Practical Performance Evaluation of Entity Resolution Algorithms: Lessons Learned at PatentsView.org

Olivier Binette[1,2], Sokhna A York[2], Emma Hickerson[2], Youngsoo Baek[1], Sarvo Madhavan[2], and Christina Jones[2]

[1]Duke University
[2]American Institutes for Research

August 7, 2022

Abstract

This paper introduces a novel evaluation methodology for entity resolution algorithms. It is motivated by PatentsView.org, a U.S. Patents and Trademarks Office patent data exploration tool that disambiguates patent inventors using an entity resolution algorithm. We provide a data collection methodology and tailored performance estimators that account for sampling biases. Our approach is simple, practical and principled – key characteristics that allow us to paint the first representative picture of PatentsView's disambiguation performance. This approach is used to inform PatentsView's users of the reliability of the data and to allow the comparison of competing disambiguation algorithms.

1 Introduction

Entity resolution (also called record linkage, deduplication, or disambiguation) is the task of identifying records in a database that refer to the same entity. An entity may be a person, a company, an object or an event. Records are assumed to contain partially identifying information about these entities. When there is no unique identifier (such as a social security number) available for all records, entity resolution becomes a complex problem which requires sophisticated algorithmic solutions (Herzog et al., 2007; Christen, 2012; Dong and Srivastava, 2015; Ilyas and Chu, 2019; Christophides et al., 2021; Christen, 2019; Papadakis et al., 2021; Binette and Steorts, 2022).

For instance, the U.S. Patents and Trademarks Office (USPTO) makes available patent data dating back to 1790 (digitized full-text data is available from 1976). However, there is no standard for uniquely identifying inventors on patent applications. The result is a set of ambiguous mentions of inventors, where a single person's name may be spelled in different ways on two applications and where two different inventors with the same name may be difficult to distinguish. Inventor mobility further complicates the use of contextual information for disambiguation. Much research has been done within the fields of economics, computer science and statistics to disambiguate inventor mentions in this situation (Trajtenberg and Shiff, 2008; Ferreira et al., 2012; Ventura et al., 2013; Li et al., 2014; Ventura et al., 2015; Kim et al., 2016; Yang et al., 2017; Morrison et al., 2017; Müller, 2017; Traylor et al., 2017; Balsmeier et al., 2018; Tam et al., 2019; Monath et al., 2019; Doherr, 2021).

The Office of the Chief Economist at the USPTO and the American Institutes for Research (AIR) currently maintain **PatentsView.org**, a patents knowledge discovery service that provides disambiguated patents data based on statistical entity resolution (Toole et al., 2021). Our paper is motivated by this disambiguation service. Specifically, we are interested in the problem of *evaluating* the accuracy of PatentsView's disambiguation, for the purpose of informing users of the reliability of the data and in order to support methodological research to improve upon PatentsView's disambiguation algorithm. We expand on the evaluation problem below.

1.1 The Evaluation Problem

The entity resolution evaluation problem is to extrapolate from observed performance in small samples to real performance in a database with millions of records. Wang et al. (2022) refer to this as bridging the reality-ideality gap in entity resolution, where high performance on benchmark datasets often does not translate into the real world. Here, performance may be defined as any combination of commonly used evaluation metrics for entity resolution, such as precision and recall, cluster homogeneity and completeness, rand index, or generalized merge distance (Maidasani et al., 2012). These metrics can be computed on benchmark datasets for which we have a ground truth disambiguation. However, the key evaluation problem is to obtain estimates that are representative of performance on the full data, for which no ground truth disambiguation is available. This is challenging for the following reasons.

First, entity resolution problems do not scale linearly. While it may be easy to disambiguate a small dataset, the opportunity for errors grows *quadratically* in the number of records. As such, we may observe good performance of an algorithm on a small benchmark dataset, while the true performance on the entire dataset may be something else entirely.[1] This is a problem that PatentsView.org currently faces. Despite encouraging performance evaluation metrics on benchmark datasets, with nearly perfect precision and recall reported in the latest methodological report (Monath et al., 2021), the data science team at AIR observes lower real-world accuracy. This phenomenon is illustrated in example 1 below.

A second problem is large class imbalance in entity resolution (Marchant and Rubinstein, 2017). Viewing entity resolution as a classification problem, the task is to classify record pairs as being a match or non-match. However, among all pairs of records, only a small fraction (usually much less than a fraction of a percent) refer to the same entity. The vast majority of record pairs are not a match. This makes it difficult to evaluate performance through random sampling of record pairs.

A third problem is the multiplicity of sampling mechanisms used to obtain benchmark datasets. To construct hand-disambiguated datasets, blocks, entity clusters, or predicted clusters may be sampled with various probability weights. These sampling approaches must be accounted for in order to obtain representative performance estimates (Fuller, 2011).

Our approach, detailed in sections 1.1.3 and 2.3, addresses these challenges by putting forward novel cluster-based expressions for performance metrics that reflect various sampling schemes. Each of these representations immediately suggests simple estimators that properly account for the above issues.

Example 1 (Bias of precision computed on benchmark datasets)**.** To exemplify the problem with the trivial use of performance evaluation metrics on benchmark datasets, we carried out a simple experiment that is described in detail in appendix A.1. In short, we evaluated a disambiguation algorithm by sampling ground truth clusters and computing pairwise precision[2] on this set of sampled clusters. This is analogous to the

[1] This particular effect of dataset size in entity resolution is explored in Draisbach and Naumann (2013) in the context of choosing similarity thresholds.

[2] Pairwise precision is the probability that a pair of record predicted to be a match is indeed a match. See section 2.3 for a formal definition

way that many real-world benchmark datasets are obtained and typically used. In this experiment, we know that the disambiguation algorithm has a precision of 52% for the entire dataset.

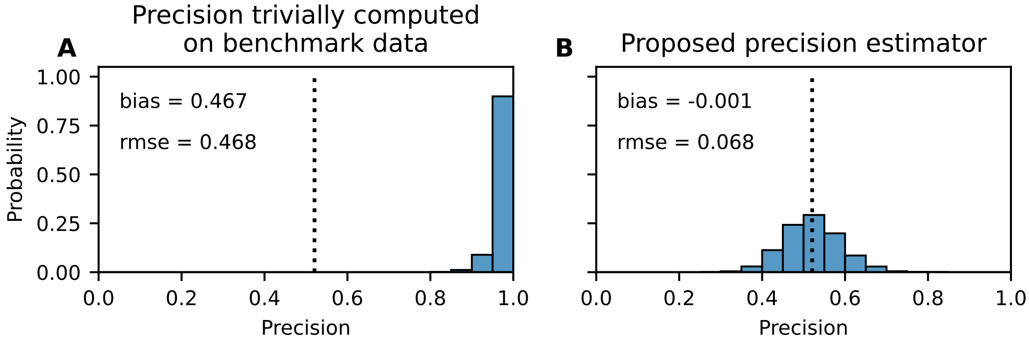


Figure 1: *Distribution of precision estimates versus the true precision of* 52% *(shown as a dotted vertical line). Panel* ***A*** *shows the trivial precision estimates computed for sampled records. Panel* ***B*** *shows our proposed precision estimates which accounts for the sampling mechanism. Sample bias and root mean squared error (rmse) are reported in each figure.*

In panel **A** of figure 1, we see the distribution of precision estimates versus the true precision of 52% shown as a dotted vertical line. Precision estimates are usually very close to 100% and always higher than 80%, despite the truth being a precision of only 52%. In contrast, panel **B** shows the distribution of our proposed precision estimator which is nearly unbiased. Both precision estimators rely on exactly the same data. They only differ in how they account for the underlying sampling process and the extrapolation from small benchmark datasets to the full data.

1.1.1 Why Bother With Evaluation?

There are two main uses for accurate and statistically rigorous evaluation methodology.

The first is model selection and comparison. PatentsView.org continually works at improving disambiguation methodology. This requires choosing between alternative methods and evaluating the results of methodological experiments. Without sound evaluation methodology, decisions regarding the disambiguation algorithm may not align with real-world use and real-world performance. Notably, for a performance metric such as the pairwise f-score, one algorithm may perform better than another on a small benchmark dataset, while the opposite may hold true for performance on the entire data. This problem arises with typical benchmark datasets obtained from randomly sampling blocks or randomly sampling clusters (see section 2.3 for a definition of different sampling mechanisms).

The second is adequate use of disambiguated data. PatentsView.org's disambiguation results have been used in numerous scientific studies (Toole et al., 2021). These studies make assumptions about the reliability of the data that need to be validated and upheld. In short, users of disambiguated data need to understand its reliability in order to make scientifically appropriate use of it. Evaluation aims to provide this rigorous reliability information.

1.1.2 Past Work

Much of the past literature has focused on defining and using relevant clustering evaluation metrics. The topic of estimating performance from samples has received much less attention, usually focusing on importance sampling estimators based on record pairs. We review the contributions to these two main topics below.

Metrics Pairwise precision and recall metrics were first reported in Newcombe et al. (1959), with Bilenko and Mooney (2003) and Christen and Goiser (2007) emphasizing the importance of precision-recall curves for algorithm evaluation. However, there are issues with the use of pairwise precision and recall in entity resolution applications, such as the large relative importance of large clusters. As such, other clustering metrics have been proposed, including cluster precision and recall, cluster homogeneity and completeness, and generalized merge distances (Michelson and Macskassy, 2009; Menestrina et al., 2010; Maidasani et al., 2012; Barnes, 2015). Important practical issues regarding the use of aggregate metrics are discussed in Hand and Christen (2018).

While our work focuses on estimating pairwise precision and recall, the general approach also applies to any cluster-based performance evaluation metrics, such as those described in Michelson and Macskassy (2009).

Estimation Regarding the reliable estimation of performance metrics, Belin and Rubin (1995) first proposed a semi-supervised approach to calibrating error rates when using a Fellegi-Sunter model (Fellegi and Sunter, 1969). Marchant and Rubinstein (2017) proposed an adaptive importance sampling estimator to estimate precision and recall from sampled record pairs. Other approaches to the estimation of performance metrics are model based, where estimated precision and recall can be obtained from predicted match probabilities between record pairs (Enamorado et al., 2019). However, these model-based approaches cannot be used when working with black-box machine learning models or ad hoc clustering algorithms.

In contrast, our approach to estimation is more practical than pairwise sampling and applies to any black-box disambiguation algorithms such as those used at PatentsView.

1.1.3 Our Approach

Our approach to estimating performance metrics is based on the use of benchmark datasets that already exist or that can be collected in a cost-effective way. These datasets contain entity clusters corresponding to either: (a) sampling records and recovering all associated instances, (b) directly sampling clusters, or (c) sampling blocks. For each of these sampling processes, we propose estimators that correct for the issues discussed in section 1.1, are nearly unbiased, and are easy to use in practice.

Our approach has the following advantages:

1. It can leverage existing benchmark datasets as well as new datasets collected specifically for performance evaluation.

2. It can easily be generalized to estimate other clustering metrics, such as cluster precision, cluster recall, cluster homogeneity and completeness, and other generalized merge distances.

3. For evaluation, the review of entity clusters is much more efficient than the review of record pairs. We can achieve high accuracy with small samples without relying on sophisticated sampling schemes.

Furthermore, our approach is novel. To our knowledge, we are the first to propose unbiased performance estimators based on cluster and block samples. Past work either ignored biases when computing precision and recall from benchmark datasets (Frisoli and Nugent, 2018; Monath et al., 2021; Han et al., 2019), did not provide estimates for precision or recall (McVeigh et al., 2019), or provided solutions tailored to very specific record linkage models (Belin and Rubin, 1995). We provide the first general solution to entity resolution evaluation that does not rely on sampling record pairs and that applies to any disambiguation algorithm.

In short, the proposed approach is simple, principled, and practical. It is simple to use, it is statistically principled in its account of sampling processes and uncertainty, and it is practical in the way that it can provide cost-effective estimates for any disambiguation algorithm.

1.2 Structure of the Paper

The rest of the paper is organized as follows. In section 2, we describe benchmark datasets, our hand-disambiguation methodology for evaluation, the proposed estimators, and our simulation study that we use to validate the performance of our estimators. Section 3 then presents our performance estimates and results from the simulation study. Section 4 summarizes the paper and explores future research directions.

2 Data and Methodology

In this section, we introduce the benchmark datasets used at PatentsView, our hand-disambiguation methodology, our proposed performance metric estimators, and the simulation framework that we use to compare estimators. Note that we focus on *inventor* disambiguation throughout, rather than on the related problems of assignee and location disambiguation.

2.1 Benchmark Datasets for Inventor Disambiguation

We consider the following benchmark datasets for inventor disambiguation.

Israeli Inventors Benchmark Trajtenberg and Shiff (2008) disambiguated the U.S. patents of Israeli inventors that were granted between 1963 and 1999. A total of 6,023 Israeli inventors were identified for this time period with 15,310 associated patents.

Li et al. (2014)'s Inventors Benchmark Based on an original dataset from Gu et al. (2008), Li et al. (2014) disambiguated the patent history (between 1975 and 2010) of 95 U.S. inventors.

2.2 Hand-Disambiguation Methodology

In addition to considering the above benchmark datasets, we have carried out hand-disambiguation of inventor mentions. This was motivated by the evaluation of the current PatentsView inventor disambiguation using the estimators proposed in section 2.3.

In total, 100 inventors were sampled with probability proportional to their number of granted patents. This was done by sampling inventor mentions uniformly at random and recovering all patents for a given inventor. These inventor mentions were from U.S. patents granted between 1976 and December 31, 2021.

Two AIR staff were tasked with recovering inventors' patents given sampled inventor mentions. First, given a sampled inventor, the associated predicted cluster was reviewed and any wrongly assigned patents were removed. Next, PatentsView's search tools were used to find additional mentions of similarly named inventors. These inventor mentions were reviewed and added to the predicted cluster, if appropriate. The two AIR staff had an initial training session, followed by a test run on 10 inventors, before carrying out the rest of the data collection. They worked independently, which resulted in two datasets being obtained for the same inventor mentions. In section 3, these are referred to as the **Staff 1** and **Staff 2** datasets.

Note that our data collection methodology is biased toward PantentsView's current disambiguation. Indeed, we did not expect the staff to have found all errors or all missing inventor mentions from the predicted clusters. The staff used their best judgment, supported by a thorough search, to resolve inventor mentions. In cases where no errors were found, the current disambiguation was assumed to be correct. Performance estimates based on this data might therefore be slightly optimistic, which should be acknowledged when reporting performance estimates to PatentsView.org users. Otherwise, for the purpose of improving the current disambiguation algorithm, this data is still appropriate to use. It represents the most visible errors in the current disambiguation rather than the totality of them.

2.3 Proposed Performance Estimators

Throughout the rest of the paper, we focus on pairwise precision and pairwise recall (defined below in (1)) as our performance evaluation metrics.

2.3.1 Representation Lemmas

First, we define pairwise precision and recall in terms of the number of links between records. Let $\mathcal{D} = \{1, 2, 3, \ldots, N\}$ index a set of records let $\mathcal{C}$ be the partition of $\mathcal{D}$ representing ground truth clustering, and let $\widehat{\mathcal{C}}$ be a set of predicted clusters. Now let $\mathcal{T}$ be the set of record pairs that appear in the same cluster in $\mathcal{C}$ (matching pairs), and let $\mathcal{P}$ be the set of record pairs that appear in the same predicted cluster in $\widehat{\mathcal{C}}$ (predicted links). Pairwise precision (P) and pairwise recall (R) are then defined as

$$P = \frac{|\mathcal{T} \cap \mathcal{P}|}{|\mathcal{P}|}, \quad R = \frac{|\mathcal{T} \cap \mathcal{P}|}{|\mathcal{T}|}. \tag{1}$$

Note that $P = R\,|\mathcal{T}|/|\mathcal{P}|$. As such, precision and recall are equal if and only if the right number of matching pairs is predicted under $\widehat{\mathcal{C}}$.

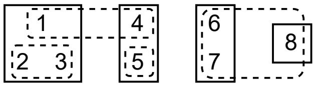


Figure 2: *Example of ground truth clustering $\mathcal{C}$ (represented by boxes with a full border) and predicted clustering $\widehat{\mathcal{C}}$ (rounded boxes with a dotted border) of elements $\mathcal{D} = \{1, 2, \ldots, 8\}$. Here $\mathcal{T} = \{(1,2),(2,3),(1,3),(4,5),(6,7)\}$ and $\mathcal{P} = \{(1,4),(2,3),(6,7),(7,8),(6,8)\}$. As such, $P = R = 2/5$ in this example.*

We now provide three alternative representations of precision and recall that correspond to the processes of sampling records, sampling true clusters, and sampling blocks. These representations will be used to obtain precision and recall estimators under these sampling processes.

Record Sampling Representation For a given $i \in \mathcal{D}$, let $c(i) \in \mathcal{C}$ be the ground truth cluster associated with i in $\mathcal{C}$. For a given $c \in \mathcal{C}$, we define

$$f(c, \widehat{\mathcal{C}}) = \sum_{\hat{c} \in \hat{\mathcal{C}}} \binom{|c \cap \hat{c}|}{2}, \quad g(c, \widehat{\mathcal{C}}) = \frac{f(c, \widehat{\mathcal{C}})}{\sum_{\hat{c} \in \hat{\mathcal{C}}} \binom{|\hat{c}|}{2}}. \tag{2}$$

Lemma 1. *If i is distributed over $\mathcal{D}$ with probabilities $p_i > 0$, then*

$$P = \mathbb{E}\left[\frac{g(c(i), \widehat{\mathcal{C}})}{p_i |c(i)|}\right], \quad R = 2\frac{\mathbb{E}\left[f(c(i), \widehat{\mathcal{C}})/(|c(i)|p_i)\right]}{\mathbb{E}\left[(|c(i)| - 1)/p_i\right]}. \tag{3}$$

Cluster Sampling Representation In the cluster sampling case, sampling probabilities are typically known only up to a normalizing factor. This is because the total number of true clusters and other aspects of the ground truth cluster distribution are unknown in practice. As such, we provide expressions for precision and recall that only require knowing the sampling probabilities up to a normalizing factor. This allows the consideration of sampling uniformly at random and sampling clusters with probability proportional to their size.

Lemma 2. *If c is distributed over $\mathcal{C}$ with probabilities proportional to $p_c > 0$, then*

$$P = \frac{N\,\mathbb{E}\left[g(c, \widehat{\mathcal{C}})/p_c\right]}{\mathbb{E}\left[|c|/p_c\right]}, \quad R = \frac{\mathbb{E}\left[f(c, \widehat{\mathcal{C}})/p_c\right]}{\mathbb{E}\left[\binom{|c|}{2}/p_c\right]}. \tag{4}$$

Note that, with clusters sampled with probability proportional to their size, the expression for precision simplifies to $P = N\,\mathbb{E}\left[g(c, \widehat{\mathcal{C}})/|c|\right]$.

Remark 1. Lemma 1 and lemma 2 can be generalized to apply to any performance metric that can be expressed as a sum $\sum_{c\in\mathcal{C}} h(c, \widehat{\mathcal{C}})$, for some function h, or as a function of such sums. For instance, the use of the so-called *cluster* precision and *cluster* recall (Barnes, 2015), or of cluster homogeneity and completeness (Barnes, 2015), can be more appropriate in the presence of large clusters. We leave these generalizations as extensions of our work.

Disjoint Block Sampling Representation Let $\mathcal{B}$ be a partition of $\mathcal{D}$ such that for every $c \in \mathcal{C}$, there exists $b \in \mathcal{B}$ with $c \subset b$. For a given $b \in \mathcal{B}$, let $\mathcal{T}_b$ be the set of ground truth links contained within b, let $\mathcal{P}_b$ be the set of predicted links contained within b, and let $\mathcal{P}_b^-$ be the set of predicted links with a single record in b (i.e., $\mathcal{P}_b^-$ is the set of outgoing links from b).

Lemma 3. *If b is distributed over $\mathcal{B}$ with probabilities proportional to $p_b > 0$, then*

$$P = \frac{\mathbb{E}\left[|\mathcal{T}_b \cap \mathcal{P}_b|/p_b\right]}{\mathbb{E}\left[(|\mathcal{P}_b| + \frac{1}{2}|\mathcal{P}_b^-|)/p_b\right]}, \quad R = \frac{\mathbb{E}\left[|\mathcal{T}_b \cap \mathcal{P}_b|/p_b\right]}{\mathbb{E}\left[|\mathcal{T}_b|/p_b\right]}. \tag{5}$$

Remark 2. Lemmas 1 − 3 are formulated in terms of sampled ground truth clusters and sampled blocks that do not contain errors. However, given the duality between precision and recall (interchanging the roles between $\mathcal{C}$ and $\widehat{\mathcal{C}}$ interchanges precision and recall), the results also apply to sampling *predicted* clusters.

2.3.2 Proposed Estimators

All of the expressions for precision and recall in lemmas 1 − 3 are either population means or ratios of population means. As such, they can be estimated using sample means and ratios of sample means. For readability, we present here a generic formula for an approximately unbiased estimator of the ratio of means and then specify the needed quantities for each representation below. The estimator applies a first order bias correction to the ratio of sample means, based on a Taylor approximation. Approximate confidence intervals can be computed based on the variance estimator of the ratio of sample means by Taylor approximation, assuming the corrected estimator has a small bias (Fuller, 2011). The generic formula for estimating the ratio of T-sized "population" (of records/clusters/blocks) means of the form

$$E = \frac{1/T \sum_{i=1}^{T} B_i}{1/T \sum_{i=1}^{T} A_i}, \tag{6}$$

assuming we have sampled n elements (records/clusters/blocks), is

$$\widehat{E} = \frac{\bar{B}_n}{\bar{A}_n}\left\{1 + \frac{\theta_{n,T}}{n(n-1)}\sum_{s=1}^{n}\frac{A_s}{\bar{A}_n}\left(\frac{B_s}{\bar{B}_n} - \frac{A_s}{\bar{A}_n}\right)\right\},\ \bar{A}_n = \frac{1}{n}\sum_{s=1}^{n} A_s,\ \bar{B}_n = \frac{1}{n}\sum_{s=1}^{n} B_s \tag{7}$$

We note that an additional symbol $\theta_{n,T}$ is introduced for a possible finite population correction when relatively large number of elements are sampled without replacement (see below). Classical adjustment is set to $\theta_{n,T} = (1 - \frac{n-1}{T-1})$ (Cochran, 1977). For practical purposes when T is large, $\theta_{n,T} = 1$ will suffice. In fact, knowledge of T is not needed at all as long as it is large enough relative to n, which is useful because the total number of true clusters/blocks is not known in advance. Confidence intervals can be computed based on the variance estimate of the above, which is

$$\widehat{V}(\widehat{E}) = \left(\frac{\bar{B}_n}{\bar{A}_n}\right)^2 \frac{\theta_{n,T}}{n(n-1)}\sum_{s=1}^{n}\left(\frac{A_s}{\bar{A}_n} - \frac{B_s}{\bar{B}_n}\right)^2. \tag{8}$$

The specific values for A_s, B_s, and T in each of our representation are described in appendix A.2.

Remark 3. In entity resolution applications, we can typically assume that elements have been sampled with replacement (or closely so). Indeed, with a small proportion of sampled elements, nonreplacement samples are approximately equivalent to samples with replacement. However, if dealing with relatively large nonreplacement samples, then the sampling probabilities used in the definition of the estimators should be adjusted to reflect the size-dependent effect of nonreplacement (Horvitz and Thompson, 1952).

2.4 Simulation Study

In order to assess the performance of the proposed estimators, we carried out a simulation study based on PatentsView's inventor disambiguation. Specifically, in the context of the simulation, we considered PatentsView current inventor disambiguation as the ground truth clustering. A simulated set of predicted clusters was obtained by introducing errors (misattribution of inventor mentions) into the current disambiguation. We then estimated the precision and recall of this predicted clustering using our estimators based on random cluster samples. The process of sampling clusters and estimating precision/recall was repeated 100 times in order to provide the distribution of the estimators and metrics such as bias and root mean squared error (rmse).

To introduce errors, we picked records at random and changed their cluster assignment to that of other records picked at random. This is a simple process that ensures that larger clusters are more likely to contain errors. In our simulation, we considered rates of 5%, 10%, and 25% for the proportion of records that are sampled for cluster misassignment. Although the larger error rates are more realistic, the 5% misattribution rate helps showcase the properties of our estimators when only a small proportion of the sampled clusters is associated with errors.

For the sampling process, we considered sampling records uniformly at random and recovering their associated clusters. This is the same as sampling clusters with probability proportional to their size. In the record/cluster sampling cases, we looked at the effect of sampling 100, 200 and 400 records/clusters.

Finally, we compared the following three precision and recall estimators:

`P_naive`, `R_naive` This is the "naive" precision (respectively recall) estimator obtained by computing precision (respectively recall) when only looking at records that appear in the sampled clusters.

`P_record`, `R_record` These are the precision and recall estimators corresponding to uniformly sampling records in lemma 1 ($p_i \propto 1$) with the bias adjustment given in (7). Note that these are the same as

Table 1: *Bias and root mean squared error (rmse) of precision estimators for the simulation study described in section 2.4. The* `rate` *variable represents the percentage misattribution rate.*

	rate	5			10			20		
	sample size	100	200	400	100	200	400	100	200	400
bias	P_cluster_block	-0.004	-0.002	-0.001	-0.002	-0.001	0.001	-0.004	-0.001	-0.000
	P_naive	0.099	0.099	0.099	0.195	0.195	0.195	0.366	0.366	0.365
	P_record	-0.002	0.013	0.009	-0.001	0.011	0.008	-0.001	0.009	0.006
rmse	P_cluster_block	0.045	0.033	0.021	0.041	0.038	0.026	0.063	0.052	0.034
	P_naive	0.099	0.099	0.099	0.195	0.195	0.195	0.366	0.366	0.365
	P_record	0.294	0.232	0.169	0.260	0.205	0.150	0.207	0.162	0.117

the estimators obtained from lemma 2 when sampling clusters with probability proportional to their size ($p_c \propto |c|$).

`P_cluster_block` This is the precision estimator obtained by considering each sampled cluster as its own block in lemma 3, where clusters have been sampled with probability proportional to their size ($p_b \propto |b|$) and with the bias adjustment given in (7). Note that in the case of recall with cluster blocks, the estimator corresponding to lemma 3 is the same as the one corresponding to lemma 2 and lemma 1.

3 Results

3.1 Results From the Simulation Study

Figure 3 shows the distribution of the three precision estimators used in the simulation study (see section 2.4) compared to ground truth precision. The block sampling estimator `P_cluster_block` is highly accurate, while `P_record` is more variable and `P_naive` is entirely uninformative. Note that `P_record` can take values greater than 1 (not shown in this figure), so that truncating it to be less than 1 introduces a bias in some cases.

`P_cluster_block` performs better than `P_record` because `P_record` has been derived in a generic way that applies to any performance metric that can be expressed in cluster form similar to (4). On the other hand, `P_cluster_block` relies on specific properties of precision. Among other things, this ensures that `P_cluster_block` is constrained to be between 0 and 1. In practice, `P_cluster_block` should be preferred as a pairwise precision estimator. The bias and rmse of the precision estimators are reported in table ??.

Regarding recall, figure 4 shows the distribution of the two estimators used in the simulation study. The naive recall estimator performs well in this case. However, the recall estimator accounting for the sampling mechanism is more accurate. The bias and rmse of the recall estimators are reported in table ??.

3.2 Evaluation of PatentsView's Disambiguation

Table 3 shows estimated pairwise precision and recall from benchmark datasets and from our two hand-curated datasets. Note that each estimate is associated with a given population of inventor mentions which is a subset of granted U.S. patents since 1976. We focused on U.S. patents granted since 1976 as this is the main data product of PatentsView.org.

Figure 3: *Distribution of precision estimates for various sample sizes and misattribution rates, as described in section 2.4. Ground truth precision is marked by a dotted vertical line. The `rate` variable represents the percentage misattribution rate. The estimator `P_cluster_block` is highly accurate, while `P_naive` is almost always close to* 1.0*, having little to do with the true precision.*

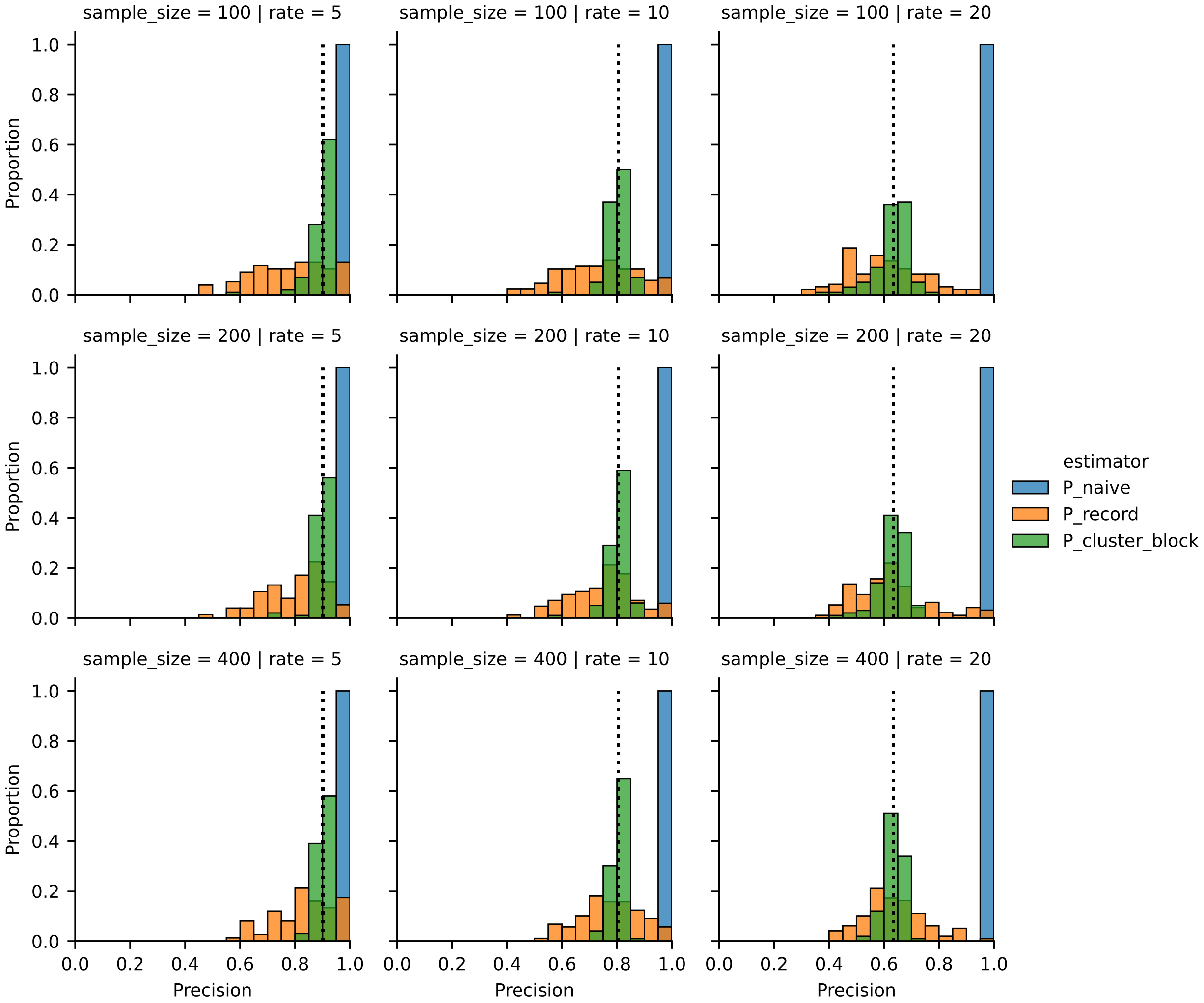
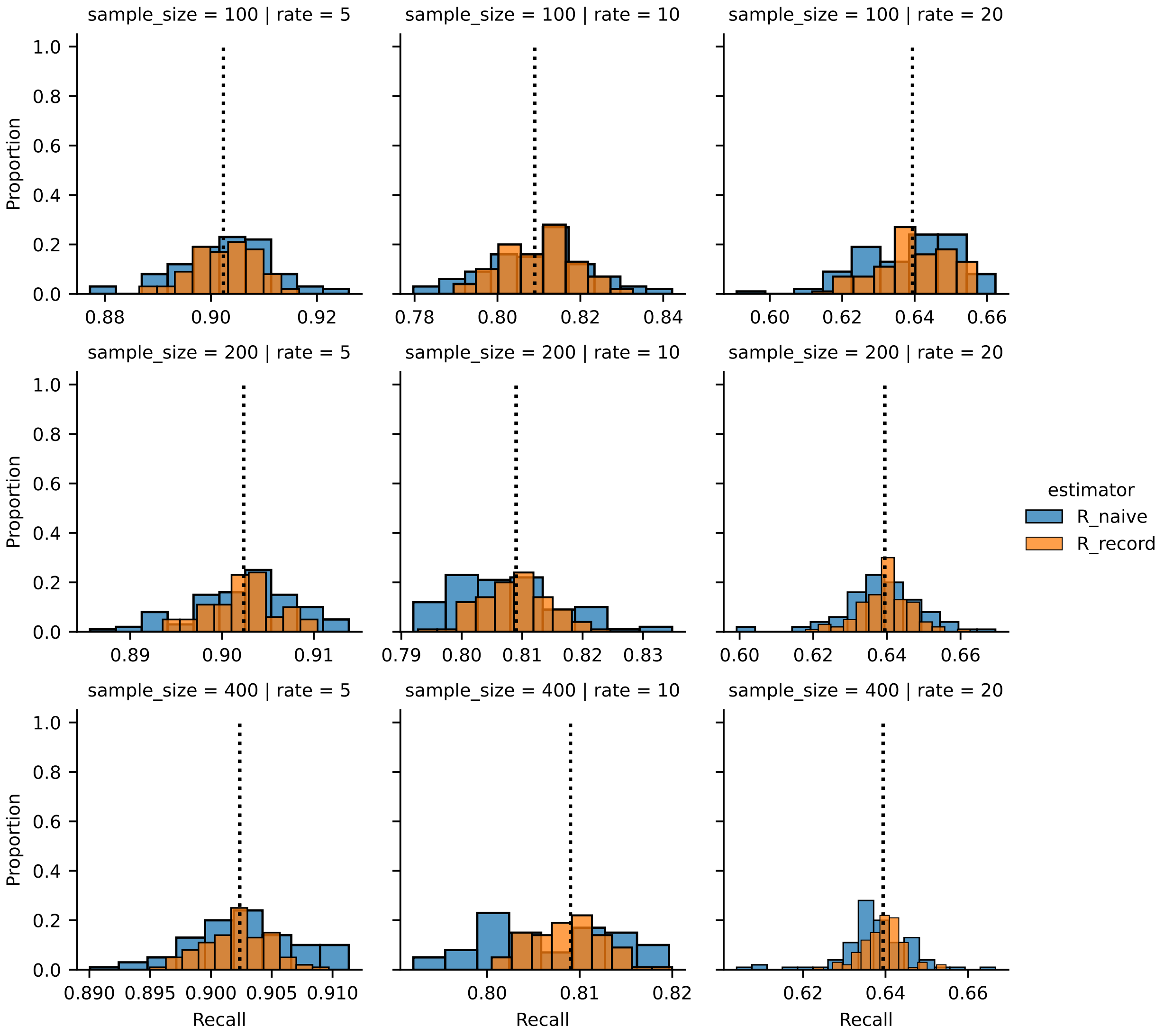


Table 2: *Bias and root mean squared error of recall estimators for the simulation study described in section 2.4. The `rate` variable represents the percentage misattribution rate.*

	rate	5			10			20		
	sample size	100	200	400	100	200	400	100	200	400
bias	R_naive	0.0007	0.0001	0.0004	0.0009	-0.0015	-0.0024	-0.0008	-0.0010	-0.0017
	R_record	0.0002	-0.0001	-0.0002	0.0011	-0.0002	-0.0004	0.0005	0.0001	-0.0002
rmse	R_naive	0.0089	0.0056	0.0043	0.0123	0.0089	0.0075	0.0132	0.0109	0.0091
	R_record	0.0060	0.0037	0.0028	0.0089	0.0056	0.0039	0.0099	0.0071	0.0050

Figure 4: *Distribution of recall estimates for various sample sizes and misattribution rates, as described in section 2.4. Ground truth recall is marked by a dotted vertical line. The* `rate` *variable represents the percentage misattribution rate.*



The choice of estimators for the results presented in table 3 is as follows. Since our two hand-curated datasets (see section 2.2) were obtained by sampling inventor clusters with probabilities proportional to cluster sizes, we used the `P_cluster_block` and `R_record` estimators with corresponding probability weights. For the Israeli benchmark dataset, we assumed that a single block of inventor clusters was sampled and used the corresponding estimators defined through (5) with a single block sample. Note that no variance estimates can be given for single samples. For Li et al. (2014)'s inventors benchmark, given that the inventor clusters were originally obtained from a set of inventor curriculum vitae, we assumed that the clusters were sampled uniformly at random. As such, we used cluster block estimators with constant probability weights.

Table 3: *Estimated pairwise precision and recall (with estimated standard deviation) from our four benchmark datasets.*

dataset	**est. precision ($\hat{\sigma}$)**	**est. recall ($\hat{\sigma}$)**	**scope**
Staff 1	88% (3.4%)	95% (1.1%)	1976 – Dec. 31, 2022 (U.S. granted)
Staff 2	87% (3.6%)	96% (1.0%)	1976 – Dec. 31, 2022 (U.S. granted)
Israeli Benchmark	79% (NA)	94% (NA)	1976 – 1999 (U.S. granted)
Li et al. (2014)'s Benchmark	91% (2.7%)	91% (5.0%)	1976 – 2010 (U.S. granted)

Overall, our performance estimates paint the first realistic picture of PatentsView's disambiguation accuracy in practice. Precision is not nearly 100%, as would be assumed from naively computing precision on benchmark datasets. Rather, there is significant room for improvement. Our hand-curated datasets and data collection methodology provide the necessary basis to investigate errors and plan for improvements to the disambiguation algorithm.

4 Discussion

Motivated by PatentsView's disambiguation, this paper introduced a novel evaluation methodology for entity resolution algorithms. The methodology relies on benchmark datasets containing ground truth clusters and estimators that account for biases inherent to these datasets. For PatentsView, this provided the first representative estimates of its disambiguation performance. Furthermore, all data and code used in this paper, as well as other tools developed to facilitate evaluation at PatentsView, are freely available at https://github.com/PatentsView/PatentsView-Evaluation/.

There are two main products resulting from this work. The first is the appropriate understanding of the quality of the data provided to PatentsView's users. Our performance estimates indicate that, despite an overall accurate disambiguation, there is significant room for improvement. Notably, the current disambiguation over-estimates the number of matching inventor mention pairs. The second product is the set of tools needed for methodological research and model comparison. Given our evaluation methodology, we can now reliably compare algorithms and decide with confidence on changes that will affect users.

One important topic for future work is the quantification of uncertainty associated with errors in the hand-disambiguation process. This is challenging problem given the lack of validation information available. Surveying inventors to validate the hand-disambiguation process would be one way to explore this issue. Sensitivity analyses involving potential errors in the hand-disambiguation process could also be informative. We refer the reader to Bailey et al. (2017) for a state-of-the-art hand-labeling study that evaluated human review accuracy.

Another important topic for future work is the development of estimators for additional performance metrics. As we have seen in the simulation study, the generically derived estimators (e.g., from lemma 2) do not perform as well as estimators derived using specific properties of pairwise precision and recall (lemma 3). As such, care should be taken to obtain efficient estimators for every metric of interest.

Finally, we note that the performance of estimators can degrade when dealing with heavy-tailed cluster size distributions. The bias of ratio estimators can be high in this case, especially when using small samples and when sampling clusters uniformly at random rather than with probability proportional to size. Model-based estimators that exploit known properties of the cluster size distribution could be developed to improve estimation accuracy in such cases.

Data and Code

All data and code used for this paper are available as part of the PatentsView-Evaluation Python package at https://github.com/PatentsView/PatentsView-Evaluation/.

Author Contributions

Olivier Binette led the evaluation project and wrote most of this manuscript. Sokhna A York and Emma Hickerson carried out the data collection by manually reviewing inventor clusters. Youngsoo Baek provided bias adjustment and uncertainty quantification for ratio estimators. Sarvo Madhavan was a technical advisor and contributed to code. Christina Jones was an advisor and project manager. All authors provided input on the manuscript.

References

Bailey, M. J., C. Cole, M. A. Henderson, and C. G. Massey (2017). *How Well Do Automated Methods Perform in Historical Samples?: Evidence From New Ground Truth.* National Bureau of Economic Research.

Balsmeier, B., M. Assaf, T. Chesebro, G. Fierro, K. Johnson, S. Johnson, G.-C. Li, S. Lück, D. O'Reagan, B. Yeh, et al. (2018). Machine learning and natural language processing on the patent corpus: Data, tools, and new measures. *Journal of Economics & Management Strategy 27*(3), 535–553.

Barnes, M. (2015). A Practioner's Guide to Evaluating Entity Resolution Results. pp. 1–6.

Belin, T. R. and D. B. Rubin (1995). A method for calibrating false-match rates in record linkage. *Journal of the American Statistical Association 90*(430), 694–707.

Bilenko, M. and R. Mooney (2003). On evaluation and training-set construction for duplicate detection. *Proceedings of the KDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation*, 7–12.

Binette, O. and R. C. Steorts (2022). (Almost) all of entity resolution. *Science Advances 8*(12), eabi8021.

Christen, P. (2012). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Data-Centric Systems and Applications. Berlin Heidelberg: Springer-Verlag.

Christen, P. (2019). Data linkage: The big picture. *Harvard Data Science Review.*

Christen, P. and K. Goiser (2007). Quality and complexity measures for data linkage and deduplication. *Studies in Computational Intelligence 43*, 127–151.

Christophides, V., V. Efthymiou, T. Palpanas, G. Papadakis, and K. Stefanidis (2021). An overview of end-to-end entity resolution for big data. *ACM Computing Surveys 53*(6), 1–2.

Cochran, W. G. (1977). *Sampling Techniques.* New York: John Wiley.

Doherr, T. (2021). Disambiguation by namesake risk assessment. *ZEW-Centre for European Economic Research Discussion Paper* (21-021).

Dong, X. L. and D. Srivastava (2015). *Big Data Integration.* Morgan and Claypool Publishers.

Draisbach, U. and F. Naumann (2013). On choosing thresholds for duplicate detection. *Proceedings of the 18th International Conference on Information Quality, ICIQ 2013*.

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

Fellegi, I. P. and A. B. Sunter (1969). A theory for record linkage. *Journal of the American Statistical Association 64*(328), 1183–1210.

Ferreira, A. A., M. A. Gonçalves, and A. H. Laender (2012). A brief survey of automatic methods for author name disambiguation. *ACM Sigmod Record 41*(2), 15–26.

Frisoli, K. and R. Nugent (2018). Exploring the effect of household structure in historical record linkage of early 1900s ireland census records. In *Proceedings of the 2018 IEEE International Conference on Data Mining Workshops*, pp. 502–509. IEEE.

Fuller, W. A. (2011). *Sampling Statistics.* John Wiley.

Gu, G., S. Lee, and J. Kim (2008). Matching accuracy of the lee-kim-marschke computer matching program.

Han, H., Y. Yu, L. Wang, X. Zhai, Y. Ran, and J. Han (2019). Disambiguating USPTO inventor names with semantic fingerprinting and DBSCAN clustering. *The Electronic Library 37*(2), 225–239.

Hand, D. and P. Christen (2018). A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing 28*(3), 539–547.

Herzog, T., F. Scheuren, and W. Winkler (2007). *Data Quality and Record Linkage Techniques.* New York, NY: Springer.

Horvitz, D. G. and D. J. Thompson (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association 47*(260), 663–685.

Ilyas, I. F. and X. Chu (2019). *Data Cleaning.* New York, NY, USA: Association for Computing Machinery.

Kim, K., M. Khabsa, and C. L. Giles (2016). Random forest DBSCAN for USPTO inventor name disambiguation. *arXiv:1602.01792*.

Li, G. C., R. Lai, A. D'Amour, D. M. Doolin, Y. Sun, V. I. Torvik, A. Z. Yu, and F. Lee (2014). Disambiguation and co-authorship networks of the U.S. patent inventor database (1975-2010). *Research Policy 43*(6), 941–955.

Maidasani, H., G. Namata, B. Huang, and L. Getoor (2012). Entity Resolution Evaluation Measures. Technical report, University of Maryland.

Marchant, N. G. and B. I. Rubinstein (2017). In search of an entity resolution OASIS: Optimal asymptotic sequential importance sampling. *Proceedings of the VLDB Endowment 10*(11), 1322–1333.

McVeigh, B. S., B. T. Spahn, and J. S. Murray (2019). Scaling bayesian probabilistic record linkage with post-hoc blocking: an application to the california great registers. *arXiv:1905.05337*.

Menestrina, D., S. E. Whang, and H. Garciamolina (2010). Evaluating entity resolution results. *Proceedings of the VLDB Endowment 3*(1), 208–219.

Michelson, M. and S. A. Macskassy (2009). Record linkage measures in an entity centric world. *Proceedings of the 4th Workshop on Evaluation Methods for Machine Learning*.

Monath, N., C. Jones, and S. Madhavan (2021). PatentsView: Disambiguating Inventors, Assigness, and Locations. Technical report, American Institutes for Research, Arlington, Virginia.

Monath, N., A. Kobren, A. Krishnamurthy, M. R. Glass, and A. McCallum (2019). Scalable hierarchical clustering with tree grafting. In *In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1438–1448.

Morrison, G., M. Riccaboni, and F. Pammolli (2017). Disambiguation of patent inventors and assignees using high-resolution geolocation data. *Scientific Data 4*(1), 1–21.

Müller, M.-C. (2017). Semantic author name disambiguation with word embeddings. In *Proceedings of the 21st International Conference on Theory and Practice of Digital Libraries*, pp. 300–311. Springer.

Newcombe, H. B., J. M. Kennedy, S. J. Axford, and A. P. James (1959). Automatic linkage of vital records. *Science 130*(3381), 954–959.

Papadakis, G., E. Ioannou, E. Thanos, and T. Palpanas (2021). *The Four Generations of Entity Resolution.* Morgan & Claypool Publishers.

Sariyar, M. and A. Borg (2022). *RecordLinkage: Record Linkage Functions for Linking and Deduplicating Data Sets.* R package version 0.4-12.3.

Tam, D., N. Monath, A. Kobren, A. Traylor, R. Das, and A. McCallum (2019). Optimal transport-based alignment of learned character representations for string similarity. *arXiv:1907.10165*.

Toole, A., C. Jones, and S. Madhavan (2021). Patentsview: An open data platform to advance science and technology policy.

Trajtenberg, M. and G. Shiff (2008). *Identification and Mobility of Israeli Patenting Inventors.* Pinhas Sapir.

Traylor, A., N. Monath, R. Das, and A. McCallum (2017). Learning string alignments for entity aliases. In *In Proceedings of the 31st Conference on Neural Information Processing Systems.*

Ventura, S. L., R. Nugent, and E. R. Fuchs (2013). Methods matter: Rethinking inventor disambiguation with classification & labeled inventor records. In *Academy of Management Proceedings*, Volume 2013. Academy of Management Briarcliff Manor, NY 10510.

Ventura, S. L., R. Nugent, and E. R. Fuchs (2015). Seeing the non-stars: (Some) sources of bias in past disambiguation approaches and a new public tool leveraging labeled records. *Research Policy 44*(9), 1672–1701.

Wang, T., H. Lin, C. Fu, X. Han, L. Sun, F. Xiong, H. Chen, M. Lu, and X. Zhu (2022). Bridging the gap between reality and ideality of entity matching: A revisiting and benchmark re-construction. arxiv:2205.05889.

Yang, G.-C., C. Liang, Z. Jing, D.-R. Wang, and H.-C. Zhang (2017). A mixture record linkage approach for US patent inventor disambiguation. In *Advanced Multimedia and Ubiquitous Engineering*, pp. 331–338. Springer.

A Appendix

A.1 Bias of Precision Computed on Benchmark Datasets

This section provides more information on example 1.

For this example, we considered the RLdata10000 dataset from Sariyar and Borg (2022). This is a synthetic dataset containing 10,000 records with first name, last name, and date of birth attributes. There is noise in these attributes and a 10% duplication rate. Ground truth identity is known for all records.

The disambiguation algorithm we consider matches records if any of the following conditions are met:

- records agree on first name, last name, and birth year,
- records agree on first name, birth day, and birth year, or
- records agree on last name, birth day, and birth year.

Note that this is not at all a good disambiguation algorithm. It has 52% precision and 83% recall. However, it allows us to to showcase the issue with nonadjusted precision computed on cluster samples.

In our experiment, we have repeated 5,000 times the following three-steps process 5,000 times:

1. First, 200 records were sampled and the ground truth clusters associated with them were recovered. This step provided a "benchmark" dataset that was used for evaluation.
2. Second, a trivial precision estimate was obtained by computing precision over the benchmark dataset. That is, predicted cluster assignments were restricted to records that appear in the benchmark data and precision was compared for these records. More often than not, the result was an observation of 100% precision.
3. Third, we computed our proposed precision estimator which corresponds to lemma 1 and the estimator $\widehat{P}_{\text{block}}$ defined in (14), with blocks corresponding to clusters and with sampling probabilities $p_b \propto |b|$.

The distributions of the two precision estimates over the 5,000 repetitions are shown in figure 1. Our proposed estimator is accurate and nearly unbiased, whereas the trivial precision estimates have almost nothing to do with actual algorithmic performance.

A.2 Precision and Recall Estimator Formulas

This section describes specific values for A_s, B_s, and T in (7) in order to obtained nearly unbiased precision and recall estimators based on each of the representation in section 2.3. We use the symbols $\widehat{P}$ and $\widehat{R}$, indexed by either "rec", "clust", or "block", in order to refer to precision and recall estimators corresponding to record, cluster, and block sampling representations, respectively.

Record Sampling Estimators

$\widehat{P}_{\textbf{rec}}$ Estimating P is a special case that does not require ratio-of-means estimation. We propose to use a simple unbiased estimator:

$$\widehat{P}_{\text{rec}} = \frac{1}{n}\sum_{s=1}^{n} \frac{g(c(i_s), \widehat{\mathcal{C}})}{|c(i_s)| p_{i_s}}. \tag{9}$$

The unbiased estimator of the variance of $\widehat{P}_{\text{rec}}$ is also available:

$$\widehat{V}(\widehat{P}_{\text{rec}}) = \frac{\theta_{n,N}}{n(n-1)} \sum_{s=1}^{n} \left(\frac{g(c(i_s), \widehat{\mathcal{C}})^2}{|c(i_s)|^2 p_{i_s}^2} - \widehat{P}_{\text{rec}}^2. \right) \tag{10}$$

$\widehat{R}_{\textbf{rec}}$ Going forward, we refer to formulae (7) and (8). $\widehat{R}_{\text{rec}}$ and its variance estimate $\widehat{V}(\widehat{R}_{\text{rec}})$ are given by (7) and (8), where we substitute in

$$T = N,\ A_s = \frac{(|c(i_s)| - 1)}{p_{i_s}},\ B_s = 2\frac{f(c(i_s), \widehat{\mathcal{C}})}{|c(i_s)| p_{i_s}}. \tag{11}$$

Cluster Sampling Estimators

$\widehat{P}_{\textbf{clust}}$ The estimator and its variance estimate $\widehat{V}(\widehat{P}_{\text{clust}})$ are given by (7) and (8), where we substitute in

$$T = |\mathcal{C}|,\ A_s = \frac{|c_s|}{p_{c_s}},\ B_s = N\frac{g(c_s, \widehat{\mathcal{C}})}{p_{c_s}}. \tag{12}$$

$\widehat{R}_{\textbf{clust}}$ The estimator and its variance estimate $\widehat{V}(\widehat{R}_{\text{clust}})$ are given by (7) and (8), where we substitute in

$$T = |\mathcal{C}|,\ A_s = \frac{\binom{|c_s|}{2}}{p_{c_s}},\ B_s = \frac{f(c_s, \widehat{\mathcal{C}})}{p_{c_s}}. \tag{13}$$

Disjoint Block Sampling Estimators

$\widehat{P}_{\textbf{block}}$ The estimator and its variance estimate $\widehat{V}(\widehat{P}_{\text{block}})$ are given by (7) and (8), where we substitute in

$$T = |\mathcal{B}|,\ A_s = \frac{|\mathcal{P}_{b_s}| + \frac{1}{2}|\mathcal{P}_{b_s}^-|}{p_{b_s}},\ B_s = \frac{|\mathcal{T}_{b_s} \cap \mathcal{P}_{b_s}|}{p_{b_s}}. \tag{14}$$

$\widehat{R}_{\textbf{block}}$ The estimator and its variance estimate $\widehat{V}(\widehat{R}_{\text{block}})$ are given by (7) and (8), where we substitute in

$$T = |\mathcal{B}|,\ A_s = \frac{|\mathcal{T}_{b_s}|}{p_{b_s}},\ B_s = \frac{|\mathcal{T}_{b_s} \cap \mathcal{P}_{b_s}|}{p_{b_s}}. \tag{15}$$

A.3 Estimators Implementation

The proposed estimators have been implemented as part of the **pv_evaluation** Python package available on GitHub at `https://github.com/PatentsView/PatentsView-Evaluation`. Given a predicted clustering `prediction` and given a set of sampled clusters `sample`, each represented as a membership vector (pandas Series objects with mention identifiers as an index and cluster assignment identifiers as values), estimates can be obtained as follows. First, the sampling mechanism has to be chosen. This can be one of "`record`" for record sampling as in lemma 1, "`cluster`" for the estimator corresponding to cluster sampling as in lemma 2, "`cluster_block`" for cluster sampling with the estimator corresponding to lemma 3, and "`single_block`" for the sampling of a single block with the estimator corresponding to lemma 3. Then, probability weights can be set to either "`uniform`" for uniform sampling, or "`cluster_size`" for sampling with probability proportional to size.

For example, when sampling clusters with probability proportional to their sizes and using the estimator corresponding to lemma 3, precision estimates can be obtained as follows:

```
from pv_evaluation.estimators import pairwise_precision_estimator
pairwise_precision_estimator(
    prediction,
    sample,
    sampling_type="cluster_block",
    weights="cluster_size"
)
```

Standard deviation estimates are obtained similarly using the `pairwise_precision_std()` function from the `pv_evaluation.estimators` module. The complete API documentation can be consulted at https://patentsview.github.io/PatentsView-Evaluation.

A.4 Proofs

Proof of lemma 1. By breaking down $\mathcal{P}$ and $\mathcal{T}$ over predicted clusters, we find

$$P = \sum_{c\in\mathcal{C}}\sum_{\hat{c}\in\widehat{\mathcal{C}}}\binom{|c\cap\hat{c}|}{2}\Big/\sum_{\hat{c}\in\widehat{\mathcal{C}}}\binom{|\hat{c}|}{2}. \tag{16}$$

Now writing $\sum_{c\in\mathcal{C}}\sum_{\hat{c}\in\widehat{\mathcal{C}}}\binom{|c\cap\hat{c}|}{2} = \sum_{i=1}^{N}\frac{1}{|c(i)|}\sum_{\hat{c}\in\widehat{\mathcal{C}}}\binom{|c(i)\cap\hat{c}|}{2}$ and substituting $g(c(i),\widehat{\mathcal{C}})$, we obtain

$$P = \sum_{i=1}^{N} p_i \frac{g(c(i)),\widehat{\mathcal{C}})}{p_i|c(i)|} = \mathbb{E}\left[\frac{g(c(i)),\widehat{\mathcal{C}}}{p_i|c(i)|}\right]. \tag{17}$$

For recall, write

$$R = \sum_{c\in\mathcal{C}}\sum_{\hat{c}\in\widehat{\mathcal{C}}}\binom{|c\cap\hat{c}|}{2}\Big/\sum_{c\in\mathcal{C}}\binom{|c|}{2}. \tag{18}$$

Through a similar argument as above, we may express the numerator as

$$\sum_{c\in\mathcal{C}}\sum_{\hat{c}\in\widehat{\mathcal{C}}}\binom{|c\cap\hat{c}|}{2} = \mathbb{E}\left[f(c(i),\widehat{\mathcal{C}})/(|c(i)|p_i)\right]. \tag{19}$$

For the denominator, we have

$$\sum_{c\in\mathcal{C}}\binom{|c|}{2} = \sum_{i=1}^{N}\frac{1}{|c(i)|}\binom{|c(i)|}{2} = \mathbb{E}[(|c(i)|-1)/(2p_i)]. \tag{20}$$

Combining (19) and (20) yields the result. □

Proof of lemma 2. Let $\pi > 0$ be such that $\sum_{c\in\mathcal{C}}\pi p_c = 1$. Now write

$$P = \sum_{c\in\mathcal{C}} g(c,\widehat{\mathcal{C}}) = |\mathcal{C}|\sum_{c\in\mathcal{C}}\pi p_c g(c,\widehat{\mathcal{C}})/(\pi p_c|\mathcal{C}|) = |\mathcal{C}|\,\mathbb{E}\left[g(c,\widehat{\mathcal{C}})/(\pi p_c|\mathcal{C}|)\right] \tag{21}$$

and

$$|\mathcal{C}| = \frac{N}{\frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}|c|} = \frac{N}{\mathbb{E}[|c|/(\pi p_c|\mathcal{C}|)]}. \tag{22}$$

Simplifying $\pi|\mathcal{C}|$ from the numerator and denominator then yields the expression for precision.

The expression for recall follows in a straightforward way from (18). □

Proof of lemma 3. Since the blocking procedure is assumed to have no error (for every $c \in \mathcal{C}$, there exists $b \in \mathcal{B}$ with $c \subset b$), we can break down $\mathcal{T}$ as the disjoint union of the $\mathcal{T}_b$'s over $b \in \mathcal{B}$. It follows that $|T| = \sum_{b\in\mathcal{B}}|\mathcal{T}_b|$ and $|\mathcal{T} \cap \mathcal{P}| = \sum_{b\in\mathcal{B}}|\mathcal{T}_b \cap \mathcal{P}_b|$. The expression for recall in (5) follows directly. For precision, we can express $|\mathcal{P}|$ as the number of links within blocks plus the number of links across blocks. Since the number of links across blocks is counted twice when each block is considered, we obtain $|\mathcal{P}| = \sum_{b\in\mathcal{B}}\left(|\mathcal{P}_b| + \frac{1}{2}|\mathcal{P}_b^-|\right)$. □