// NO.21-021 | 03/2021

DISCUSSION PAPER

// THORSTEN DOHERR

Disambiguation by Namesake Risk Assessment

Leibniz Association

ZEW

Disambiguation by Namesake Risk Assessment

Thorsten Doherr[a,b]

February 2021

a) Leibniz Centre for European Economic Research (ZEW), Mannheim, Germany
b) University of Luxembourg

Abstract Most bibliometric databases only provide names as the handle to their careers leading to the issue of namesakes. We introduce a universal method to assess the risk of linking documents of different individuals sharing the same name with the goal of collecting the documents into personalized clusters. A theoretical setup for the probability of drawing a namesake depending on the number of namesakes in the population and the size of the observed unit replaces the need for training datasets, thereby avoiding a namesake bias caused by the inherent underestimation of namesakes in training/benchmark data. A Poisson model based on a master sample of unambiguously identified individuals estimates the main component, the number of namesakes for any given name. To implement the algorithm, we reduce the complexity in the data by resolving similarity in properties. At the core of the implementation is a mechanism returning the unit size of the intersected mutual properties linking two documents. Because of the high computational demands of this mechanism, it is a necessity to discuss means to optimize the procedure.

Keywords: homonymy, namesakes, disambiguation, scientific careers, inventors, patents, publications

JEL: C18, C36

1 Introduction

Bibliographic and patent databases have comparable structures as they both are collections of documents cataloged by fixed retrieval criteria, like author or inventor name, title, abstract, academic discipline or international patent classifications, keywords, filing and publications dates, citations, affiliations respectively applicants and so on. Patent offices, public institutions and even private providers foster the access to this data to extend the knowledge about the treasures they hoard through academic research. The ubiquity of the data facilitates new research approaches especially in the fields of innovation economics and bibliometric analysis. It does not take long until researchers were not content by only exploiting the document, i.e. patent or publication, as observation unit. Linking the documents to other sources, for instance firm panels, extends the utility for researchers to deepen the insights into the mechanisms of innovation and research. However, the actual protagonists of these mechanisms, the authors and inventors, are in most cases not the focus of these efforts. The fuzziness of names as the main handle to their careers requires disproportionately complex identification strategies. Nevertheless, these individuals are the main driver of human progress and deserve close inspection.

The best solution to the issue would be the assignment of a unique author identification number (UAIN) to every author or inventor retrievable from every document or patent published (Fallgas, 2016). An implementation of a mandatory author identifier only exists for Brazil, the Netherlands and for some selected research fields (Fenner, 2010). As far as we now, no patent authority has introduced a mandatory identifier for inventors. Other efforts, like the ORCID (Open Researcher and Contributor ID), target specifically large publication data providers like Web of Science or Scopus, which are more open to the needs of their prime audience, the researchers. Patent authorities are less inclined to support researchers because their assignment is the administration of legal documents. Their key audience consists of lawyers, patent assignees, firms and other patent authorities. Even though some institutions already apply administrative methods to identify authors, this only covers documents filed under the current regime. Older documents remain unchanged and the associated author careers ambiguous. This situation forced researchers to implement their own disambiguation methods. An early effort by Singh (2003) relies on a combination of name and patent subcategory match, while Jones (2005) and Fleming and Marx (2006) concentrate mainly on

the names, latter taking the frequencies of the last names and mutual co-inventors into account. Trajtenberg, Shiff and Melamed (2004 and 2006) inaugurate the “Names Game” season, introducing a score based method with matching parameters fine-tuned by using a dataset of manually disambiguated Israeli inventors. The paper already acknowledges the effect of large assignees or cities in conjunction with common names on the probability of causing false positives. The size of an assignee or city and the commonness of an inventor name is measured by the number of patents sharing this specific unit. A link between two documents by these criterions is regarded weaker for high patent counts. These frequencies are part of the parameters, determining the strength of a document link, to be weighted by the iterative fine-tuning process. Trajtenberg et al. (2006) also discusses the existence of an intransitivity “conundrum”: document A can be linked to document B with a high probability. The same is valid for B and C, but A and C do not match. The authors decide to impose transitivity in such cases stating this as the only plausible action, even though they consider this not an “innocent decision”.

In another approach, Torvik and Smalheiser (2009) apply the “Author-ity” model to 15.3 million articles in the MEDLINE database. It requires training data consisting of a match set of pairs of articles with a high probability being from the same author and a non-match set of document pairs of obviously different authors. For a given document pair a similarity profile is computed and compared with both training sets, returning the respective relative frequency of the profile within each set. The ratio of both values is the so-called r-value. The main formula to estimate the pairwise probability of a valid match incorporates, next to the r-value, the document count per name as a priori match probability. Although the authors criticize the inaccuracy of this proxy, they consider it a reasonable heuristic value for the ensuing steps. Given the Bayesian approach, tolerating intransitivity is not an option and therefore solved by iterative smoothing of triplet violations. The final clustering of the comprising tuples relies on a maximum likelihood framework.

Pezzoni, Lissoni and Tarasconi (2012) construct a list of 17 matching criteria for their “Massacrator” algorithm. Some of these include Meta information based on aggregated data. They classify an applicant as small, if less than 50 inventors are affiliated. The paper omits the method of the size estimation. We insinuate an aggregation on the inventor name level by applicant as the most obvious procedure. They identity a rare surname by counting its

occurrence by patents within the inventor's country. The criteria are weighted by a Monte Carlo simulation balancing recall and precision measured by a training dataset consisting of the "Noise Added French Academic" (NAFA) and the "Noise Added EPFL" (NAE) benchmark data promoted by the "Name Game" algorithm challenge of the APE-INV (Academic Patenting in Europe) initiative of the European Science Foundation (Lissoni, 2010). Schön, Heinisch and Bünsdorf (2014) apply a similar method based on less matching criteria, called classifiers. A classifier differentiates between matching, non-matching and missing patent properties implementing an additional degree of freedom. The classifier patterns returning the highest accuracy is determined by testing them against a set of manually matched document pairs, enjoying a high confidence of being from the same inventor, and a randomly matched negative set. The team obviously found a way to identify "common surnames" as a classifier, but did not elaborate on how that was conducted.

Other approaches are deeply rooted in the realm of machine learning. Therefore, they always require a training dataset of already classified documents. Kim, Kabsa and Giles (2016) use pairs of patents from the same inventor and pairs from different inventors to train a random forest classifier to produce a decision tree on the matching criteria. The output of this tree is the distance between documents. By applying a DBSCAN clustering algorithm, they resolve the resulting document network. Petrie, Julius and Thomson (2017) use a similar training set consisting of matching and non-matching document tuples to train a neural network called AlexNet (Krizhevsky, 2017) specialized in image classification. To feed the network they converted the document data into a graphical representation based on color coding and 2D mapping.

The literature so far does not provide deeper insights of the main culprit inciting all these efforts, the namesake. A namesake is "someone or something that has the same name as another person or thing" (Merriam-Webster's Learner's Dictionary). This issue is generally circumvented by methods optimizing parameters and thresholds for sets of matching criteria using training data. In the ideal case, that data is based on real word observation of authors or inventors like the surveyed samples of French and Swiss inventors of the NAFA and NAE benchmarks, or the self-assigned ORCHID. As the availability of those convenient datasets is the exception, researchers have to fall back on classification of sample documents based on intuitive assessment of whether a document is from the same individual or from two

namesakes. This article introduces a theoretical model for the probability of encountering namesakes simulating the intuitive namesake risk assessment and replacing the necessity of training data.

Some names are comparable with unique identifiers without any risk of encountering namesakes, while other, more common names involve a high risk of linking documents from namesakes. One challenge is to find a way to differentiate between unique and common names to assess the risk of linking careers of namesakes. For any given name, this risk increases with the number of namesakes in the population. It also depends on the size of the observed reference unit. Although a person may have a very common name, there is a low risk of encountering a namesake, if the reference is a small firm. A reference unit is not limited to physical stations like affiliations accrued during an author's or inventor's career, but can also be a research area, a technology field, a co-author network, special interests manifested by citations, keywords, repeating topics in titles or even a combination of multiple contexts. In this paper, we discuss the theory of namesake risk assessment, a method to estimate the number of namesakes, the identification of unit sizes and the inherent underrepresentation of namesakes in training/benchmark data inevitably resulting in a namesake bias. We substantiate this knowledge by describing the implementation of a universal disambiguation algorithm. The paper concludes with application examples omitting benchmarks for reasons explored in the paper.

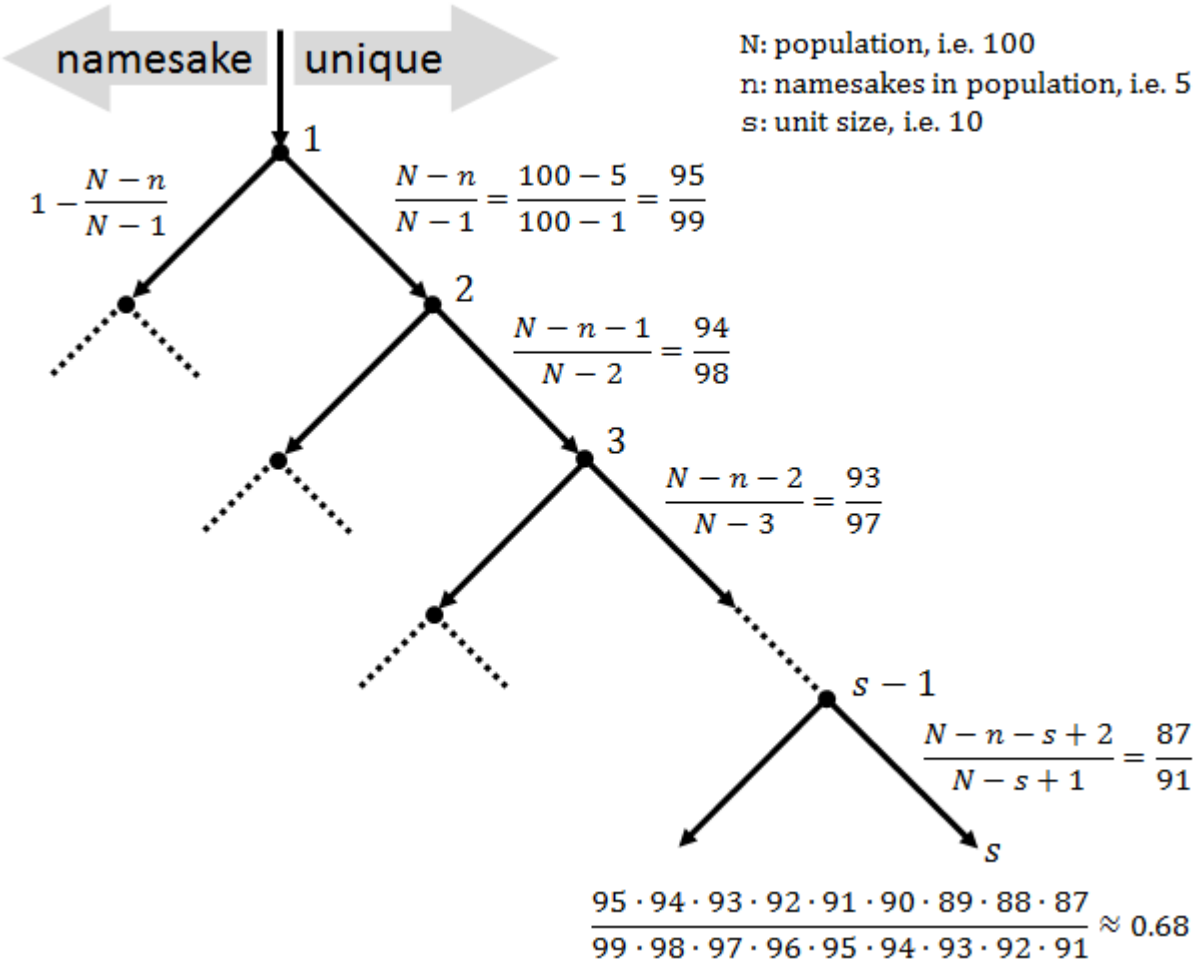
2 Namesakes

2.1 Likelihood of a Namesake

To explain the theory of namesakes, it is helpful to picture a concrete example for a reference unit: a firm. For our analogy, we assume that our firm has only one employee at the beginning - the founder. The firm employs more and more individuals from a finite pool, until it reaches its current size. With every new entrant, the risk for a namesake to the founder increases. The extreme case of employing/drawing the complete population dictates this interrelation. As we are only interested in the risk of drawing any namesakes, it is sufficient to handle the reverse case of drawing no namesakes. The probability of drawing a valid employee equals the remaining number of individuals in the population that are no namesakes to the founder, divided by the remaining population size. With every new employee, the numerator and the

denominator of this relation decrease by one. Figure 1Figure 1illustrates the development of the probability of the founder staying unique.

Figure 1: Probability tree for drawing a namesake vs. staying unique



In our example, the population N consists of 100 individuals. The number of namesakes n in the population is 5, including the founder. The final unit size, respectively firm size, s is 10. The probability of drawing a namesake with the first employee is $1-95/99\approx 0{,}0404$. The probability of drawing a namesake to the founder with the second employee is only slightly larger: $1-94/97\approx 0{,}0408$. The parallel decrement of the numerator and the denominator has only a very small impact on the probabilities for larger populations, a circumstance we will exploit for a simplification. The product of the probabilities of the unique branch is 68%, therefore is the likelihood for at least one other person in the firm with the same name as the founder 32%.

The most straightforward implementation of the probability of drawing a namesake for an individual with n namesakes within a population of N individuals for a unit with size s is 1 minus the product of all stepwise probabilities:

$$P(namesake) = 1 - \prod_{i=0}^{s-2} \frac{N-n-i}{N-1-i} \quad (1)$$

The number of operations to calculate the probability depends on the unit size s, making (1.1) an unwieldy proposition for large units. Figure 1 already hints a way for a formula that is independent from the unit size. The term at the bottom shows a division of product sequences, which can be constructed by using factorials:

$$P(namesake) = 1 - \frac{(N-n)!/(N-n-s+1)!}{(N-1)!/(N-s)!} \quad (2)$$

Of course, factorials are even more cumbersome as they grow very fast beyond the capabilities of contemporary computing systems. For example, the factorial of 171 is already too large to be properly represented by the numerical data type with the highest precision used by statistical software packages (double, 8 bytes). It is suggested to use the natural log of the factorials approximated by James Stirling's formula published 1730:

$$lnf(x) = \left(x + \frac{1}{2}\right) ln(x+1) - (x+1) + \frac{1}{2} ln(2\pi) \quad (3)$$

With the support of this handy approximation, it is possible to rewrite (1.3) avoiding factorials altogether:

$$P(namesakes) = 1 - exp\big(lnf(N-n) - lnf(N-n-s+1) - lnf(N-1) + lnf(N-s)\big) \quad (4)$$

Because the number of operations to calculate the probability stays always constant, we use the final form (4) in our implementation of the disambiguation algorithm.

To fill this theory with life, we need to determine the number of namesakes for any given name in a population. Of course, we do not have the luxury of name repositories for any occasion. The following section discusses a method to estimate the number of namesakes

based on a representative sample and a highly correlated indicator, which fulfills the requirement of being derivable for any name population.

2.2 The Indicator

In most cultures, the last name is inherited from the parents and there are only few instances where it may be subject to change, i.e. marriage. Parents that bear a common last name may choose an exotic first name for their child to stand out. They may choose to name their child according to their family tradition, i.e. first name of a grandparent and thus explicitly creating a namesake. The density of namesakes also depends on temporary trends that influence naming decisions. A common last name generates a high number of variants in terms of first names. For a common first name, we naturally observe a high amount of different last names in the population. If we pick an uncommon first or last name, we expect less variation within the other part of the name. A look in an old-fashioned telephone book reveals that the combination of common last names with common first names is responsible for most namesake occurrences. In fact, such an entry can only be saved from having a namesake by a less common first name. This trivial observation leads us to the assumption, that the number of namesakes is positively correlated with the occurrence of the part of a name that appears less often in the population.

To define the indicator, we aggregate a population on the name level by removing duplicate entries. This procedure can be conducted for any population containing names, regardless of the original context of an observation. For every name, a minimum occurrence is calculated by counting the frequencies of every name part in the name aggregate and choosing the respective minimum. After sorting the name aggregate by the minimum occurrences in ascending order, we apply a dense ranking ("1223444...") and a final normalization to the range]0,1]. The intention of the normalized minimum occurrence rank, further called *minocc*, is the harmonization of the distributions of different name aggregates by obfuscating the frequencies. The *minocc* of a common name is close to 1 whereas the minocc of a unique name is not far from zero.

2.3 The Master Sample

To estimate the number of namesakes, we need a representative master sample containing unambiguously defined individuals along with their names. Our master sample is the

stakeholder database of the German credit rating agency "creditreform". It contains 6731543 owners, managers and major shareholders of almost all German firms over a period of 15 years. It also includes a large proportion of German micro firms. Therefore, we do not expect a bias towards specific ethnical groups, which would be the case, if only larger firms were in the sample. From a demographic point of view, the master sample is not representative. For instance, only 27% of the stakeholders are female. However, we consider the data a healthy sample in regard of its representation of the variation of names.

The names are cleaned from additional clutter like academic titles and other not birth name related appendages, converted to upper case and special letters, like the German "ß" or French “âcçènts”, are replaced with the most common alphabetical letter representation. Potential target data needs to be prepared in a similar way. Because the name format greatly influences the distribution of namesakes over the minimum occurrence rank and access to the master sample, to reiterate the estimation, should not be a requirement for a disambiguation run, we conducted our analysis on the three most common name formats encountered in patent and bibliographic data:

- **Format A: last name, first name**
  Last and first name are or can be separated into two distinct fields.
  minocc is based on the minimum occurrence of the last respectively first name in population.

- **Format B: last name, initials**
  First names are represented by starting letters only.
  minocc is based on the number of initials per last name in the population.

- **Format C: unordered name**
  Last and first name are in one field without specific order.
  minocc is based on the word of a name with the lowest occurrence in the population.

In case of format C, we have to handle a methodical weakness of this specific name representation. As there is no way to differentiate between last and first name, there exists a group of outliers with a high *minocc* and relatively low number of namesakes because they have a common first name as the last name, e.g. Maria Peter. We eliminate this group from the master sample by identifying the percentile of the most common first names and removing all observations with these first names as last name. We lose 145,587 names (345,576 persons) of the aggregated master sample. For these cases, the predicted number of namesakes will be consistently overestimated, but keeping them in the sample would lead to

a general underestimation. As format A and format C both provide the non-truncated name information, format A should always be the preference if applicable. The following table shows the effect of the different name formats on the name aggregation, minimum occurrence and the number of namesakes per name:

Table 1: Master sample by name formats

			Minimum occurrence				Namesakes per name			
F	People	Names	min	max	E	sd	min	max	E	sd
A	6,731,543	4,691,779	1	7,975	105.2	230.4	1	1,118	1.43	3.56
B	6,730,633	2,544,481	1	1,264	32.2	77.5	1	5,366	2.65	15.37
C	6,385,967	4,546,192	1	27,150	192.2	570.3	1	1,035	1.40	3.35
C*	6,731,543	4,691,779	1	98,363	279.3	1264.5	1	1,118	1.43	3.56

*including outliers

2.4 Predictive Model

Our predictive model consists of a weighted Poisson regression of the number of namesakes on a polynomial of the 5th degree of *minocc*. The model takes the following form:

$$namesakes = exp\left(\beta_0 + \beta_1 minocc + \beta_2 minocc^2 + \beta_3 minocc^3 + \cdots + \beta_5 minocc^5\right) + \varepsilon \quad [pweight: namesakes] \tag{5}$$

The population weight is the number of namesakes itself, as every observation in the aggregated data is a name representing namesake persons in the master sample. Using an unweighted model would underestimate the namesakes, because the aggregation inflates the relation of unique names against namesake afflicted names. The model is equivalent to an unweighted regression of the non-aggregated data, where every observation unit is an individual represented by its number of namesakes and a name with standard errors clustered by names. As we also want to show the difference to an unweighted model, we adhere to the name-based version. Nevertheless, we do expect a proper estimator for namesake predictions of individuals represented by their names. This estimator will overestimate the population size, if we accumulate all predicted namesakes for every name in the aggregated population, as the number of namesakes is actually a property of the name aggregate and not of an individual. This is an issue we need to address before we can calculate unit sizes.

Because the interpretation of high-degree polynomials is not very intuitive and due to the univariate design of the model, we discuss the results of the regressions based on figures showing scatter plots of namesakes on *minocc* overlaid by predicted namesakes (weighted and unweighted) for all three formats. We present the actual regression results in Appendix B, Table 4.

Figure 2: Predicted namesakes (weighted vs. unweighted) on scatter plot: Format A

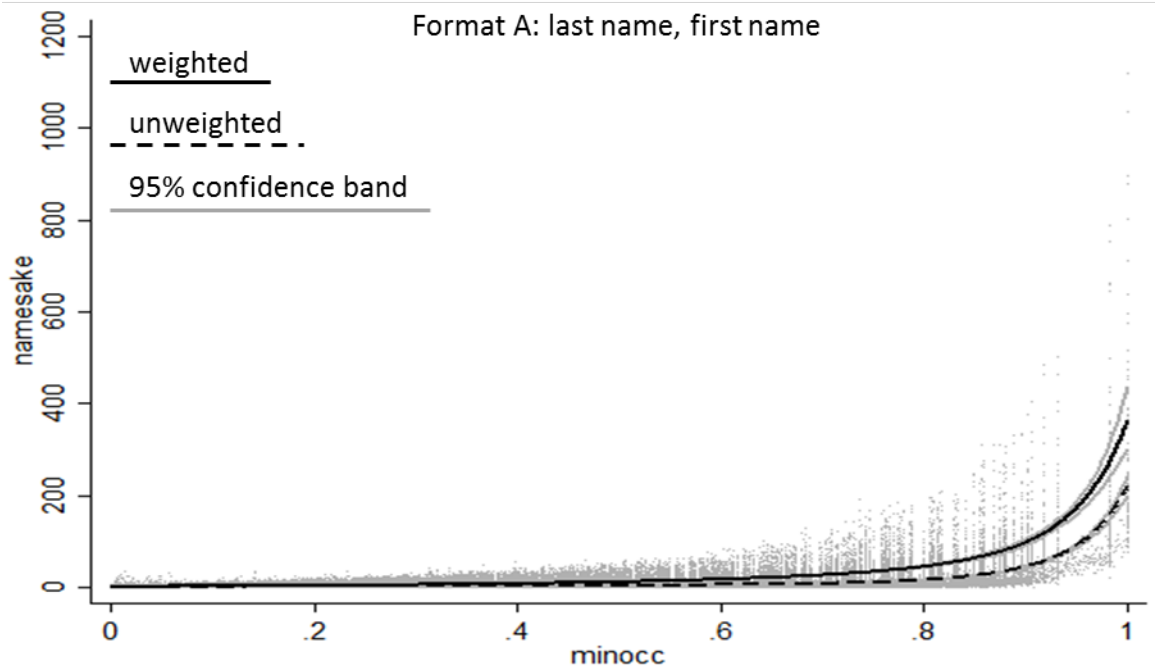


Figure 3: Predicted namesakes (weighted vs. unweighted) on scatter plot: Format B

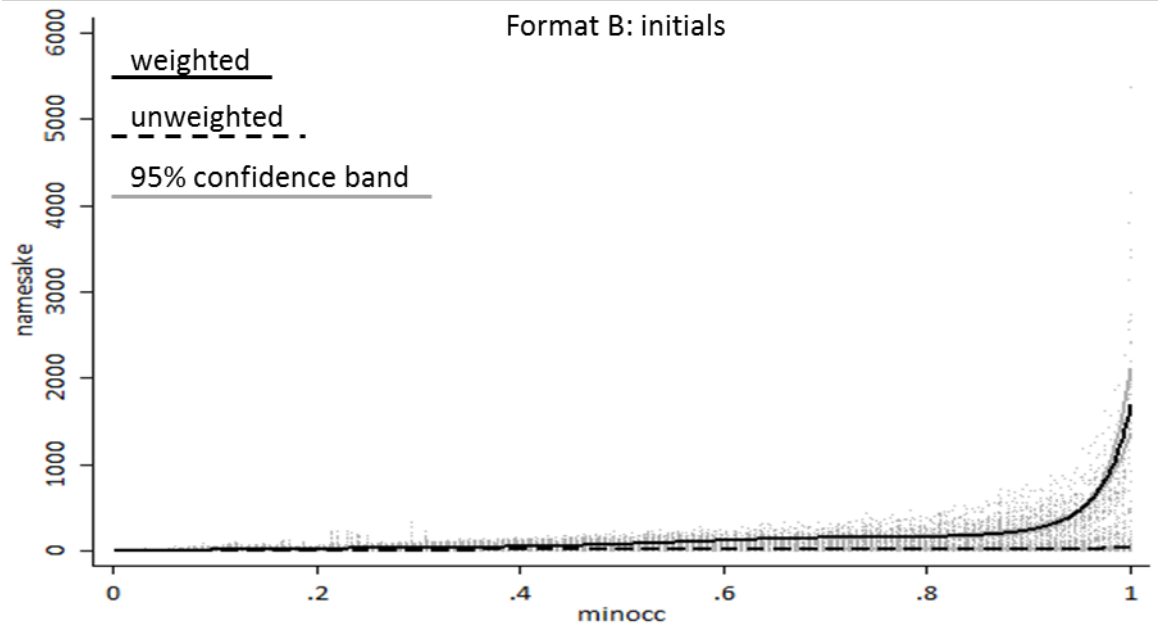
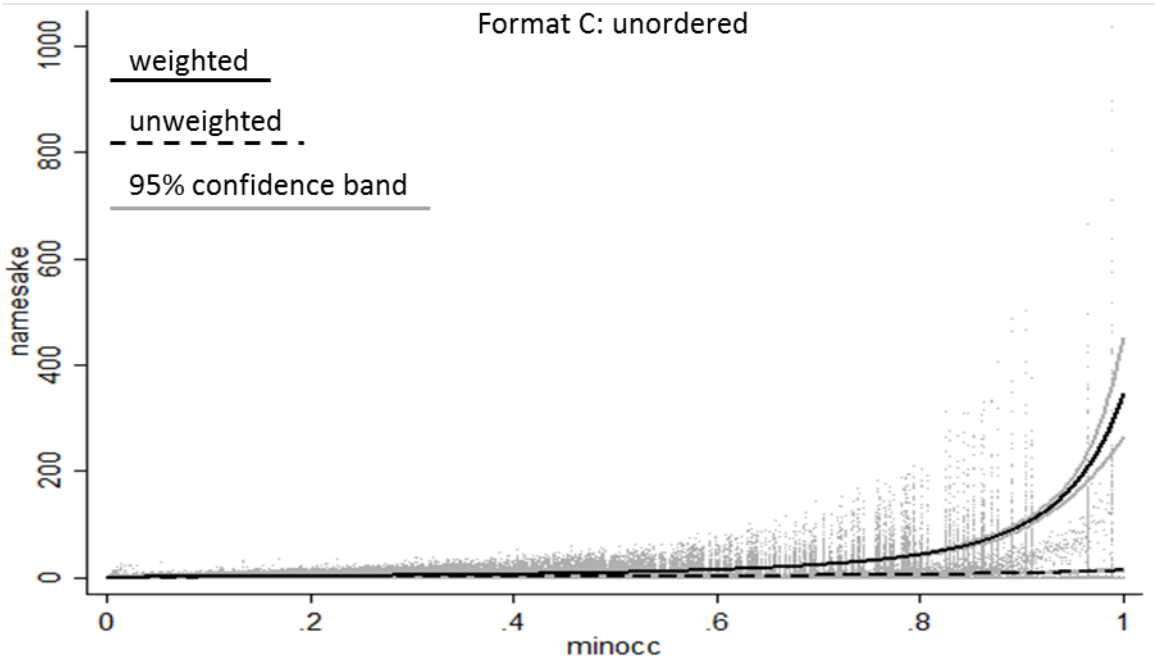


Figure 4: Predicted namesakes (weighted vs. unweighted) on scatter plot: Format C



The graph for format A already shows an unweighted curve that follows the rising of namesakes with higher values of *minocc*. Both variables are positively correlated. We observe only a negligible number of outliers with a high *minocc* but a low number of namesakes. The other formats display a much larger discrepancy between the two curves. The unweighted prediction is forced to the bottom on the right side of the graph by a relatively high number of names seemingly violating our key assumption. However, the weighted curve tells us, that rare names have a lower individual support, i.e. namesakes, than common names. Even though the R^2 for format B is the highest of all formats, we consider the predictive power being the lowest, because of the high information loss by using initials instead of first names. In any case, the risk of encountering a namesake afflicted individual is larger for higher values of *minocc*. Figure 8 in the Appendix shows the graph for the unmodified format C (including outliers).

2.5 Representativeness

Our prediction of namesakes is based on the master sample and scaled in relation to this specific population. The target data may be considerably larger or smaller than the master sample, raising the question, if the size of the population N and the estimated number of namesake n has to be adjusted accordingly. Most target data, e.g. patent data of a specific

office, have usually smaller populations in regard of names and individuals as the master sample. They are units which are constantly growing, fueled by an incessant stream of new entrants drawn from a pool represented by the master sample. Of course, the same can be said about the master sample, which feeds from the real population consisting of all individuals eligible for doing business in Germany. The actual question is: can the master sample represent this elusive pool. This is the case when the share of namesakes grows proportionate with the size of the virtual pool. In such a scenario, we could adjust both parameters N and n with a scaling factor f. In the following paragraph we show that such a factor will dissipate.

We start with the straightforward implementation of the namesake probability based on the products of the stepwise probabilities (1). The parallel decrement of the numerator and denominator has only a minimal impact on the result. It is only relevant for already small values, which only occur for s or n being of the same magnitude as N, a highly unlikely situation. Therefore, we approximate the namesake probability with

$$P(namesake) \approx 1 - \left(\frac{N-n}{N}\right)^{s-1} \tag{6}$$

by replacing the stepwise probabilities with a constant probability. When we adjust our approximation by plugging in a scaling factor f for N and n we get

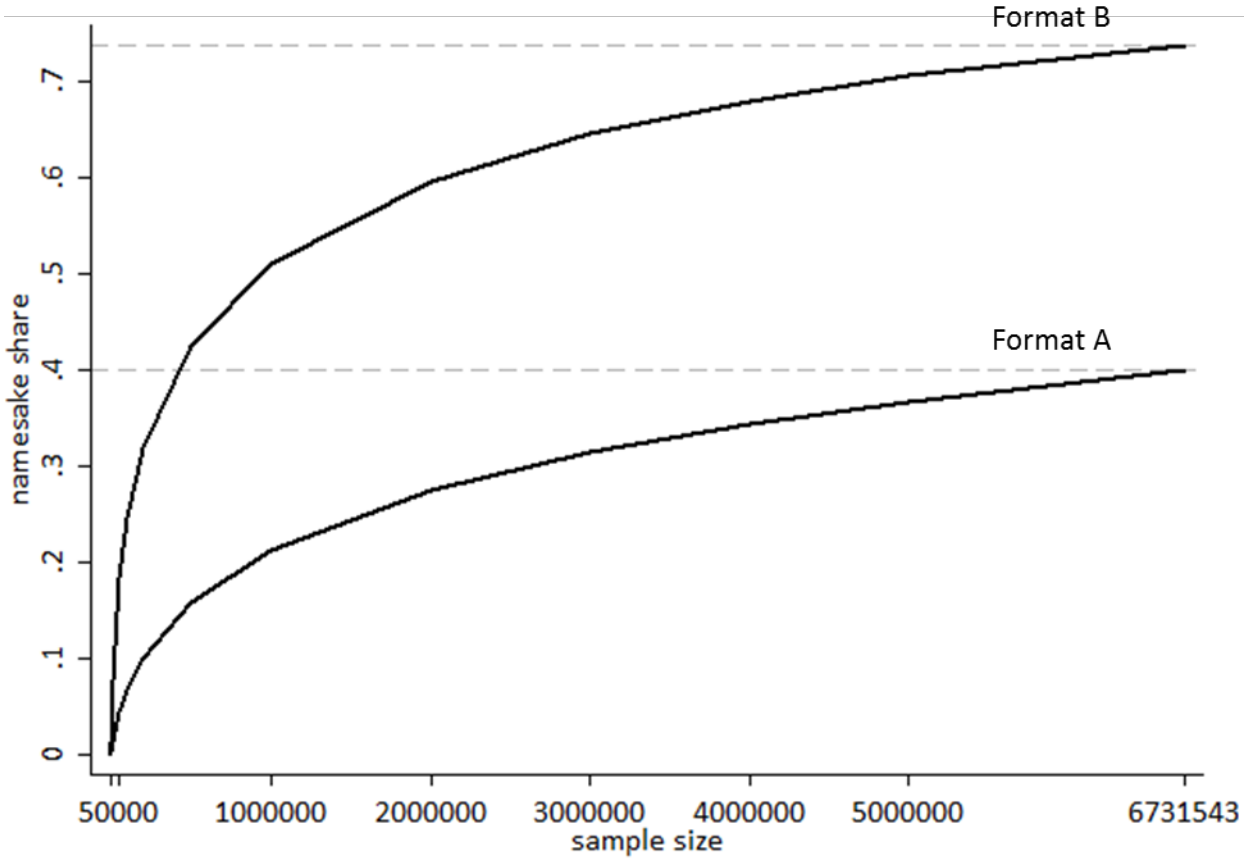
$$P_f(namesake) \approx 1 - \left(\frac{fN-fn}{fN}\right)^{s-1} = 1 - \left(\frac{N-n}{N}\right)^{s-1} \tag{7}$$

and witness the elimination of the scaling factor. This relieves us from the issue of adjustment.

To determine if our master sample is large enough to provide this linear relation, we conduct a Monte Carlo experiment. Figure 5 depicts the results of this experiment. For a selection of sample sizes between 100 and 5 Million individuals, we repeated a random draw without replacement from the master sample 200 times for every sample size. The master sample is replenished after every sample draw. We record the number of namesake afflicted individuals for every draw to calculate the average share of namesakes per sample size. For the name format A (last name, first name), we observe 40% of namesake afflicted individuals in the

master sample. This number rises to 74% for the denser aggregate on initials and last name (format B). The curves show a steep catchup of the namesake share consolidating in an almost linear progression as the name population is exhausted with larger sample sizes. This close linearity supports our confidence in the representativeness of the master sample.

Figure 5: Share of namesakes in relation to sample size



Note: Confidence intervals are too narrow to be visible on this scale.

This figure additionally unfolds that in a considerably smaller sample the share of namesakes would have been irredeemably underestimated. A fate that is inherent in most training or benchmarking data causing the namesake bias discussed in Chapter 3.

There is a caveat pertaining the western origin of the master sample. Even if we assume that the *minocc* normalization is able to capture the relative shape of other populations, dominated by different naming conventions, the same cannot be said about the absolute namesake predictions. A *minocc* of 1 returns a namesake prediction of 362. A number which clearly is too low for Asian name populations. The best solution is to provide specialized master samples for different naming cultures. This will not be feasible in many cases, therefor

we suggest to transform the predicted curves as shown in Figure 2 to Figure 4 into segmented gradients to create a flexible prediction based on the interpolation between linchpins. The right tail of this artificial curve can be raised, increasing the estate below it, to accommodate denser name populations or to reduce the general risk of false positives. Alas, this is one of many arbitrary interventions imposed by the incompleteness of the available information we will discuss in Section 4.4.

2.6 Unit Sizes

To assess the risk of a namesake, we need the number of namesakes n in the population N and the unit size s. The parameter n can be estimated using the normalized minimum occurrence based on the target data. The estimate is scaled in relation to the population size N of the master sample needing no adjustment as shown in the previous section. A preliminary proxy $\check{s}$ is defined by the size of the name aggregate of the unit, for instance, the number of distinct inventor names appearing in the patent documents of a specific applicant. We call this set of names U. The number of names in U is always underestimating the real unit size because of namesakes. We receive the augmented proxy $\hat{s}$ by accumulating the estimated number of namesakes in U in respect to a unit with size $\check{s}$:

$$\hat{s} = \check{s} + \delta(\check{s} - 1) \sum_{i \epsilon U} 1 - \frac{N - \hat{n}_i}{N - 1} \tag{8}$$

The probabilities of drawing a namesake on the first step in the probability tree (see Figure 1) for all involved names are summarized and multiplied by the number of steps required to reach unit size $\check{s}$. The size $\hat{s}$ is the number of expected additional individuals because of namesakes plus the number of names $\check{s}$, as every name in U already represents at least one individual. We already know, that by using the predicted number of namesakes $\hat{n}$, we will overestimate the population on the name level. This positive bias is challenged by the negative of using the name count $\check{s}$ as a lower-bound proxy for s. In addition, we use the probability of the first step although there is a very slight incremental shift in the probability of drawing a namesake by going further down the tree. Finally, we have to take the fragmentation of the population into account. For small units the proxy $\check{s}$ is closer to the real size than for large units, because the probability of encountering namesake-afflicted names increases with the size and therefore the difference to the name aggregate. To balance these contradictory

effects, we introduce the parameter δ, which is retrieved from a Monte Carlo experiment simulating a fragmented population.

First, we separate the randomly sorted master sample into virtual units with the following equally distributed and randomly chosen sizes: 10, 20, 100, 500, 1000, 2500, 5000, 10000, 50000 and 200000. We aggregate the data on the name and unit level, keeping the actual size s as a reference. We repeat this process 15 times, appending the resulting data to get a virtual, fragmented population. For every unit, we calculate the improved proxy for the unit size $\hat{s}$ without the balancing coefficient ($\delta = 1$) using equation (1.8). After aggregating the data on unit level, we regress the real number of additional individuals by namesakes on the estimated number:

$$s - \check{s} = \delta(\hat{s} - \check{s}) \tag{9}$$

We omit the intercept to force the slope through the origin. The level of fragmentation is an arbitrary choice mimicking the natural separation of a population into units. The actual fragmentation of a population depends on the context. The context "applicant" generates a different fragmentation than the context "technological classification". As we also have to consider combinations of contexts, which create additional layers of fragmentation, we decided to tackle this issue by a generous mixture of unit sizes.

Table 2 shows the improvement of the preliminary proxy $\check{s}$ to the balanced estimate $\hat{s}$ by applying equation (8) based on the virtual population derived from the master sample for format A and format B. The need to include the estimated namesakes increases with the density of the name aggregate. Format B has a higher density, meaning less name variation, whereby the degree of the bias for the unmodified proxy is exacerbated compared to format A. For the latter format, the additional effort shows a smaller improvement, but it is still justified by the robustness gain against outliers in regard of the unit composition, i.e. units with a high share of common or rare names. Further, the upper half of the table alludes to the issue that the share of namesakes within a unit is not a linear function of the unit size, a circumstance leading to the namesake bias introduced by unbalanced training respectively benchmarking data. In section 3, we explain why this bias is almost unavoidable but, fortunately, not affecting our disambiguation approach.

Table 2: Estimated unit sizes for format A and format B

	last name, first name					last name, initials				
s	**N**	**min**	**max**	$\boldsymbol{E(\check{s})}$	$\boldsymbol{\sigma}$	**N**	**min**	**max**	$\boldsymbol{E(\check{s})}$	$\boldsymbol{\sigma}$
20	285	20	20	20.00	0.00	271	19	20	19.99	0.09
100	263	99	100	100.00	0.06	248	99	100	99.93	0.26
1000	275	995	1000	999.32	0.87	288	986	1000	993.53	2.71
50000	304	48696	48896	48791	36.64	258	43816	44200	43999	74.20
200000	259	187175	187801	187485	124.0	282	153909	155012	154469	186.0
	$\boldsymbol{\delta = 0.487103\ (0.00031)}$					$\boldsymbol{\delta = 0.444869\ (0.00438)}$				
s	**N**	**min**	**max**	$\boldsymbol{E(\hat{s})}$	$\boldsymbol{\sigma}$	**N**	**min**	**max**	$\boldsymbol{E(\hat{s})}$	$\boldsymbol{\sigma}$
20	285	20	20	20.00	0.00	271	19	20	20.00	0.09
100	263	99	100	100.00	0.06	248	99	100	100.00	0.26
1000	275	996	1001	1000.00	0.87	288	991	1006	999.13	2.74
50000	304	49889	50090	49987	38.31	258	49176	49800	49423	105.9
200000	259	199631	200330	200007	131.3	282	199131	201966	200365	469.8

Note: cluster robust standard errors in parentheses

3 Namesake bias

The need for reliable benchmark respectively training datasets always accompanies the development of the various disambiguation efforts. These datasets are not only used to compare the performance of different approaches, but also to tune the parameters of the algorithms to produce the desired outcome: improving precision while maintaining a high recall rate. These goals cannot be maximized independently as this would lead to conflicting solutions, i.e. deeming all names unique minimizes the number of false negatives while maximizing the number of false positives. Researchers use training dataset to adjust the weights of matching criteria and algorithm specific parameters in a multitude of ways to balance both goals. There exists several benchmark datasets like the Benchmark Israeli Inventors Set (BIIS) (Trajtenberg et. Al., 2008), the Noise Added French Academe (NAFA) and Noise Added EPFL datasets (NAE) (Lissoni et. Al., 2010), which are publicly available. These datasets trace the careers of individual inventors by their output. As the personal inquiry of this information is an expensive process, the samples are often not randomly drawn but

chosen by ease of access. This by itself can already introduce a bias caused by clustering of similar career profiles. Although, the more concerning bias is systematically inherent in the fact that the samples are based on individuals and not on names.

Magerman (2015) criticized benchmark datasets in general for severely underrepresenting careers of homonymous researchers and therefore not being exhaustive. Even one of the largest datasets, the "E&S" labeled dataset (Chunmian, Ke-Wei, Ping, 2016), linking 96,104 patents to 14,293 inventors, contains only 10 homonymous cases, a circumstance implying the deliberate selection of uncommon inventor names to reduce workload. Unfortunately, these datasets are not suited as training data because they concentrate on identifying the careers of individual authors or inventors and not on *namespaces*, the complete manifestation of all careers sharing a specific name in the data. The Monte Carlo experiment, outlined in Table 2, simulates this drawing process. In the top half, we can see that the risk of encountering a namesake is usually very low for small units and not representative to the whole population. Even a very common name may appear unique in a relatively small sample, not reflecting the need of disentangling multiple individuals sharing that name in the whole population.

Observing the lower bound name proxy for unit size $\check{s}$ in Table 2 culminating in the final name aggregate of the population, as seen in Table 1, clearly shows the non-linearity of the relation between a sample size and the encountered namesakes. If we revisit Figure 5 in Section 2.5 we realize how far even the largest training data available is apart from representativeness depicted by the dotted lines. This systematic underrepresentation of namesakes in a sample stems from the fact that the property "namesake" is only defined in the name aggregate and not on the individual level, requiring at least two randomly drawn individuals with the same name in the sample to be identified as such. At the beginning of a sequential drawing procedure, this conditional probability is much smaller than the probability of drawing an individual with a fresh or unique name but increases with the exhaustion of the name population. Hence, any sample size will lead to an underestimation of the real namesake distribution. Of course, this is also true for the master sample, but, as we have shown in Section 2.5, it is large enough that the relation between namesakes and population size became almost linear and hence neutral.

All algorithms exploiting training data based on random or selective draws of individuals suffer the namesake bias. They will persistently underestimate the risk of encountering namesakes up to the point where other matching criteria beyond the name itself become pointless, because names are perceived as reliable unique keys. This is especially true for training data deliberately constructed from uncommon names to save the effort of validating document links. The namesake bias does not affect training data based on the semi-random draw of document tuples as long as both documents belong to the same namespace and there is no selection preferring uncommon names to minimize the effort of validation.

A training dataset providing a robust framework for algorithms should be based on a two- step procedure: First, a representative selection of names has to be determined by the name aggregation of a reasonably sized draw of individuals or, if that is not possible, documents. Second, the complete disambiguation of all careers manifested by documents bearing the drawn names. Unfortunately, the verification by surveying all the authors or inventors sharing a specific name is an impossible task for obvious reasons, i.e. deceased authors, language barriers, obsolete or insufficient addresses. One could argue that identification on the personal level is not required, as this would include information based on confounding parameters having no representation in the data. Even then, the creation of such a dataset is an enormous task requiring coordinated action of several teams to install an overlapping monitoring system to prevent biases.

Even a perfectly balanced training dataset is only valid for the parametrization for one specific bibliometric database. The transferability of these parameters onto other databases assumes a high level of compatibility. The method of assessing the risk of encountering a namesake for every single document link does not require a training dataset and is therefore by definition whether bound to a specific database nor affected by the namesake bias. It rather embraces the concept of namesakes by dynamically adjusting its parameters instead of relying on a predetermined statistical average. We will see in Chapter 4, discussing the implementation, that the algorithm is still not free from arbitrary decisions, typically permeating most heuristic approaches. Nevertheless, this is a small price to pay given the advantages of not requiring training data, which, as a side effect, allows rapid deployment of the method on any person related bibliometric database.

4 Implementation

The representation of documents like patents or scientific publications in bibliographic databases accessible to researchers usually does not include the full document itself. It concentrates mainly on bibliometric properties to support the retrieval of documents by their authors or inventors, affiliations, locations of the aforementioned, keywords and topics, titles, classifications, journals, date of publishing and so on. To identify the documents of a specific person, one would first search for the name of the person. If the name is exotic, the result of the search already portraits the career of the person. For a common name, it is required to supplement the search with additional information about the person, for example the name of a co-author or an affiliation. Whether the found documents belong to the person of interest or are from a namesake depends on the commonness of the name and the identification potential of the additional information. If the co-author has an exotic name or the affiliation is only a small company, the likelihood of getting the wrong person is small. Obviously, the identification potential corresponds with the perceived size of the search criteria relating to the peer group of authors or inventors. The search results of the first step reveal new document properties to be included in further-reaching queries, leading to new documents to be subject of namesake risk assessment and, again, the retrieval of new search criteria. A good depiction of this recursive procedure is a network analogy, where the documents are nodes connected by mutual properties. The searcher traverses along the edges from node to node, collecting all touched nodes into a cluster list. The accessibility of an edge depends on whether the risk of connecting documents of namesakes is below a general threshold. To assess the risk, the searcher has to estimate the number of namesakes for the given name and the unit size. The latter is determined by intersecting the peer groups of the connecting mutual properties. The disambiguation algorithm separates the network, spread out by mutual properties of documents sharing a specific name, into clusters with a low risk of containing the work of namesakes.

4.1 Reducing Complexity

We classify document properties into two different kinds. *Hard properties* have no variation in regard of the entity they designate. Categorical memberships of documents like standardized technological classifications, research field categories or unique identifiers like cited patent numbers are hard properties. We consider properties whose variation can be

eliminated by trivial cleaning procedures, like removing non-numerical characters or transferring all characters to upper case, as hard. *Soft properties* have no trivial to eliminate variation in regard of the entity they designate. They require the usage of the adjective "similar" to describe their relation, e.g. similar inventor name, similar affiliation, similar applicant, similar topics in titles and so on. There is a high variation in the portrayal of the same entity in bibliographic or patent data because the focus of the managing institutions is the proper representation of documents but not the administration of specific databases to harmonize all properties. Besides the identification of entities within the data, we are also interested in detecting similarity in descriptive properties like titles or keyword lists sharing a specific topic. The goal is to identify cluster IDs for variants of the same entity or topic for all properties. We further call these cluster IDs *traits* of a document.

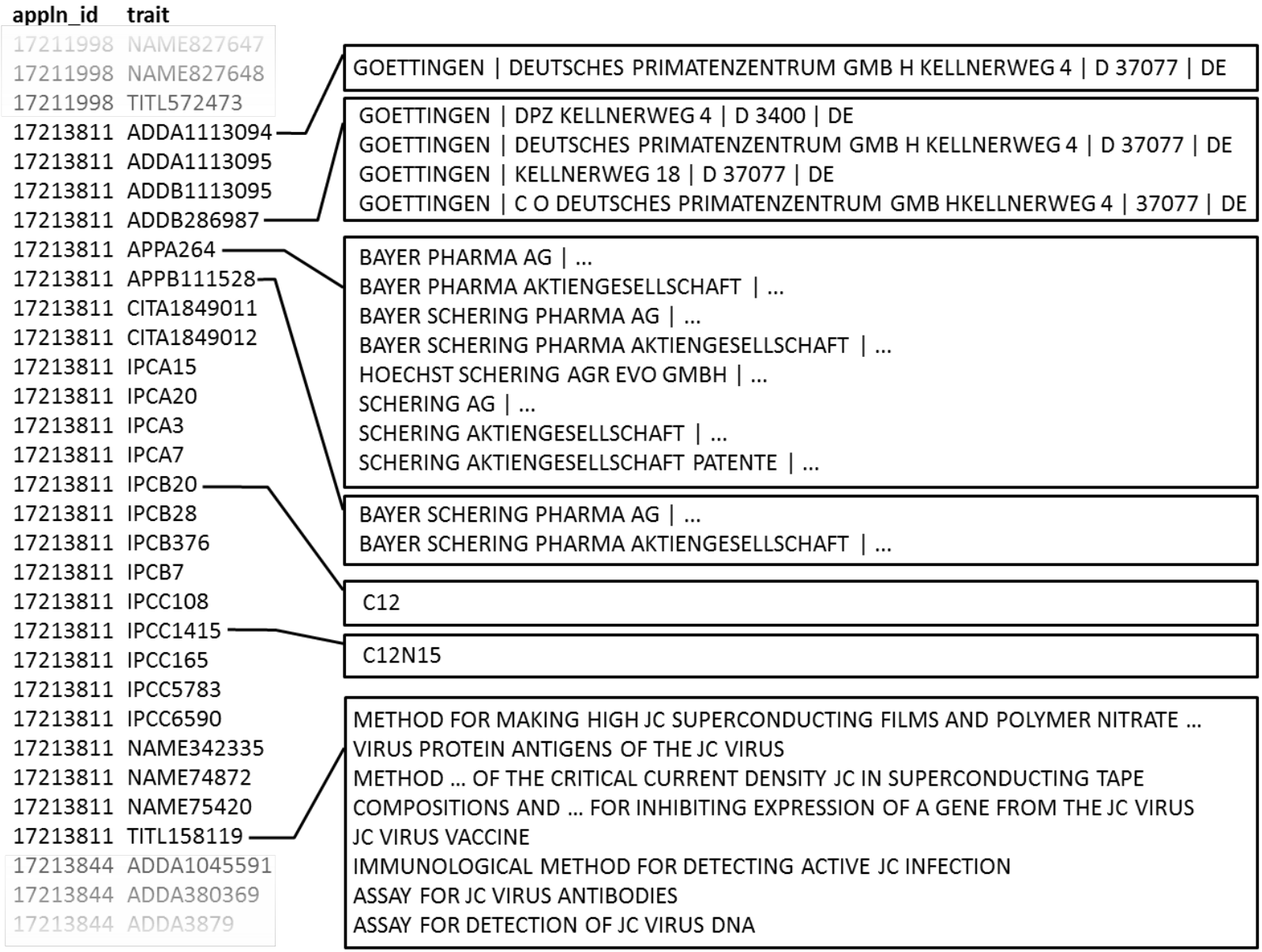
The clustering of soft properties, especially of topics, is a complex endeavor. Even though we have found a feasible solution, we do not promote it as a gold standard. It yields good results for entity clustering but has shortcomings in topic clustering, where more contemporary methods like LDA, Doc2Vec or Word Embedding, to name a few, prevail. A considerable disadvantage of the namesake risk assessment is that mere distance calculations between properties are not sufficient as distances do not provide a peer group. Only trait clusters encompass a set of individuals necessary to calculate a unit size. Besides measuring distances, methods to cluster those into meaningful entities and topics are mandatory. We abridge this discussion here and refer to a description of our method called "Nested Cascaded Traversal of Intransitive Similarity Networks" in Appendix A (includes a link to the used program).

After creating the clusters of the soft properties and the recoding of the hard poperies, we consolidate all cluster IDs into a single table. The *trait vector* associates the document IDs with the respective traits of the documents. A trait is a key composed of a prefix designating the context and a cluster ID. A complex relational database becomes a simple vector of tuples.

For a better understanding, the following paragraph describes in full detail an excerpt of the trait vector we constructed for the EPO patents, shown in Figure 6. The traits with the prefix NAME refer to three different inventors. The patent has two citations designated by the prefix CITA. The choice of the prefixes indicates that we conducted multiple cluster formations for the different soft properties and for the hard property IPC (International Patent Classification).

We create three different aggregation levels for the IPC by truncation at coherent positions prefixed by IPCA, IPCB and IPCC. The inventor address prefixes ADDA and ADDB and applicant prefixes APPA and APPB are based on different cluster building cascades. The trait ADDA1113094 refers to only one address, but cluster ADDB286987 reveals that there are actually four different variants in the data describing this specific location. Because the cascade definitions for the inventor addresses are complementary, it is not necessarily the case that ADDB always returns a stricter defined cluster than ADDA. The applicant cluster APPA264 is a typical representation of name changes during mergers. The pharmaceutical company “Schering” was bought by “Bayer” to become a part of the “Bayer Pharma” group. Before that merger, “Schering” and “Höchst” had a joint venture, called “Höchst Schering AGR EVO Gmbh”, to join their crop protection divisions. The whole process is traceable because the partial overlapping of the names creates intransitive links between these applicants. For reasons of clarity, addresses are not listed. The stricter APPB cluster returns only the name variants close to the time of invention. Finally, the title cluster TITL158119 shows some incoherent entries. Obviously, the term “JC” plays a defining role for two completely different technologies, once as a virus and as a component for superconducting tapes. These incoherent clusters do not pose a high risk, as the probability of joining namesakes by a cluster representing such a small peer group, aka unit size, is marginal at worst. On the other hand, a cluster that completely got out of bounds is hedged by the fact that the inflated unit size automatically increases the calculated probability for a namesake, therefore mitigating false positives.

Figure 6: Excerpt from the trait vector for the EPO data



4.2 Mutual Traits

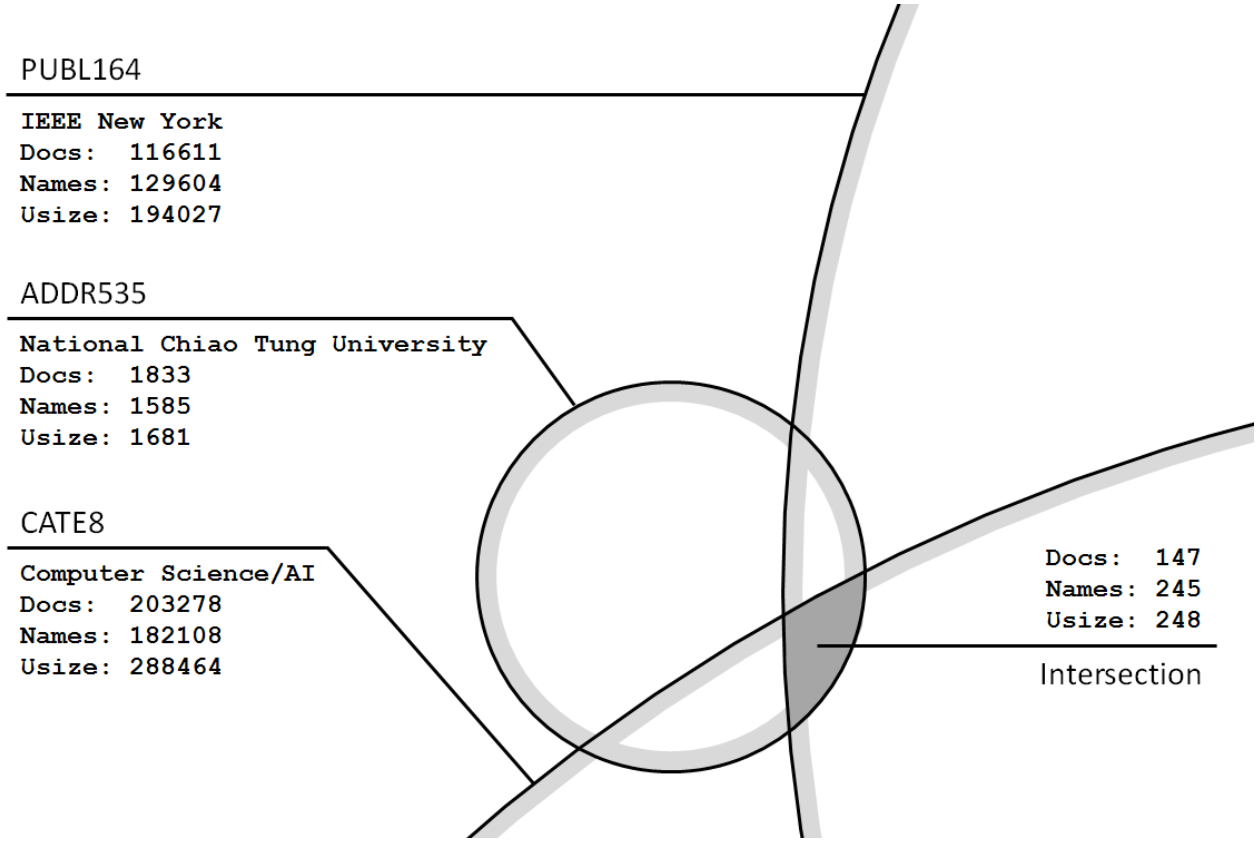
The trait vector reduces the complexity of documents as bags of words, where similarities are obfuscated by noisy variation, into a simple collection of traits. The otherwise comprehensive task of identifying common properties between documents transforms into a trivial SQL statement. Another prerequisite for the disambiguation algorithm is the calculation of the estimated number of namesakes for every author respectively inventor name encountered in the target data. A name is defined as the cluster designating the name in the trait vector, i.e. NAME74872 (see Figure 6). As a name cluster may contain several variants because of misspellings, positional variation and so on, only the maximum of the minimum occurrences within every name cluster is ranked and normalized into the *minocc*, ready to be plugged into the prediction equation (5).

The algorithm sequentially creates enclosed networks of linked documents for every name cluster. We call such an enclosed network a *namespace*. A link between two documents is defined by the mutual traits of these documents. Of course, the respective name trait is excluded from the mutual traits as this would lead to a completely connected network. The exclusion has to be exerted for all contextually related traits, like author, inventor or applicant, bearing the risk of tautological links, i.e. linking all documents where the inventor is also the applicant within the namespace. Independent careers appear already as separated sub-graphs. However, all sub-graphs need to be scrutinized for potential breaking lines. With the number of namesakes n and the population size N already in place, we only need the unit size s to assess the risk of a link connecting documents of two different persons sharing the same name. For every mutual trait of a link, we collect the document IDs in the trait vector sharing this trait. We intersect the resulting documents by means of simple SQL joins. The two originating documents are always in the center of the final intersection. The next step is the identification and aggregation of the involved names to get the preliminary proxy $\check{s}$ by selecting all name traits of these documents in the trait vector. Now we have all parameters required to resolve equation (8). With the approximated parameter $\hat{s}$, replacing unit size s in equation (4), we can assess the risk of having a namesake for the given name in a peer group of authors or inventors whose profiles match the mutual traits of the two documents. If the risk is above an arbitrarily defined threshold, the link will be destroyed. We repeat this process

for every link in the network. Finally, we just need to traverse the remaining links of the namespace to identify the document clusters representing specific author or inventor careers.

Figure 7 shows an intersection of mutual traits from a trait vector based on all documents of the research area “Computer Science” in the “Web of Science” database. The areas of the circles (or circular segments) representing the three traits ADDR535, PUBL164 and CATE8 are in proportion to the respective unit sizes. The example illustrates the extremely high computational effort to determine the unit size for a single combination of mutual traits. Intersecting trait PUBL164 with trait CATE8 requires 116611 comparison operations. Starting the sequence with trait ADDR535, the first intersection costs only 1833 operations with a decreasing amount for subsequent intersections. Being the central part of the algorithm, requiring most of the computational resources, improving the performance of the intersection procedure is a worthwhile endeavor.

Figure 7: Example of an intersection of mutual traits



4.3 Optimization

The magnitude of the document count of weak traits calls for a more sensible approach than intersecting mutual traits in a random order. Starting with a trait appearing in hundreds of thousands documents significantly slows down the whole process. For that reason, we introduce an additional vector table called *meta vector*. It contains all traits of the trait vector with already calculated unit sizes. Joining mutual traits with this vector allows for efficient ordering of the intersection sequence. Having direct access to the actual unit sizes of the mutual traits may even lead to skipping the intersection effort altogether, if the unit size of the smallest trait already returns a namesake probability equal or below the threshold.

Parallel runs of different approaches have shown that calculating the namesake risk after every intersection, requiring the aggregation of the intersected documents on the name level, is still faster than always conducting the complete intersection sequence. The linkage between the names and their corresponding number of namesakes to calculate the unit size $\hat{s}$ by equation (8) is only required if the provisional unit size $\check{s}$, defined by the size of the name aggregate, already returns a preliminary namesake risk equal or below the threshold.

Furthermore, there is a high level of repetition among the links of a namespace. Combinations of mutual traits repeat themselves within the network because of the state dependency found in most inventor or author careers and because of weak links based on combinations of common categories. Identifying the different mutual trait combinations within a namespace reduces the frequency of mutual trait evaluations to the number of combinations. For instance, on average a namespace within the EPO data consists of 123 links based on only 17 combinations.

We also apply two derivations, reasoned by simple set theory, on the intersection procedure: First, if we have completely intersected a combination, but the resulting unit size is still too large in regard of the namesake risk threshold, all remaining combinations that are a subset of the unsuccessful combination are also invalid. Second, if a unit size emerges during the intersection sequence that is small enough to satisfy the namesake risk threshold, all remaining combinations that are a superset of the successful combination are also valid. In both cases, we skip the evaluation of the indirectly rated combinations.

All these optimizations only affect the actual namespace. To convey already made efforts beyond the actual namespace, we introduce a *shortcut table* containing all already assessed combinations and the associated unit sizes. Every combination is represented by a string of traits concatenated in a fixed order. Only new combinations have to be evaluated to end up as another shortcut record in this table.

Finally, the separation into namespaces is a textbook example for applicability of parallelization. A CPU process cycle consists of looking for a free namespace in a namespace registry and reserving it for disambiguation. Contemporary computer systems allow for multiple parallel processes. The only bottleneck is the collective access on key tables like the trait and the meta vector. Every process has its own shortcut table to prevent further accessibility conflicts. The following list summarizes all implemented optimizations:

- Preparatory sorting of mutual traits by unit size using the meta vector.
  → Intersections start already small.
- Intersecting stops early when risk of namesakes is equal or below threshold.
  → Not all mutual traits have to be intersected.
- Identification of mutual trait combinations defining the links in the namespace
  → Number of combinations is much smaller than the number of links.
- If a completely intersected unit size is still too large, all remaining combinations that are a subset of the unsuccessful combination are also invalid.
- If a valid unit size is intersected, all remaining combinations that are a superset of the successful combination are also valid.
- Saving of evaluated combinations and associated unit sizes in a shortcut table prevents repetition of already made efforts beyond the actual name space.
- Separation into name spaces allows for a simple separation of the workload for a multiprocessing approach.

4.4 Parameters

The key advantage of this approach is the independency of training data. No sample of disambiguated document sets has to be created, be it by common sense assessment, surveys on authors or inventors or by exploiting existing identification keys like ORCHID. Of course, having such a reference group can provide a guideline to improve the settings of the algorithm, which have emerged during the development phase. Until now, we have only

discussed the threshold for the namesake risk as the only parameter. A threshold of 5% seems acceptable for any link between two documents, but by the very nature of the intransitive networks defining the namespaces, the risk of falsely linking separate individuals accumulates, leading to the intransitivity "conundrum" mentioned by Trajtenberg et.al. (2006). A lower threshold alleviates this issue at the cost of increasing the amount of false career splits especially within namespaces of common names. Inspection of these large namespaces has shown that the culprit are in most cases rarely used classifications issued by external authorities and not by the creators of the document. Hence, it is possible to designate specific trait prefixes as supplemental only, if the estimated number of namesakes exceeds an arbitrary limit defining the upper bound of a "small" namespace, e.g. 10. Supplemental traits are used to intersect the unit size, but never define a link exclusively. Besides downgrading specific types of traits, it is also possible to declare trustworthy trait prefixes. Links between documents also based on trustworthy traits enjoy a relaxed namesake risk threshold, if the temporal difference between the documents filing respectively publishing date is below a limit, e.g. 4 years. This feature introduces a time component, which otherwise is difficult to translate into a trait. Finally, it is possible to set a lower bound for the number of namesakes to prevent that namespaces with a very low namesake estimate always are bundled into one cluster regardless of unit sizes. This is especially advised when the target data is relatively small compared to the master sample resulting in an overrepresentation of unique names and a skewed *minocc* distribution.

Finally, the Western origin of the master sample is not capable to capture the density of other naming cultures, e.g. of Asian heritage. We need to find a way to simulate an estimate based on denser name populations. To achieve this, we create a gradient chain based on the deltas of 100 equidistant readings of the original estimate. Any prediction can be interpolated by accumulating the gradients to get the linchpins before and after the requested *minocc* value. By applying a factor to the gradients, the area under the curve can be inflated, simulating the namesake estimate of a denser name population. The factor determines the maximum namesake prediction at *minocc* = 1 (362 for the original estimate, factor = 1). Of course, it is more efficient to replace the gradients directly with the linchpins after applying the factor to spare the accumulations for further predictions. This discretized list of points replaces the prediction by equation (5). Using an inflated namesake prediction is not only beneficial to

emulate other naming realms but also as means to reduce false positives in the case of a highly sensitive disambiguation project.

5 Applications and Conclusion

We have learned that missing or ambiguous information leads to the necessity of arbitrary countermeasures constituting heuristic methods in principal. Of course, having a proper training dataset to tune those parameters is much more convenient than relying on intuition. Unfortunately, as shown in Chapter 3, this sentiment may lead to an involuntarily introduced namesake bias. A simple test based on the combined Noise Added French Academe and the Noise Added EPFL datasets illustrates this conflict. We disambiguate the EPO data using our elaborate approach to achieve a recall of 94% and a precision of 99%. However, if we pretend, that every name is unique and namesakes does not exist, we still end up with a recall of 97% and a precision of 97%. Apparently, a small sample of 517 individuals, already containing noise in the form of random namesakes to the properly identified inventors, is not sufficient to represent the population in regard of namesakes. However, even significantly larger benchmark datasets, not based on namespace exhaustion, may produce similar results due to the inherent namesake bias.

We applied the algorithm to the data of three major patent offices: EPO, USPTO and JPO. For the EPO and the USPTO, we can rely on original data sources provided by the offices. The Japanese data is obtained from the Patstat, a worldwide patent data repository maintained by the EPO. All data sources were released in 2015. We define traits based on inventor names, inventor addresses, applicant names and addresses, title topic clusters, forward and backward citations and international patent classifications (IPC). All soft properties have two context prefixes representing a strict and a more lenient clustering. The trait vector contains three aggregation levels of the IPC: class level (length 3), sub class level (length 4) and group level (delimited by slash). Table 3 shows the results and settings of our disambiguation efforts.

Table 3: Disambiguation results for three major patent offices

Office	**EPO**	**USPTO**	**JPO**
Source	EPO 2015	USPTO 2015	Patstat 2015
Patents	2,796,553	5,282,235	10,625,369
Names	1,872,103	2,603,181	1,963,483
Inventors	2,382,035	3,295,523	4,004,029
Inventors/Names	1.27	1.27	2.04
Patents/Inventors	1.17	1.60	2.65
Crossing Borders	46,033	67,635	low data quality
Threshold	$2.5\%\|\Delta t > 3y$ $10\%\|\Delta t \leq 3y$	$2.5\%\|\Delta t > 3y$ $10\%\|\Delta t \leq 3y$	$1\%\|\Delta t > 2y$ $5\%\|\Delta t \leq 2y$
Lower Bound	5	5	5

For the EPO and USPTO the main settings are equal. We declare IPC traits as supplemental if the estimated number of namesakes exceeds 10. For every namespace, we enforce at least 5 namesakes as the lower bound. The default threshold is 2.5% respectively 10% if we consider the mutual trait combination as trustworthy. That is the case, if it contains a trait other than an IPC and the filing dates are not more than 3 years apart. Given the expected higher density of namesakes in the Japanese data, we have to adjust the settings accordingly by reducing the limit of the validity of IPC exclusivity to 5 namesakes and demanding a lower namesake risk threshold of 1% in the default case and 5% in the trustworthy case, which additionally has a shorter time window of 2 years[1].

The line "Names" designates the name aggregate of the respective population. For example, there are around 1.8 Million distinct inventor names in the EPO data. A slight name clustering to handle misspellings already curtails this number. It would also be the total number of inventor careers given the naïve assumption of name uniqueness. The "Inventors" line shows the count of disambiguated individual inventors. Both, the USPTO and the EPO have a similar

[1] At the time we applied the algorithm on the JPO data, the gradient chain approach to adjust the namesake prediction was not yet conceived.

ratio in regard of average inventor careers spawned per name. As expected, the higher namesake density in the Japanese population leads to more inventors hiding in a namespace. The average Japanese inventor has also more patents than her European or US counterpart, reflecting the fact that the Japanese patent system has a different definition of inventive claims. The USPTO and the EPO have roughly the same share of 2% of inventor careers yielding patents in different countries (see “Crossing Borders”). We were not able to retrieve this information from the Japanese data because the source database Patstat is notorious for insufficient data quality in terms of addresses.

We have shown that it is possible to disambiguate large bibliometric databases without the requirement of training datasets, which, given the precarious representation of namesakes in samples based on individuals, are of questionable value. The algorithm, at its core, relies also on a training dataset of mostly German individuals to aggregate a name population for the namesake estimator. We are aware of the fact that this estimator may under-perform for especially homogenous name populations. Applying a more restrictive threshold strategy and gradient chain adjustment of the namesake prediction can alleviate this shortcoming at the cost of uncomfortably arbitrary decisions. Nevertheless, the intuitive nature of the namesake risk assessment is well suited to monitor the impact of different settings on handpicked namespaces allowing a relatively quick and hassle free disambiguation of any bibliometric data source.

A. Appendix: Nested Cascaded Traversal of Intransitive Similarity Networks

First, we compress the data of all properties into respective versions without duplicates. For hard properties, the sequence number of the compressed data is already the cluster ID. For soft properties, the compressed table is the source and the target for a self-referential search algorithm to identify the similarities between the entries. For every entry, the algorithm selects potential candidate entries using Meta information about the frequencies of words within the data, retrieved from the data source itself. Every word of the search entry is weighted by the inverse of its respective frequency retrieved from the Meta data. The algorithm perceives anything separated by blanks as a word. Internal preparation routines guarantee a general harmonization level (upper case, replacement of special characters and so on) and optionally implement linguistic methods like Soundex, Metaphone or n-grams to improve the robustness against misspellings. Common words, like legal forms or frequent phrases, get low weights compared to more identifying words. The algorithm further separates the Meta data according to the originating source field to avoid the blending of frequencies of different contexts. A common street name does not swamp the frequencies of the applicant name field where the same word appears less often. Superordinate weights on these contexts allow for extensive control over the search and the measurement, i.e. putting 70% on the applicant name and 30% on the address with a threshold of 90% enforces the requirement of partial similarity of the address even if the name matches perfectly. The share of the weights, the joint words accumulate, measures the quality of a candidate.

The result of the self-referential search is a list of matches consisting of all distinct property entries, the respective candidates and their similarity scores, which are greater equal a high threshold. This list is neither commutative nor transitive allowing the following cases: A matches B but B does not match A; A matches B and B matches C but C and A do not match. If the list would have been transitive, we could already designate the cluster IDs by simply choosing the minimum or maximum entry ID per candidate. This method is not applicable to our result, which needs recursive traversal following the intransitive links through an implicitly constructed network to let the inherent clusters emerge. On first sight, this seems to be a disadvantage to methods producing or enforcing transitive results, but the additional freedom creates flexibility. For example, a firm group name is matched with several subsidiaries

containing the mother's name as part of a much longer specification. Because of the additional clutter in the names, the subsidiaries are in most cases not matched with the mother. Some subsidiaries may be joint ventures connecting to a different group of firms spreading out the network. Even links between different historical versions of the same firm name, including mergers, can be detected, as long as there are still some overlaps. This is especially important as bibliographic and patent data notoriously contain historical information. On the other hand, this behavior also creates completely unrelated connections, usually if matches with a low identification potential are involved. Traversing these networks without further precautions will lead to unexpectedly large clusters containing mostly unrelated entities. We call the method to handle this issue *nested cascaded traversal*. In short, it introduces a sequence of arbitrary size limiters combined with incremental conditions on the match quality, both based on experience with the data and educated guesses. During traversal, every time the cluster size exceeds a cascade limit, the associated rule set activates, enforcing higher requirements on the quality of a match and reinitiating the traversal at the respective start node.

The term *nested* refers to the augmentation of this simple method by inducting a second layer of cascades. The first cascades implement strict rules based on high thresholds. They should harmonize the data by already bundling clusters of very similar entries. This addresses the disparity between long entries with many words and short, concise entries with much less room for variations. After harmonization by the first layer, the assessment of cascade limits of the second layer does not have to cater for those intricacies, but can concentrate on the general tolerance of major, intransitive transitions.

The separation of the search process and the clustering facilitates a high level of flexibility. Different cascade definitions can be applied without repeating the time consuming search. The universal approach of our disambiguation routine allows for multiple differently granulated cluster formations of the same context. For technical details, see Doherr (2017). The program is available on GitHub: https://github.com/ThorstenDoherr/searchengine

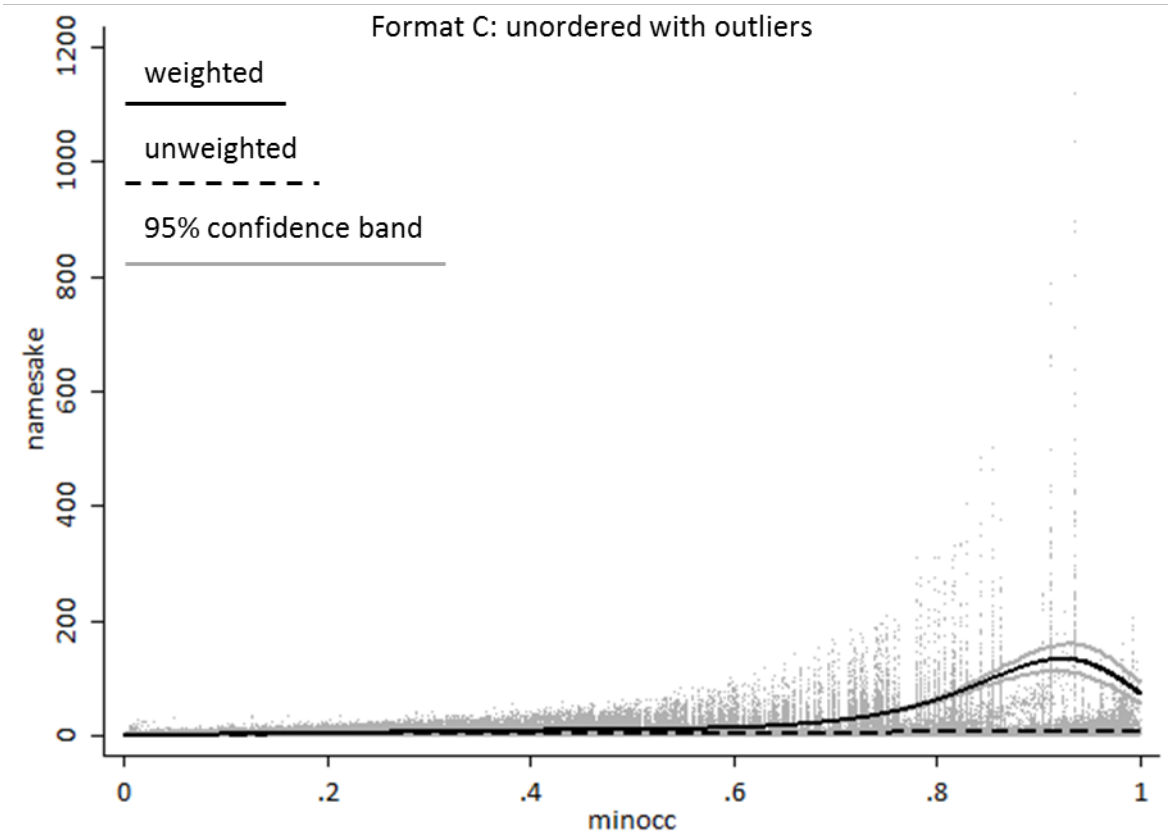
B. Appendix: Tables and Figures

Table 4: Weighted Poisson regression of namesakes on *minocc* (5th degree polynomial).

namesakes	Format A first name, last name	Format B initials	Format C unordered
minocc	9.393937*** (1.0015)	45.904456*** (3.6871)	9.050922*** (1.1975)
$minocc^2$	−25.628787*** (8.6151)	−210.223650*** (24.4674)	−24.064323** (10.3669)
$minocc^3$	58.247685** (25.4380)	483.059392*** (64.1039)	51.376449* (31.0283)
$minocc^4$	−64.434441** (30.4690)	−511.296700*** (71.6261)	−52.977835 (37.6794)
$minocc^5$	28.335309** (12.7382)	200.781658*** (28.5667)	22.484273 (15.9620)
const	−0.0215839 (0.01635)	−0.791724*** (0.1542)	−0.025849 (0.0217)
Pseudo R^2	0.7733	0.8411	0.7179
Observations	4,691,779	2,544,481	4,546,192

Notes: Equivalent to the non-aggregated model (one observation designates a person instead of a name) with standard errors clustered on names.

Figure 8: Predicted namesakes (weighted vs. unweighted) on scatter plot: Format C, including outliers



References

Doherr, Thorsten, 2017. Inventor Mobility Index: A Method to Disambiguate Inventor Careers. ZEW Discussion Paper No. 17-018, Mannheim

Doherr, Thorsten, 2021. The SearchEngine. https://github.com/ThorstenDoherr/searchengine

Falagas, M.E., 2006. Unique author identification number in scientific databases: a suggestion. PLoS Medicine 3(5)

Fenner, Martin, 2010. ORCID or how to build a unique identifier for scientists in 10 easy steps. Gobbledygook Blog, http://blogs.plos.org/mfenner

Jones, Benjamin, 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

Kim, Kunho, Madjan Kabsa, Lee C. Giles, 2016. Inventor Name Disambiguation for a Patent Database using a Random Forest and DBSCAN. IEEE/ACM Joint Conference on Digital Libraries (JCDL)

Krizhevsky, Alex, Ilya Sutskever, Geoffrey E. Hinton, 2017. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, Volume 60, Issue 6, 84-90

Lissoni, Francesco, Andrea Maurino, Michele Pezzoni, Gianluca Tarasconi, 2010. APE-INV's "Name Game" Algorithm Challenge: A guideline for benchmark data analysis & reporting. European Science Foundation, http://www.esf-ape-inv.eu/download/Benchmark_document.pdf

Magerman, Tom, 2015. PatentsView Disambiguation Inventor Workshop. Presentation at the USPTO, https://livestream.com/uspto/PatentsViewInventorWorkshop/videos/100138868

Petrie, Steve, T'Mir Julius, Russel Thomson, 2017. Author name disambiguation using a neural network-based algorithm. Presentation at the Web of Science data workshop at EPFL, Lausanne

Pezzoni, Michele, Francesco Lissoni, Gianluca Tarasconi, 2012. How to kill Inventors. Testing the Massacrator Algorithm for Inventor Disambiguation. Cahiers du GREThA, No. 2012-29

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

Schoen, Anja, Dominik Heinisch, Guido Buenstorf, 2014. Playing the "Name Game" to identify academic patents in Germany. Scientometrics, Volume 101, Issue 1, 527–545

Singh, Jasjit, 2003. Inventor Mobility and Social Networks as Drivers of Knowledge Diffusion. Havard University Working Paper Series

Stirling, James, 1730. Methodus differentialis, sive tractatus de summatione et interpolatione serierum infinitarum. Royal Society, London

Torvik, Vetle I., Neil R. Smalheiser, 2009. Author Name Disambiguation in MEDLINE. National Institutes of Health Public Access, Author Manuscript

Trajtenberg, Manuel, 2004. The "Names Game": Using Inventors Patent Data in Economic Research. Presentation: http://www.tau.ac.il/~manuel/

Trajtenberg, Manuel, Shiff Gil, Melamed Ran, 2009. The "Names Game": Harnessing Inventors' Patent Data for Economic Research. NBER Working Paper 12479

Download ZEW Discussion Papers from our ftp server:

http://ftp.zew.de/pub/zew-docs/dp/

or see:

https://www.ssrn.com/link/ZEW-Ctr-Euro-Econ-Research.html
https://ideas.repec.org/s/zbw/zewdip.html

IMPRINT

ZEW – Leibniz-Zentrum für Europäische Wirtschaftsforschung GmbH Mannheim
ZEW – Leibniz Centre for European Economic Research

L 7,1 · 68161 Mannheim · Germany
Phone +49 621 1235-01
info@zew.de · zew.de