is a weighted combination of the state/country and city similarity scores. The optimal-threshold version of Lai et al (2009) yielded splitting and lumping metrics of 8.08% and 1.51%, respectively. It should be noted that the default version of Lai et al (2009) performed almost as well as the version with optimized thresholds.

4.3. Evaluation of Classification Models for Inventor Disambiguation

Using pairwise comparisons of labeled inventor records, we build and evaluate five commonly used classification models. The results shown in Table 5 are based on out-of-sample predictions of each classification method (Section 3.4.2). 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 feautres that best separates two or more classes of objects or events (e.g. match vs. non-match). Logistic regression was used in Lai et al's (2011a) semi-supervised leang approach and is one of the most well-known classification methods for binary responses. Classification trees and random forests are described in Section 3.3.2. Each of these methods is described in further detail in Hastie et al (2009).

Table 5 Evaluation of Classification Models for Inventor Disambiguation

Disambiguation Method	Splitting (%)	Lumping (%)
Fleming et al (2007)	13.50	0.68
Lai et al (2009)	8.39	4.13
Linear Discriminant Analysis	8.48	1.64
Quadratic Discriminant Analysis	3.19	1.62
Classification Trees	2.23	2.49
Logistic Regression	1.68	1.64
Random Forests	0.62	0.74

Note: The classification model results shown above are pre-transitivity due to computational restrictions and to facilitate out-of-sample testing. Post-transitivity results were calculated for some classification methods, and the results did not change substantially.

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 give unbiased estimates of inventor mobility. 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 exception of good performance regardless of the subsequent contextual application.

Given this goal, each of the classification methods perform about as well as (and, in most cases, better than) the existing inventor disambiguation algorithms evaluated here, based on a balance of low splitting and low lumping. Random forests outperforms other classification methods by a relatively wide margin. Note that although the Fleming et al (2007) results have a lumping metric similar to that of random forests, random forests performs better overall given the large disparity between the splitting metrics. We choose to focus on developing variants of the random forest method for improved inventor disambiguation.

4.4. Evaluation of Conditional FoRF for Inventor Disambiguation

Recall that a Forest of Random Forests with random subsets allows us to use the random forests classification method in computationally intractable situations. To evaluate this variant, we compare its evaluation metrics to the standard random forests methods using 10-fold cross validation on a randomly chosen subset of 1,500,000 pairwise comparisons. This subset is small enough that 10-fold cross validation with random forests is feasible, and large enough so that individual forests in the FoRF will have enough training data. We run FoRF with 15 random subsets, so that within each fold of the cross-validation, each random forest will have about 10,000 pairwise comparisons. Results are shown in Table 6. As expected, the standard random forests slightly outperforms FoRF with random subsets, indicating that the FoRF is a reasonable variant in computationally difficult situations.

Before building and evaluating a Conditional FoRF, we must first decide which conditions to use. In this example, we choose to condition on whether a middle name is missing for the two records, since different combinations could change the feature importance in determining matches and non-matches. There are three possible conditions of missingness in middle name: (1) both middle names are missing, (2) one middle name is missing, and (3) neither middle name is missing. (For these purposes, any record that does not list a middle name would be characterized as having a "missing" middle name.) For example, if two records have middle names listed, then the similarity score of these middle names should inform our decision of whether or not to match these two records. However, if neither has a listed middle name, then the importance of the middle name similarity will likely change. Additionally, missingness in middle name is a known feature of both the labeled and unlabeled pairwise comparisons, meaning that once we build our Conditional FoRF classifier on labeled pairwise comparisons, we can easily predict the match vs. non-match outcome of unlabeled pairwise comparisons. Recall that with a Conditional FoRF and a known condition, we use only the forest corresponding to the appropriate conditional subset when predicting the match vs. non-match outcome of unlabeled pairwise comparisons of inventor records. We expect that these predictions will be more accurate than the standard random forests predictions since each forest in the Conditional FoRF is tailored specifically to the feature distribution corresponding to its middle name missingness category. The prediction results of the Conditional FoRF are shown in Table 6.

The Conditional FoRF improves the disambiguation results over the standard random forests approach by reducing the lumping metric from 0.74% to 0.64%. Additionally, Conditional FoRF reduces splitting and lumping by 92.7% and 84.5% respectively over Lai et al (2009), based on our set of labeled OE inventor records. As discussed in Section 4.6, these results may not reflect

Table 6 Evaluation of Conditional Forest of Random Forests

Prediction Method	Splitting (%)	Lumping (%)
Fleming et al (2007)	13.50	0.68
Lai et al (2009)	8.39	4.13
Random Forests	0.62	0.74
FoRF	1.06	1.05
Conditional FoRF	0.61	0.64

Note: The random forests, FoRF, and Conditional FoRF results shown above are pre-transitivity due to computational restrictions and to facilitate out-of-sample testing. Post-transitivity results were calculated for some classification methods, and the results did not change substantially.

the results of the disambiguation of non-OE USPTO inventor records. However, we expect that Conditional FoRF will similarly outperform the other methods.

Additionally, we examine the variable importance statistics returned by random forests for each of the three forests (both middle names missing, one middle name missing, neither middle name missing) in the Conditional FoRF. These quantities measure the relative importance of each variable in determining whether or not each pair of records is a match or a non-match, with more important features having higher statistics. Note that it is not appropriate to compare the individual importance statistics across forests; rather, we compare the variable importance statistics across different features (last, first, and middle name similarities) within the same forest and check their ratios relative to each other across forests. We expect the ratios of importance statistics to change across the forests depending on the three different missingness conditions in middle name.

Table 7 Random Forests Variable Importance

Forest / Variable	Last	First	Middle
Both Missing	113.56	2806.99	0.00
One Missing	158.16	1146.27	0.00
None Missing	743.25	8356.24	2833.54

When one or both middle names are missing, the middle name comparison score has zero importance. When both middle names are present (none missing), the middle name importance becomes non-zero. This behavior confirms our assumption that the conditional forests will accurately reflect any feature distributional differences in matching properties across the different subsets. Finally, note that in this example, last name importance seems unrealistically low compared to first name. This is simply because the labeled records corresponding to this sample generally all have very similar last names, while their first and middle names differ substantially.

Next, it should be noted that we have taken several measures to avoid overfitting our models to our set of pairwise comparisons of labeled inventor records. First, we use only a small set of

explanatory variables to model the match versus non-match outcome of a pair of records. We use the similarity scores described in Section 3.2.2 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. Second, we use out-of-sample testing via the procedure described in Section 3.4.2 when calculating all classifier error metrics. Third, we utilize our sample of CVs corresponding to random OE inventors to show that our methods are not biased towards matching prolific inventors (Section 4.6). Given these approaches, our models are less likely to suffer from overfitting to the pairwise comparisons of labeled inventor records, and more likely to give robust results.

4.5. Comparing FoRF to Lai et al (2011a)

The Lai et al (2011a) algorithm is a semi-supervised learning approach; it is statistical in nature but relies on approximated labels during the disambiguation process. Algorithmically, there are four key differences between our FoRF approach and the Lai et al (2011a) approach. First, our method uses supervised learning techniques, while Lai et al (2011a) use the semi-supervised learning techniques from Torvik and Smalheiser (2009). In particular, we use an extensive set of hand-labeled inventor records to train our classifiers, while Lai et al (2011a) use generated sets of probable "matches" and "non-matches" for pairs of records. Second, Lai et al (2011a) use logistic regression to classify pairs of records as matches or non-matches, while we use random forests, shown empirically here in Section 4.3 to have better error rates. Third, we use a more parsimonious set of features and similarity scores, while Lai et al (2011a) use a larger feature set of similarity scores that includes several non-linear transformations and discretizations, which may make their approach more susceptible to overfitting. Finally, Lai et al (2011a) use one classifier to predict whether or not all pairs of records match. We explore conditioning on features of the records and similarity scores and train classifiers on each of these conditional subsets, allowing our Conditional FoRF method to exploit different sets of matching properties. As shown in Section 4.4, this can give more accurate predictions in the inventor disambiguation context.

In practice, another key difference is that the Lai et al (2011a) algorithm has been used to disambiguate the full USPTO database, while our Conditional FoRF approach has so far been used only to disambiguate OE inventor records. Because of the size of the full USPTO database, the Lai et al (2011a) algorithm uses computational reduction techniques such as blocking, which strategically limits the number of comparisons the algorithm makes. Currently, the Conditional FoRF approach does not use techniques like blocking.

These computational differences are important to keep in mind when comparing the error rates of the two approaches, since applying either algorithm to datasets of different sizes could change the error rates. In particular, for the Lai et al (2000 and 2011a) posted full USPTO disambiguation

results, these algorithms are able to take advantage of more patent-inventor information to link inventor records together, which could potentially lead to better evaluation metrics. In Lai et al (2009), for example, some non-OE inventor record in the full USPTO database might help match two OE inventor records which previously would not have been linked (via transitivity of matching). In Lai et al (2011a), the algorithm may not have enough training data when used to disambiguation smaller databases.

Our evaluations of the posted Lai et al (2009; 2011a) results are in Table 8. Note that for Lai et al (2009 and 2011a) posted results, these metrics represent our evaluation of the authors' full USPTO database disambiguation results posted online, not our implementation of these algorithms on our set of labeled OE inventor records. We extract the resulting labels from the full USPTO disambiguation corresponding to our set of labeled OE inventor records and evaluate their error rates.

Table 8 Comparing Conditional FoRF to Lai et al (2011a)

Disambiguation Method, Full USPTO Posted Results	Splitting (%)	Lumping (%)
Lai et al (2009) Posted Results, OE Subset	9.18	0.76
Lai et al (2011a) Posted Results, OE Subset	2.49	0.39
Disambiguation Method, Implemented on OE Subset	**Splitting (%)**	**Lumping (%)**
Lai et al (2009)	8.39	4.13
Random Forests	0.62	0.74
Conditional FoRF	0.61	0.64

Note: Lai et al (2011a) results are taken from the authors' interim results and may not reflect the performance of the final algorithm

While the Lai et al (2011a) posted results are better than those of Lai et al (2009) based on a reduction in both error rates, note that the splitting metric is still relatively high. Similar to Fleming et al (2007) and Lai et al (2009), this prevalence of false negative errors in the results indicates that some inventors are not getting credit for all of their patents. As before, lists of the most prolific inventors compiled using these results may be incomplete and inaccurate, the algorithm likely overestimates the total number of unique inventors, and inventor mobility may be underestimated, although the extent of these issues is lessened in comparison to Fleming et al (2007) and Lai et al (2009). Table 8 still indicates that Conditional FoRF is likely to outperform Lai et al (2011a) on both a balance and an overall reduction of false positive and false negative errors.

4.6. Assessing the Influence of Sample Selection on Disambiguation Performance

As mentioned in Section 3.2.1, many of the unique inventors in our set of labeled inventor records were taken from lists of the most prolific inventors. Recall that (1) the inventor CVs were originally collected for a separate study on economic downturns, inventor mobility, and technology trajectories in OE (Akinsanmi et al 2012), and (2) prolific inventors by definition have the most patents, meaning that CVs collected from these inventors will yield more labeled records (on average) than CVs collected from a randomly chosen group of inventors. In particular, these inventor CVs correspond to inventors who are prolific in terms of either the total number of patents in their career or their rate of patenting. Thus, our set of labeled inventor records may be biased toward inventors with prolific patenting careers.

Subsequently, our classification models could be similarly biased. Specifically, if comparisons of our labeled inventor records have any characteristics unique to comparisons of prolific inventors, then the predictions (match vs. non-match) for unlabeled comparisons of records may be appropriate only comparisons of prolific inventors, a potential issue when disambiguating the full USPTO database.

To assess this potential bias, we examine how well our methods perform for comparisons of labeled inventors from a sample of 169 random inventors without patents in class 385/14 (RI). The RI sample is assumed to be representative of the general OE population and thus should not be biased towards prolific inventors. We expect to see some small changes in the resulting error rates when our methods are applied to this sample. In particular, since randomly chosen inventors should have fewer total patents on average than prolific inventors, we expect the algorithm to favor predicting non-matches to matches. (Note that a dataset with a small number of matching pairs is naturally more difficult to disambiguate, since there is less information about what characteristics are associated with record-pairs matching.) More non-matches will likely result in lower false negative error rates (splitting %) and higher false positive error rates (lumping %). Note that this issue – having a disproportionate ratio of one binary outcome to the other – is a common problem in statistical literature (e.g. Hastie et al 2009). Future work will test adjustments to our classification models to account for this potential disproportion. For the sake of comparison, we also include an evaluation of the Lai et al (2009 and 2011a) posted disambiguation results for the set of RI records in Table 9.

Although the overall error rates are slightly increased from those shown in Table 6, it appears that our models still perform well at predicting matches among a random sample of OE inventors. In particular, these out-of-sample predictions yield splitting and lumping metrics similar to those shown in Table 5. Perhaps most importantly, Random Forests and Conditional FoRF still outperform the posted results from Lai et al (2009 and 2011a) by a wide margin based on a balance of

Table 9 Assessing Disambiguation Performance on Random Inventors

Disambiguation Method, Full USPTO Posted Results	Splitting (%)	Lumping (%)
Lai et al (2009) Posted Results, RI Subset	15.55	0.84
Lai et al (2011a) Posted Results, RI Subset	11.60	1.33
Disambiguation Method, Implemented on RI Subset	**Splitting (%)**	**Lumping (%)**
Lai et al (2009)	17.41	0.38
Random Forests	1.16	2.41
Conditional FoRF	1.52	2.34

low splitting and low lumping. The RI subset happens to be particularly difficult to disambiguate without yielding many false negative (splitting) errors. However, our classification algorithms have relatively stable performance when used to disambiguate this subset. Because of this, we can safely assume that with the proper training data, our methods should be able to accurately disambiguate records from any population or database, including the full USPTO database (disregarding computational issues).

Finally, there is a potential OE sample selection bias since all of our inventor records, including the prolific inventor groups and the sample of random inventors, are taken from the general OE subset of the full USPTO database. Note that while we cannot assess this potential bias directly as we did with the potential prolific inventor bias above, we can compare some simple descriptive statistics of inventor records in both populations, as shown in Table 1 (Section 3.2). We see that important quantities relevant to our comparison of two inventor records such as length of first and last names, proportions of missing middle names and assignees, number of co-inventors per patent, and proportions of US (and foreign) inventors are roughly the same in the OE subset as they are in the full USPTO population. At least in terms of these matching characteristics important in inventor disambiguation, the OE subset is similar to the full USPTO population. This similarity supports the notion that our models built using labeled OE records can be used to disambiguate non-OE inventor records.

5. Discussion

Disambiguation and record linkage algorithms within and outside statistics rarely build models or, in some cases, even evaluate results using labeled records. Instead, these algorithms often use decision rules, many of which use sets of heuristic weights and thresholds, to determine which records should be linked. In the field of technology, innovation, and entrepreneurship, even some of the most statistically sophisticated publically available disambiguation and record linkage algorithms (Lai et al 2009) fail to completely leverage recent model developments in statistics and statistical record linkage, such as the Fellegi-Sunter model for bipartite record linkage, probability models for determining matches, or supervised learning applied to disambiguation and record linkage. More

recent approaches have included using probability models and semi-supervised learning, but stop short of using labeled records during the disambiguation process (Torvik and Smalheiser 2009; Lai et al 2011a).

In this work, we leverage an unprecedented dataset of 98,762 labeled USPTO optoelectronics (OE) inventor records (where the labels are IDs indicating the true inventor). These records are obtained 824 inventor CVs collected for a study on economic downturns, inventor mobility, and technology trajectories in OE (Akinsanmi et al 2012). We (1) evaluate existing inventor disambiguation algorithms and (2) build and evaluate classification models for use in inventor disambiguation. In doing so, we find that the results of existing inventor disambiguation algorithms can suffer from systematic errors which may lead to the incorrect estimation of important metrics in TIE, such as inventor mobility. In particular, we show that the Lai et al (2009) algorithm may tend to overestimate the number of patents belonging to each prolific inventor (since the algorithm may be susceptible to false positive errors), leading to an overestimation of inventor mobility and incorrect lists of prolific inventors. Conversely, we show that the Fleming et al (2007) algorithm may tend to underestimate the number of patents belonging to each prolific inventor (since it may be susceptible to false negative errors), leading to an underestimation of inventor mobility and incomplete lists of prolific inventors.

To provide more accurate inventor disambiguation, we use our set of labeled OE inventor records to build classification models that predict whether pairs of unlabeled inventor records belong to the same inventor. We show that our approach can yield substantial improvements to the accuracy of disambiguation results in terms of a balance of both low splitting (rate of false negative errors) and low lumping (rate of false positive errors). In particular, the Conditional Forest of Random Forests (FoRF) classification model reduces splitting by 92.7% and lumping by 84.5% over the Lai et al (2009) algorithm. Additionally, although we have not yet extended Conditional FoRF to a disambiguation of the full USPTO database, we show some evidence that Conditional FoRF will likely outperform the Lai et al (2011a) disambiguation algorithm in our evaluation of Lai et al's posted disambiguation results.

Given these results, research using the disambiguation results of existing algorithms (including Lai et al 2009 and Lai et al 2011a) should be revisited to ensure that the errors inherent to these algorithms do not confound the subsequent conclusions or results. Additionally, future disambiguation and record linkage works in both TIE and statistics should consider the use of truly labeled records for building models during the disambiguation process, not just for evaluating results after disambiguation is complete. In statistics, we hope that this work emphasizes the potential value that supervised learning and, specifically, classification models can have in terms of improving the accuracy of record linkage or disambiguation results. Additionally, we hope that improvements in

disambiguation accuracy achieved when using a conditioning approach emphasize the power that conditioning can have in a disambiguation or record linkage context. Such algorithmic developments have applications not only in disambiguation and record linkage contexts where training data is available, but also in unsupervised and semi-supervised contexts. We recognize that in many record linkage and disambiguation applications, labeled records are often extremely expensive or, in some cases, impossible to attain. However, in applications where labeled records are available, such as numerous datasets in TIE, researchers should strongly advocate for their use during disambiguation.

6. Future Work

Our paper demonstrates the importance of transparent disambiguation methods in TIE research. We show that existing, publically available disambiguation algorithms and their results have systematic errors that could be expected to alter the outcomes of papers subsequently published using that data. We demonstrate that we are able to dramatically reduce these systematic errors using a new algorithm we develop. Going forward, it would be interesting to implement our algorithm on the full USPTO. To implement our algorithm on the full USPTO database, we would need to overcome two main challenges: (1) a potential bias towards OE inventors in our sample of inventor CVs, and (2) the potential computational challenges in applying our disambiguation approach to the full USPTO database.

Assessing the Influence of Optoelectronics on Disambiguation Performance: In Section 4.6, we show that our methods are successful at matching prolific and non-prolific inventors, eliminating the concern that our classification models are biased towards prolific inventors. However, all of our CV inventors come from the general OE population. Thus, although our algorithm will be effective in any setting with suitable training data (e.g. industry-specific datasets where researchers have extensive labeled data), it is possible that the classification models we trained using our labeled OE inventor records will be less effective at disambiguating USPTO datasets outside the OE industry. Future work should examine this potential bias by testing how well our classification models trained on labeled OE inventor data perform on labeled records from a set of non-OE inventors. We have identified three potential sources for additional labeled inventor records: (1) hand-disambiguated records corresponding to 95 US-based academic inventors (Lai et al 2011a), (2) hand-disambiguated records corresponding to Pierre Azoulay's work with prolific bio-tech inventors (Azoulay et al 2012), and (3) hand-disambiguated records corresponding to 675 chemical patents from DataVerse. Additional labeled inventor records from these alternative sources would be helpful for two reasons. First, they would broaden our sample to better represent the full USPTO database. Second, they would provide for testing models (on out-of-sample

records) trained on our current set of labeled records. If our classification models trained on labeled OE inventor records succeed at disambiguating records from these alternative sources, this would offer evidence that our models are generalizable to a disambiguation of the full USPTO database.

Scaling Disambiguation Algorithms to Different Databases: One issue with using FoRF methods (and classification models in general) is the substantial computation required. For FoRF to be applied to a very large database (such as the full USPTO OE patent database or the full USPTO patent database), we may need to implement computational and data storage reduction techniques. While some techniques such as pre-calculating all similarity scores to remove redundant calculations are already employed, they alone are not enough to guarantee computational feasibility. Additional techniques such as blocking to reduce the number of pairwise comparisons and parallelization of the out-of-sample predictions to reduce prediction run-time will likely be necessary to allow our methods to be used feasibly for full USPTO disambiguation of about 10 million inventor records. The central idea behind blocking is to prevent the algorithm from comparing two records that are obviously not a match, such as a pair of records with the last names "Ventura" and "Fuchs," without increasing the false positive and false negative error rates. To explore this technique, we ran some preliminary experiments and find that the use of basic parallelization techniques on only four computer processing cores can reduce the prediction time for Conditional FoRF by about 65%.

Acknowledgments

We would like to thank first and foremost, Eyiwunmi Akinsanmi, for spearheading the collection of the 824 USPTO optoelectronics inventor resumes, and allowing us to use her raw data. 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 Eyiwunmi Akinsanmi and 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; 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.

Many thanks to Lee Fleming for his early encouragement, and he and his groups extraordinary openness sharing insights from their own efforts. Thanks also to Matthew Marx for his very helpful feedback as a discussant at the 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, Mauricio Sadinle, and other members of the Carnegie Mellon University Department of Statistics Census research group; attendees of

the 2012 Classification Society Annual Meeting; and Kenneth Huang and the other participants in the 2012 Academy of Management Professional Development Workshop Measuring Knowledge Flows: Patent and Non-Patent Data 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).

References

Akinsanmi, E., R. Reagans, E. Fuchs. 2012. Economic Downturns, Technology Trajectories, and the Careers of Scientists. *Carnegie Mellon University Working Paper*.

Azoulay, P., W. Ding, T. Stuart. 2009. The Effect of Academic Patenting on the Rate, Quality, and Direction of (Public) Research Output. *Journal of Industrial Economics*. **57**(4) 637–623.

Azoulay, P., J.S.G. Zivin, B.N. Sampat. 2012. The Diffusion of Scientific Knowledge across Time and Space: Evidence from Professional Transitions for the Superstars of Medicine. *University of Chicago Press*.

Bessen, J. 2007. Programmer Documentation on PTO assignee: Compustat matching.

Bessen, J. 2007. NBER PDP Project – User Documentation: Matching Patent Data to Compustat Firms.

Bilenko, M., R. Mooney, W. Cohen, P. Ravikumar, S. Fienberg. 2003. Adaptive Name Matching in Information Integration. *IEEE Intelligent Systems*. **18**(50) 16–23.

Bound, J., C. Cummins, Z. Griliches, B.H. Hall, A.B. Jaffe. 1984. R & D, Patents, and Productivity. *University of Chicago Press*. Chapter Who Does R&D and Who Patents? http://www.nber.org/chapters/c10043

Breiman, L. 2001. Random Forests. *Machine Learning*. **45**(1) 5-32.

Carayol, N. and L. Cassi. 2009. Whos Who in Patents: A Bayesian approach.

Czarnitzki, D., B.H. Hall, R. Oriani. 2005. The Market Value of Knowledge Assets in U.S. and European Firms. *The Management Of Intellectual Property*.

Fleming, L., C. King III, A. Juda. 2007. Small World and Regional Innovation. *Organizational Science*. **18**(6).

Fuchs, E. 2012. On the relationship between manufacturing and innovation: Why not all techniques are created equal. *Carnegie Mellon University Working Paper*.

Hall, B. 2000. Innovation and Market Value, Productivity, Innovation and Economic Performance. *Cambridge: Cambridge University Press*.

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

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

Hall, B., K. Vopel. 1997. Market Value, Market Share, and Innovation. *NBER, the University of California at Berkeley, and the University of Mannheim.*

Han, H., L. Giles, H. Zha, C. Li, K. Tsioutsiouliklis. 2004. Two Supervised Learning Approaches for Name Disambiguation in Author Citations. J*oint Conference on Digital Libraries 2004.*

Hastie, T., R. Tibshirani, J. Friedman. 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. *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. *Journal of the American Statistical Association.* **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.

Kelley, R.P. 1984. Blocking Considerations for Record Linkage Under Conditions of Uncertainty. *Proceedings of the Social Statistics Section, American Statistical Association.* 602–605.

Lai, R., A. D'Amour, L. Fleming. 2009. The careers and co-authorship networks of U.S. patent-holders, since 1975.

Lai, R., A. D'Amour, A. Yu, Y. Sun, L. Fleming. 2011. Disambiguation and Co-authorship Networks of the U.S. Patent Inventor Database.

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

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

Lissoni, F., B. Sanditov, G. Tarasconi. 2006. The Keins Database on Academic Inventors: Methodology and Contents. *Cespri - Universit'a Bocconi.* Working paper 181.

Newcombe, H.B., M.E. Smith. 1975. Methods for Computer Linkage of Hospital AdmissionSeparation Records into Cumulative Health Histories. *Methods of Information in Medicine.* **14**(3) 118–125.

Sadinle, M. and S.E. Fienberg. 2013. A Generalized Fellegi–Sunter Framework for Multiple Record Linkage With Application to Homicide Record–Systems. *Journal of the American Statistical Association.* http://amstat.tandfonline.com/doi/pdf/10.1080/01621459.2012.757231

Singh, J. 2005. Collaborative Networks as Determinants of Knowledge Diffusion Patterns. *Management Science.* **51**(5): 756-770.

Treeratpituk, P. and C.L. Giles. 2009. Disambiguating Authors in Academic Publications using Random Forests. *Joint Conference on Digital Libaries 2009.*

Torvik, V., N. Smalheiser. 2009. Author Name Disambiguation in MEDLINE. *ACM Transactions on Knowledge Discovery from Data.* **3**(3), Article 11.

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

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

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

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

Winkler, W.E. 1989. Near Automatic Weight Computation in the Fellegi-Sunter Model of Record Linkage. *Proceedings of the Fifth Census Bureau Annual Research Conference.* 145–155.

Winkler, W.E. 1989. Methods for Adjusting for Lack of Independence in an Application of the Fellegi-Sunter Model of Record Linkage. *Survey Methodology.* **15** 101–117.

Winkler, W.E. 1990. String Comparator Metrics and Enhanced Decision Rules in the Felligi-Sunter Model of Record Linkage.

Winkler, W.E. 1995. Matching and Record Linkage. in B. G. Cox et al. *Business Survey Methods, New York: J. Wiley.* 355–384.

Yang, C., R. Nugent, E. Fuchs. 2012. Gains from Others Losses: Technology Trajectories and the Global Division of Firms. *Carnegie Mellon University Working Paper.*

Zucker, L., M. Darby, J. Fong. 2011. Communitywide Database Designs for Tracking Innovation Impact: COMETS, STARS, and Nanobank. *National Bureau of Economic Research.* Working Paper No. 17404.

Appendix A: Appendix: Optoelectronics Classes and Subclasses

Class ID/Subclass ID

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

* denotes all subclasses within this class - denotes a range of subclasses within this class

Definitions for these classes and subclasses can be found at:

http://www.uspto.gov/web/patents/classification/

Appendix B: Appendix: 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).

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 which indicates how closely two records match. For the purposes of this paper, we define all of these "similarity scores" as follows:

s_{ijk} represents the similarity score of inventor records i and j according to field k, where

$$i, j \in \{1, 2, ..., 453974\}$$

$$k \in \{1, 2, ..., K\}$$

$$K = \text{number of unique fields being compared}$$

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 inventor name disambiguation (Lai et al 2009), and it should be noted that changing these similarity scores could affect the disambiguation results.

B.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 result is a number (called the JW 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).

Using this method, similarity scores for long strings have the following properties. Given two long strings X_{ik} and X_{jk} for inventors i and j and field k:

$$s_{ijk} \in [0, 1]$$

$$s_{ijk} = 1 \text{ if } X_{ik} = X_{jk}$$

$$s_{ijk} = 0 \text{ if none of the characters in } X_{ik} \text{ are also in } X_{jk}$$

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; 2011a). Other string comparison metrics, such as the Jaccard distance, Soundex, TF/IDF, etc may have advantages over Jaro-Winkler in certain situations or for certain fields. For this research and comparison purposes, we only use Jaro-Winkler string comparisons. However, note that our method is flexible and does not depend on a specific type of string comparison.

B.2. Short Text Strings: State, Country, and 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:

$$s_{ijk} \in 0, 1$$
$$s_{ijk} = 1 \text{ if } X_{ik} = X_{jk}$$
$$s_{ijk} = 0 \text{ if } X_{ik} \neq X_{jk}$$

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 "MA" vs. "MN."

Finally, note that the Lai et al (2009) algorithm also uses exact matching for short strings.

B.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 that particular inventor record's patent. There are several different ways to quantify the similarity of two lists of co-inventors or class-subclass pairs, and no one method is necessarily "best." For the purposes of this paper, we use the following similarity scores when comparing lists of co-inventors or class-subclass pairs, motivated by the work of Lai et al (2009 & 2011a).

Given two lists X_{ik} and X_{jk} for inventors i and j and field k:

$$s_{ijk} \in 0, 1$$
$$s_{ijk} = 1 \text{ if } X_{ik} \cap X_{jk} \neq \emptyset$$
$$s_{ijk} = 0 \text{ if } X_{ik} \cap X_{jk} = \emptyset$$

That is, list similarity scores check if there are any shared elements across the two lists. Again, note that other list similarity scores could be substituted here.

Appendix C: Appendix: Random Forests

Brieman (2001) includes a detailed description of the original Random Forests classification method. For descriptions of the variants of Random Forests which we designed disambiguation, see below.

C.1. Forest of Random Forests with Random Subsets

Below is the algorithm for calculating a classifier using the "Forest of Random Forests with Random Subsets" method:

1. Split the pairwise comparison data into multiple groups by selecting each group randomly – i.e, split the full pairwise comparison dataset into F random subsets (random forest of random forests)

2. Train a random forest on the pairwise comparisons in each group – F random forests in total

3. Save all random forests (i.e. the forest of random forests) for use in predicting the unlabeled pairwise comparisons of inventor records (described in Section 3.3.3)

C.2. Conditional Forest of Random Forests

Below is the algorithm for calculating a classifier using the "Conditional Forest of Random Forests" method:

1. Split the pairwise comparison data into multiple groups based on any of the following reasons:
(Note: If any subgroup is too large, we can use the random subsets forest of random forests to make it computationally feasible)

(a) Define each group by conditioning on some feature of the underlying inventor records used in the comparison (e.g. inventor country, technology class, etc)

(b) Define each group by conditioning on some feature of the underlying similarity scores (e.g. by different missingness categories, different lengths of strings, frequencies / commonness of last names, etc)

(c) Define each group by conditioning on some known feature of the labeled pairwise comparisons that would be unknown in the unlabeled pairwise comparisons (e.g. number of patents that inventor has, whether or not that inventor is mobile, etc.)

2. Train a random forest on the pairwise comparisons in each group

3. Save all random forests (i.e. the forest of random forests) for use in predicting the unlabeled pairwise comparisons of inventor records (described in Section 3.3.3)