Analytics on Non-Normalized Data Sources: more Learning, rather than more Cleaning

Alexis Cvetkov-Iliev, Alexandre Allauzen, Gaël Varoquaux

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Analytics on Non-Normalized Data Sources: more Learning, rather than more Cleaning

ALEXIS CVETKOV-ILIEV[1], ALEXANDRE ALLAUZEN[2], and GAËL VAROQUAUX.[1]
[1]Soda, Inria Saclay – Palaiseau, France (e-mail: alexis.cvetkov-iliev@inria.fr, gael.varoquaux@inria.fr)
[2]ESPCI – Paris, France (e-mail: alexandre.allauzen@espci.psl.eu)

Corresponding author: Alexis Cvetkov-Iliev (e-mail: alexis.cvetkov-iliev@inria.fr).

This work was supported in part by the French *Agence Nationale de la Recherche* under Grant ANR-17-CE23-0018 (DirtyData) and Grant ANR-20-CHIA-0026 (LearnI).

ABSTRACT Data analysis is increasingly performed over data assembled from uncontrolled sources, facing inconsistency in knowledge-representation conventions. The typical practice is to create "clean" data for analysis, matching entities and merging variants to overcome differences in knowledge representation. Despite progress in data management techniques to automate this process, it still needs labor-intensive supervision from the analyst. In this paper, we evaluate the benefit of advanced statistical tools to address directly many analytic tasks across data sources without such entity-matching cleaning. Reframing analytical questions as machine-learning tasks enables to replace exact matching of entities by continuous descriptions –vectorial embeddings– that expose similarities between entries.

But are analyses with less cleaning trustworthy? We answer this question with a thorough benchmark on questions typical of socio-economic studies across 14 employee databases: we compare the approaches based on machine learning to manual data cleaning (entity matching). It reveals that using embeddings and machine learning improves results validity (smaller estimation error) more than manual cleaning, with considerably less human labor. While machine learning is often combined with data management for the purpose of cleaning, our study suggests that using it directly for *analysis* is beneficial because it captures ambiguities hard to represent during curation.

INDEX TERMS data analysis, data cleaning, data integration, embeddings, entity-matching

I. INTRODUCTION

Data analysis is increasingly performed across non-normalized data sources, facing data-integration challenges. For instance estimating product prices must match offers referring to the same product [1]; studying the influence of climate warming on plant species must overcome variability in plant names [2], [3]; and early detection of acute kidney injuries faces the heterogeneous vocabulary of clinical notes [4]. Assembling and curating data into a clean form for statistical analysis is often described as one of the biggest hurdles to data science [2], [5], [6]. Even when the schemas are aligned, data curation needs some form of entity matching. Indeed, across different information systems, there are often different ways to represent the same concept, *e.g.* "professor", "prof", or "professeur". Entity matching bridges the various knowledge-representation conventions across sources to produce normalized entries (see Fig. 1). Though this curation process can be partly automated [7], [8], it remains challenging and time-consuming as it requires tedious manual supervision and quality assurance across a large number of entries. Yet, with current analytic methodology this task is central to the validity of downstream analyses.

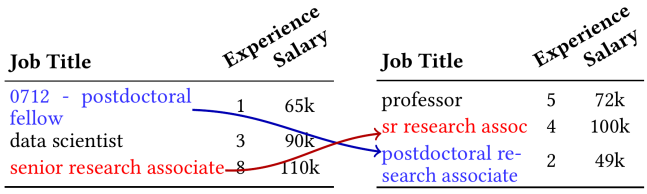

FIGURE 1. Entity matching across two employee databases.

Here we show that more sophisticated analysis pipelines

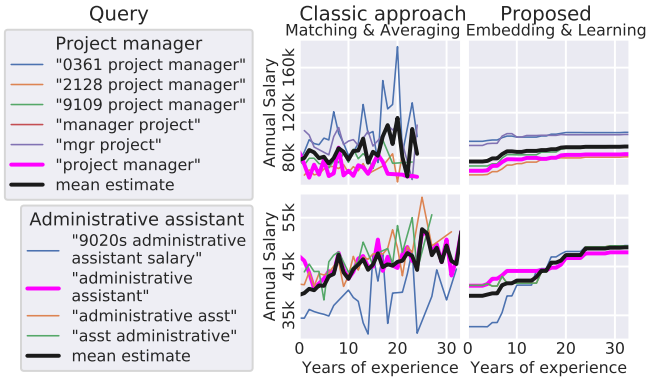

FIGURE 2. An analysis of salary as a function of experience for different job types, considering variants of *administrative assistant* and *project manager*, computed by matching & averaging, or embedding & learning. Thick magenta lines are the most natural query.

can alleviate the need for careful curation to answer many data-science questions. We empirically benchmark such approaches, drawing examples from studies of the determinants of salary, in data journalism [9] or academia [10]–[12], which ask quantitative questions such as:

1. *For a given job, how does salary evolve with experience?*
2. *For a given job, what is the 0.75-quantile of salaries?*
3. *What is the typical male-female pay gap?*

Answering these questions from data assembled across different employers must overcome the lack of correspondences in job titles: as illustrated on Fig. 1, the same job title appears with multiple variants, such as *senior research associate* and *sr research assoc*. This problem is typically addressed via *entity matching* procedures, available in increasingly sophisticated data-integration softwares such as *Wrangler* [13], *Tamr* [14], *OpenRefine* [15]. Despite these tools, entity matching remains a difficult task as it often involves domain expertise or faces the lack of clear correspondences in entities across sources. Is such matching necessary to the validity of the analysis, or can more complex statistical pipelines do without?

In this paper, we show that applying advanced statistical techniques directly to non-normalized data can avoid labour-intensive data curation for many analytical questions. We benchmark whether relying more on machine learning and less on manual data cleaning compromises or not the validity of the analysis. To answer this important question, we formalize how many analyses boil down to estimating statistical quantities, and use various experiments that give an unbiased measure of the corresponding estimation error.

The quantities needed for the analytical questions can be estimated with machine-learning models applied to continuous embeddings of entries that represent ambiguities. A suitably trained model can be directly queried to give *e.g.* the evolution of salary with experience for a given job, giving a less noisy result than standard techniques after best-effort manual cleaning (see Fig. 2). Machine learning is already increasingly used in data integration to create more uniform data warehouses [16], [17] or to clean their entries [18], [19]. Instead, our study applies it directly to the analytical question, as this can be easier than curating the data for fundamental reasons. First the analytic task provides supervision [20]–[22], while cleaning needs examples of curated data. Second, representing ambiguities in the analysis often leads to more accurate results.

We first give a brief summary of how data cleaning is used to enable data analysis with diverse sources; after which we show how data-science questions can be formulated in terms of machine learning on embeddings of entries. We then study the validity of the results: on an analysis of wages across 14 data sources, we compare manual data cleaning to a simple machine-learning approach using embeddings of entries. Qualitatively and quantitatively, analyzing the non-normalized data gives better results. Finally, we discuss perspectives on adapting data-analysis practices to rely less on cleaning.

II. THE CLASSIC VIEW: CLEANING FOR ANALYTICS ACROSS SOURCES

A. ENTITY MATCHING TO INTEGRATE DATA

Integrating data often faces alignment challenges, in particular if it is assembled across sources. Notably, correspondences are first needed at the schema level: different sources may come with different structures, *e.g.* columns (*relations*) of different names for the same information, or information split differently across columns [23]. Bridging such mismatches is known as *schema matching*, and is often required as a data-preparation step. However, it tends to burden less the human operator than instance-level entity matching, because there are less matches to check. We thus focus on entity matching in the remainder, though there are many other aspects to data quality [19].

Entity matching

For data integration, *entity matching* strives to match different variants that denote the same entity [24]. Classic situations include *deduplication* of multiple variants of the same entity in a given table, or *record linkage*, matching entities across two tables [25]. Matching entries to uncover categories is necessary for standard statistical procedures to answer a question *conditional* to a non-normalized category: analyzing one quantity –such as salary– keeping another –job title– constant. [26]. Conversely, computing marginal quantities, such as the overall distribution of salary, does not require entity matching as it relies on aggregates of all the data (assuming that there are no duplicate across the sources).

Entity-matching techniques rely on an appropriate similarity —typically across strings— and threshold to assign entries to the same entity. The issue is that both similarity and threshold must be tailored to domain specificities and the resulting matches must often be manually reviewed. The process is thus labor intensive.

TABLE 1. A few example rows of the employee tables.

JOB TITLE	HIRING DATE	SEX	ETHNICITY	SALARY ($)
Police Officer	17/03/2005	M	White	85 000
Security Manager	24/06/2017	F	Asian	70 000
Energy Analyst	04/11/1998	F	Black	105 000
Librarian	11/09/2011	M	Hispanic	50 000

Automating the match of entities is challenging because it is an unsupervised-learning problem [27], unless there are known matches for supervision [28]–[30]. Such matches must typically be constructed manually by a user, though active learning can reduce human intervention [18], [31]. Dedicated data-integration softwares, such as OpenRefine, facilitate the process with a user interface. The software suggests potential matches and enables users to tune parameters and match entries in a semi-automated way.

Automation tools can use techniques capturing natural language semantics, which shares with entity matching the challenge of relating multiple forms that denote the same things. For instance, natural language processing tools such as `fastText` [32] provide word embeddings resilient to morphological variations. Embeddings have led to many recent progresses in entity-matching pipelines [8], [30], [33], [34].

Cleaning real-world salary data

To answer our questions on salary, we consider data from a study of salaries in Texas state institutions [1], as illustrated in Table 1. The tables are assembled across 14 different employers and the Job Title information is particularly challenging: without normalization there are about 14 000 different job positions for a total of around 160 000 employees.

We performed manual entity matching on the job titles using OpenRefine. We first cleaned common abbreviations as they are the main hurdle to entity matching: string metrics struggle to capture their similarities. Typical examples include (*sr*/*senior*), (*asst*/*assistant*) or (*mgr*/*manager*). Some abbreviations can have multiple or complex meanings (*tech*/*technology*, *technician*, *technical*), (*CMC*/*Chemical, Manufacturing and Control*). Such manual cleaning is thus limited by the domain expertise of the operator. We then used OpenRefine to search and manually merge variants across sources. Around 1000 job titles were paired in the process, which took about 3 days. We believe that a more thorough entity matching —especially on rare job titles— can be performed, but would require intensive human labour to bring minor improvements.

B. STANDARD ANALYTICAL PRACTICE: MATCHING & AVERAGING

In general, the analytic questions can be formalized as estimating a quantity y for a population or group of instances who share a set of attributes X: for instance, the typical salary of a *project manager* with 3 years of experience. To that end, the standard technique consists in *matching & averaging*:

1) a query on X to match and select the relevant instances.
2) a procedure (typically a form of averaging) to aggregate the results and estimate y.

Even when the entities in the data are normalized, a successful analysis may require to match them with the vocabulary used by the analyst: for instance in some data the correct query for *project manager* may be *mgr project* (Fig. 2).

In our example analysis, studying determinants of salary, the motivation to unite data sources is to establish a more general result: project-manager salaries or male-female pay gap may vary across employers; there may be no instance of a project manager with 3 years of experience in a given employer. The underlying problem is that of statistical estimation: to compute the quantity that represents best the complete population from the instances at hand. If the entity matching is valid, matching & averaging estimates are unbiased from a statistical point of view. But they may exhibit high variance, as the data often exhibits a small number of representatives from a given category. A paradox of statistics is that the most accurate way of estimating the mean of a population from a small sample may not be the sample average, but biasing estimates with other sources of information (see Stein's paradox [35]). For instance, estimating the typical salary of an *associate professor* can leverage similar populations: *professor*, *lecturer*. Drawing information across similar entities is related to the notion of semantic queries in databases [36], as opposed to exact value matching, used in matching & averaging.

III. ANSWERING ANALYTICAL QUESTIONS WITH MACHINE LEARNING

We now give the statistical underpinnings of approaches based on machine learning. We specifically consider the three analytical questions of our example study on determinants of salary. While these originally do not appear to be machine learning problems, we show that they can be reformulated as such. An overview of our analysis pipeline is depicted in Figure 3.

A. SALARY EVOLUTION AND QUANTILES

To study the evolution of one quantity, salary, as function of another, experience, matching & averaging methods typically group employees by job and experience level, and compute the mean salary in each group. This quantity is an estimate of the conditional expectation $\mathbb{E}[\text{Salary} \mid \text{Job, Experience}]$.

Instead of averaging on groups, a machine-learning model trained to predict the salary given the job and experience level can estimate this quantity. Indeed, modeling the salary as a function $f_\theta(\text{Job, Experience})$ gives a *consistent*[2] estimate of the conditional expectation $\mathbb{E}[\text{Salary} \mid \text{Job, Experience}]$ if

[1] Data available on https://dx.doi.org/10.21227/wfjs-ya22

[2] A *consistent* statistical procedure converges to the population values with increasing data size.

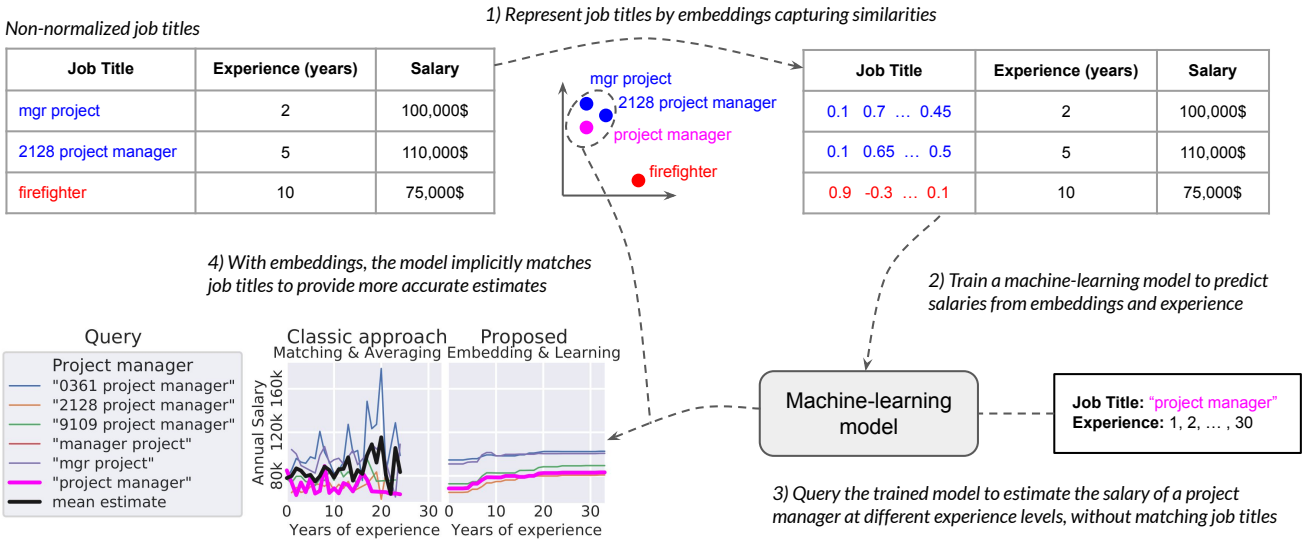

FIGURE 3. Overview of our "embedding & learning" pipeline to answer analytical questions on non-normalized data.

model parameters θ are optimized on the data to minimize the mean squared error on the salary [37, section 1.5.5].

Unlike averaging, casting the analytic question into a machine-learning task does not require matching. Rather than viewing each job title j as a discrete category, we can represent it by a vector $\mathbf{j} \in \mathbb{R}^p$, that will serve as features to train the machine-learning model f_θ.

The crucial point here is that these vectors should capture the similarities between job titles. For instance, *administrative assistant* and *administrative asst* should have close representations. This allows the machine-learning model to leverage these similarities and implicitly account for matching. We use in our experiments pretrained fastText embeddings [32], which readily provide vector representations for strings and account for semantic and morphological similarities. Other approaches to encode string similarities into vectors could be used as well [26], [38].

A wide variety of machine-learning models can be used to estimate the quantity of interest. For our experiments we use gradient boosted tree models from *scikit-learn* [39], as they generally perform well in prediction tasks. Besides avoiding tedious entity-matching, machine-learning models can form weakly-parametric estimators that are resilient to other imperfections in the data. For instance, imperfect correspondences between schemas across the sources lead to missing values: some sources may not have all the information. Despite these missing values, supervised learning can give optimal estimates without relying on probabilistic modeling of the missing-data mechanism [40], [41].

A model can be trained to estimate different statistical quantities by choosing the measure of error (loss) that it minimizes [42]: a mean squared error leads to conditional expectations. Similarly, a model $f_\theta(\text{Job})$ trained with a quantile loss estimates a quantile of the salary distribution for a given job [43]. The appendix (sec. VII-A and VII-B) gives more details about the specific implementation of the analysis.

B. PAY GAP ACROSS SEX: COUNTERFACTUAL ANALYSIS

Many of the advanced analyses and visualizations of rich data sources pertain to understanding causes and effects. For instance, when studying salaries, measuring and understanding the causes of gender gap is a long-running question [12], [44], [45]. Shedding light on this question requires contrasting salaries for man and woman at similar position, with similar experience... Can it be done without cleaning the data?

Counterfactual analysis provides a good framework to address quantitatively the question of gender pay gap. A counterfactual is a thought experiment measuring the effect on the outcome of interest —the salary y_i— of changing only the feature of interest W_i —here the sex— for an individual i. Borrowing from clinical trials, W_i is called the "treatment" in the literature. The outcome y_i can take two potential values depending on W_i: $y_i(0) = y_i(W_i = 0)$ or $y_i(1) = y_i(W_i = 1)$ (1 for a man, 0 for a woman), though for each individual only one of these is observed in the data.

The analytical quantity of interest is the typical gender pay gap, known as the *average treatment effect* $\tau = \mathbb{E}[y(1) - y(0)]$: the average difference in outcome for the same individual under scenario $W_i = 1$ and $W_i = 0$, *i.e.* that differ only by their sex [46], [47]. In general, it does not suffice to subtract the average salary of men from that of women: $\mathbb{E}[y|W = 1] - \mathbb{E}[y|W = 0] \neq \tau$. Indeed, the populations of men and women may not be directly comparable in the database. For instance, women may need to interrupt their careers during maternity leave, causing them to have less work experience and thus lower salaries. To account for such

TABLE 2. Illustration of the potential outcome framework. The data contains an observation of a male white firefighter with 15 years of experience, but not a matching female employee; likewise with an hispanic female post-doc with 2 years of experience. The challenge is to interpolate the missing data.

Covariates X			Treatment	Outcome	
Job Title	Experience	Ethnicity	W (man?)	y(0)	y(1)
Firefighter	15	White	1	NA	75000
Post-doc	2	Hispanic	0	60000	NA

confounding factors and isolate the effect of interest, the *potential outcome* framework (Table 2) uses *covariates*, extra information X on each individual, such as the job title or the experience level, allowing us to directly compare salaries for men and women with the same features.

To estimate the average treatment effect, modern causal inference techniques either rely on estimates of the outcome given the covariates and the treatment, $\mathbb{E}[y|X, W]$, or estimates of the *propensity-score* $\mathbb{P}(W = 1|X)$, i.e. the probability for an employee to be a man given its covariates. The first quantity can be estimated using regression models, trained to predict y from X and W, as detailed before. Similarly, the propensity-score can be estimated with classification models, trained to predict W from X, when they are *calibrated* [48]. Finally, powerful causal-inference tools combine both estimates for more robustness [49]. State-of-the-art approaches already rely on machine-learning models to adapt to biases and noise in the input data [50], [51]. The appendix (sec. VII-D) details the exact models used in our experiments to estimate the average treatment effect.

A pattern: machine learning

Because machine learning can capture complex links in complex data, it is increasingly used in data science to estimate quantities of interest to the analyst, whether they are intermediate quantities, as for counterfactual analysis (subsection III-B), or the direct answer to the question of interest, as for conditional links (subsection III-A). For data integration, this evolution brings exciting new opportunities: machine-learning models do not need to rely on averaging, and hence do not need actual matching of entities across sources. Rather, they can use vector representations that express, even indirectly, relevant similarities between entities.

IV. EMPIRICAL STUDY: LEARNING VERSUS CLEANING

Using machine learning can be less labor-intensive, as it does not require human-guided entity matching. But does it come at a cost to the validity of the results? We now compare empirically learning and matching-based approaches for the different analytic questions.[3]

[3]The code and data to reproduce our experiments is available on Code Ocean: https://codeocean.com/capsule/6435573/tree

A. EXPERIMENTAL DETAILS

Measuring estimation error

How to compare estimators of a quantity such as conditional expectation of salary given job title? Even without entity-matching noise, the data at hand is limited and its mean is an imperfect estimate of the unknown population quantity y. We adapt a classic procedure of machine learning: we leave out a *test* fraction of the databases, and use the rest of the data to derive estimates $\hat{y}_{\text{train}}$. Applying an averaging-based estimator on the *test* data provides another estimate $\hat{y}_{\text{test}}$, that is unbiased though noisy. Importantly, as it has been estimated from different data than $\hat{y}_{\text{train}}$, its estimation error is independent. We can thus use the difference between $\hat{y}_{\text{train}}$ and $\hat{y}_{\text{test}}$ over multiple splits —a cross-validation loop— to quantify the estimation error of the procedure that we use to compute $\hat{y}_{\text{train}}$.

Analytical approaches studied

We compare several approaches to estimate the quantities relevant to our analytical questions (implementation details are provided in the appendix):

1) **Matching & averaging**, as described in subsection II-B.
2) **Embedding & learning**: strategies of section III, relying only on standard machine-learning tools. Gradient boosted tree models from *scikit-learn* [39] are trained, using pretrained fastText embeddings [32] to represent the job titles, capturing semantic and morphological similarities.
3) **Embedding & fuzzy matching**: the notion of continuous similarities, as between embeddings, can also be exploited to define weighted averages. We modify the matching & averaging procedure to use fuzzy matches and weights defined with a cosine string similarity on the job title with an affine decay and a cut-off at zero.

Parameters such as the affine decay and cut-off, or the hyper-parameters of the machine-learning models are tuned in a nested cross-validation procedure. To study the effect of entity matching, we apply these techniques on raw and manually matched entries.

B. QUALITATIVE RESULTS: DISPERSION ACROSS VARIANTS

The curves of salary as a function of experience represented on Fig. 2 are computed either with a matching-based or a learning-based approach. Machine-learning estimates leverage job similarities and have low dispersion across variants of *project manager* or *administrative assistant*. This robustness reduces the need for manual matching: taking the model output on any variant provides reliable estimates that are representative of the whole population. It is more convenient and reliable for an analyst to query the model for "project manager" or "administrative assistant" (thick magenta curves), than to search the database for all variants and average them. Beyond the dispersion across variants, matching &

TABLE 3. Cross-validated errors for salary, quantile, and propensity-score estimation. We report here estimates of the propensity-score $P(W = 1|J)$ conditionally to the job title, rather than on all covariates, as matching-based estimates are very noisy in that case. RMSE = Root Mean Square error. MAE = Mean Absolute Error.

Estimation method	Manual matching	Salary (RMSE)	Quantile (MAE)	Propensity (Brier score)
Matching & averaging	Yes	55634	31802	0.231
Embedding & Fuzzy matching	No	52812	30955	0.195
Embedding & Fuzzy matching	Yes	51506	28851	0.192
Embedding & Learning	No	52683	28726	0.189
Embedding & Learning	Yes	50614	26713	0.184

averaging curves appear more noisy; in particular they fail to capture well the evolution of salary with experience. Finally, machine-learning estimates show plausible extrapolations for queries where there is no data with exact matches, such as project managers with more than 25 years of experience.

C. QUANTITATIVE RESULTS: CROSS-VALIDATED ERRORS

To go beyond the face validity of Fig. 2, we use cross-validation, as detailed in subsection IV-A, to quantify which approach best estimates the population quantities. The 14 databases are randomly split into two sets of 7 databases: one to compute estimates for salary, quantile, and propensity-score; and the other to measure their error, reported in Table 3. Results show that for all three quantities embeddings notably reduce the error compared to exact matching and perform best when combined with learning. Adding manual matching on top of embeddings improves further, but the benefit is smaller than that brought by embeddings & learning. The residual error is due to variance in individual salary that is not explained by the attributes of the employees present in the databases, such as the appreciation of the manager.

D. ESTIMATION OF COUNTERFACTUALS

How do the differences in estimation errors reported in Table 3 impact complex end-user analytical questions? We investigate their impact on estimation of salary gap across sex. Fig. 4 gives average treatment effects computed with statistical methods –IPS and AIPS [52]– based on embedding & learning approaches, as well as manual matching and fuzzy-matching estimates (see appendix). To force the need for analysis across the databases, we create a sex imbalance by dropping randomly a fraction of either men or women in each database, with 50/50 probability. As a result, the estimation relies on employees of opposite sex with matching job titles across databases. Machine learning methods have much less variance than matching and averaging methods, but both approaches lead to estimates across databases (large sex imbalance) that do not depart for values obtained within databases (no sex imbalance). On the other hand, fuzzy matching creates sizeable bias: an analysis performed across databases differs markedly from an analysis comparing employees inside each database. The low variance of machine-learning methods comes from their implicit interpolation, visible on Fig. 2: if a given employee lacks an opposite-sex with the same covariates, the model will use information from similar profiles.

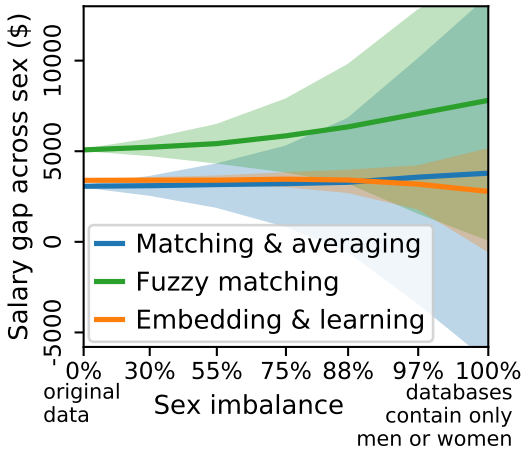

FIGURE 4. Salary gap: Average treatment effect computed with added sex imbalance in individual databases, forcing the need for analysis across databases. The error bars give the quartiles across random deletion of men or women records.

V. DISCUSSION: HOW MUCH CAN LEARNING REPLACE CLEANING?

On the data-integration problem that we have studied, relying more on learning rather than on cleaning facilitates the data analysis, and actually improves the validity of the results without manual labor. This result depart from classic data-management practices, and we now discuss its interpretation and impact for analytical practices.

A. CLEANING IS IN THE EYE OF THE BEHOLDER

Cleaning is analysis

Studying the salary gap showcases the importance of analysis across data sources: for the highest-payed positions, finding employees of opposite sex requires considering multiple companies. Matching entities faces the fundamental challenge that there might not be exact correspondences: not every institution has a *chief data officer* (CDO) and the nearest match may be *chief technology officer* (CTO). Omitting companies without CDO will bias the analysis by excluding large tech companies.

The notion of cleaning, to make data more uniform, carries in itself analytical choices which may bias the results [53], [54]. While *vinaigrette* is just French for *salad dressing*, its use on an American's restaurant menu signals upper-scale clientele. Merging the two will lead to loss of information. From an ontological point of view, the solution would be to create a new category, *posh salad dressing*. But maintaining a complete and consistent ontology, catering for all the edge cases, requires manual work each time new data is integrated. Should the necessity to merge entities be considered as a bug of analytic pipelines, rather than a feature? New tools that do not require exact matches can give more reliable analyses

in the face of ambiguity, as illustrated by the estimation of salary gap across databases with sex imbalance (Fig. 4).

Manually curating entity matching brings to the data a consistency that is good practice in production settings. Yet, as illustrated in our empirical study, favoring more advanced statistics down the line facilitates valid analysis. It can indeed be easier to pass on uncertainties to the statistical analysis tools than to resolve them in a relational store. The best data representation, clean or fuzzy, is tied to the analytic question.

B. SUPERVISION FACILITATES INTEGRATING DATA WITH AMBIGUITIES

Representing uncertainty in relational systems helps tackling ambiguities [36], [55], [56] or curating data [57]. However, extending relational data management to a general probabilistic framework is intrinsically hard. Indeed, unlike with the relational algebra, queries in a probabilistic database can suffer non-polynomial complexity [58]. Approximate probabilities [59], [60] or fuzzy logic and similarities [61] have better tractability. Yet how to weight similarities to best capture ambiguities is often a challenge in itself.

Using supervised learning to answer a given statistical question alleviates the need for probabilistic models. In particular, many recent success rely on *discriminative* modeling using empirical risk minimization, as with deep learning [62]. It is crucial to the success of our empirical study: optimizing the statistical models gives accurate estimates from non-probabilistic similarities –word representations that were not tailored to the question at hand. Such an approach goes much further than fuzzy matching (Fig. 4), as supervised learning can be seen as implicitly tuning scaling factors and thresholds to combine information optimally while minimizing noise.

Embeddings to capture ambiguities

Entity embeddings are crucial to the success of our approach, to expose ambiguities to the analysis step. Our proof of principle purposely used a very simple implementation: a general-purpose machine-learning model applied on off-the-shelf word embeddings. Yet, it is noteworthy that it leads to analyses on the unaligned data more accurate than standard statistical approaches on data cleaned with three days of manual labor using a dedicated software (Table 3, Fig. 4). There is ample room to use better embeddings of entries, for instance training them from the data at hand to adapt to its specificities, via the string forms [38] or the relations to other entities [63] including distant relational information [64].

C. THE ROAD AHEAD: RETHINKING ANALYTIC PIPELINES

More complex data-integration pipelines

The data-integration problem studied in section III is very simple: it consists in analyzing the union of tables across sources. In relational algebra terms, the machine-learning models replace a `GroupBy` followed by aggregations. However, data integration often calls for joining and aggregating across tables of different nature. Tackling these operations using machine learning on embeddings will require exploring new tools, for instance adapting similarity joins to merge information across tables [65], [66], logic inferences on top of entity embeddings [67], or graph CNNs for relational data [68].

Back to the data scientist: opening up black boxes

Without explicitly merging variants into a small number of human-recognizable entities, data-analysis pipelines can be complicated to audit for the human analyst. And yet, such human inspection of pipelines is often important for validation and debugging. Understanding analytic pipelines based on machine learning rather than cleaning will need techniques from the growing field of black-box model explanation in AI [69]: counterfactual reasoning can be applied to understand how data-assembly pipeline transforms an input [70]; permutation importance can gauge how a given attribute impacts the results by shuffling its values across instances [71]; finally, entity embeddings can be crafted to relate to human-comprehensible notions, for instance revealing latent categories [38].

D. CLEANING OR LEARNING? TWO COMPLEMENTARY TOOLS

Replacing explicit cleaning by machine learning follows the trend from "schema on write" to "schema on read": it displaces the burden from the data producer to the data consumer [72].

Cleaning is difficult, but it comes with the hope that the efforts will yield long-lasting benefits, useful for multiple usages of the data. These hopes are certainly well-grounded. Yet cleaning never ends; ambiguities in entity matching must be revisited given a new topic of analysis, or a new data source to integrate [5]. On the other hand, while variations may capture nuances –*vinaigrette* being posh for *salad dressing*–, expressing the exact same entity in two different ways is often an unnecessary hurdle to data integration. Standard vocabularies, as the universal resource identifier (URI) developed for *linked data* [73], address these hurdles. They are complementary to a strategy based on embedding and learning, and can be priceless to bridge data sources, even if only a fraction of the entities can be expressed within the vocabulary. An analysis using machine learning to tackle ambiguities will be more successful if there are only few of these ambiguities. If data is normalized *enough*, data integration can leverage off-the-shelf embeddings, as `FastText` used in our proof of concept. These continuous embeddings are complementary to standard vocabularies.

VI. CONCLUSION: LEARNING CUTS HUMAN LABOR BUT KEEPS VALID RESULTS

Ambiguities often arise when analyzing data, for instance if it comes from different sources with different conventions. The analysis then faces a fundamental challenge of validity: has the data been merged right, so as not to bias the results? The correct correspondence between entities across different data

representations depends on the goal of the analysis: when integrating a "CDO" –chief data officer– into a employee directory that does not know such role, it could be legitimate to convert "CDO" to "executive officer" to study salary, or "data scientist" to study expertise.

The traditional view is that data cleaning is necessary to a valid analysis: carefully establish correspondences, typically combining automated approaches with manual supervision and quality assurance. Rather, our benchmark shows that valid answers to a given analytic question can be assembled by exposing ambiguities to a machine-learning pipeline. Indeed, many questions that do not explicitly call for machine learning can be formulated using such models as flexible estimators of the underlying quantities. Our empirical comparison of a simple machine-learning approach to a labor-intensive manual cleaning shows that learning improved the quality of the analysis as much, if not more, than the cleaning. We hope that it can provide a point of reference to future analysts, and justify saving time on manual cleaning.

REFERENCES

[1] R. Agrawal and S. Ieong, "Aggregating web offers to determine product prices," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 435–443.
[2] A. Nazabal, C. K. Williams, G. Colavizza, C. R. Smith, and A. Williams, "Data engineering for data analytics: a classification of the issues, and case studies," *arXiv preprint arXiv:2004.12929*, 2020.
[3] A. D. Bjorkman, I. H. Myers-Smith, S. C. Elmendorf, S. Normand, N. RÃijger, P. S. A. Beck, A. Blach-Overgaard, D. Blok, J. H. C. Cornelissen, and B. C. Forbes, "Plant functional trait change across a warming tundra biome," *Nature*, vol. 562, no. 7725, pp. 57–62, 2018.
[4] Y. Li, L. Yao, C. Mao, A. Srivastava, X. Jiang, and Y. Luo, "Early prediction of acute kidney injury in critical care setting using clinical notes," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, pp. 683–686.
[5] M. Chessell, F. Scheepers, N. Nguyen, R. van Kessel, and R. van der Starre, "Governing and managing big data for analytics and decision makers," *IBM Redguides for Business Leaders*, 2014.
[6] Kaggle, "Kaggle industry survey," 2018. [Online]. Available: https://www.kaggle.com/ash316/novice-to-grandmaster
[7] V. Christophides, V. Efthymiou, and K. Stefanidis, "Entity resolution in the web of data," *Synthesis Lectures on the Semantic Web*, vol. 5, pp. 1–122, 2015.
[8] C. Zhao and Y. He, "Auto-em: End-to-end fuzzy entity-matching using pre-trained deep models and transfer learning," in *The World Wide Web Conference*, 2019, pp. 2413–2424.
[9] Texas Tribune, "Government salaries explorer," 2021. [Online]. Available: https://salaries.texastribune.org/
[10] D. H. Ciscel and T. M. Carroll, "The determinants of executive salaries: An econometric survey," *The Review of Economics and Statistics*, pp. 7–13, 1980.
[11] J. Xiao, "Determinants of salary growth in shenzhen, china: An analysis of formal education, on-the-job training, and adult education with a three-level model," *Economics of Education Review*, vol. 21, p. 557, 2002.
[12] F. D. Blau and L. M. Kahn, "The gender wage gap: Extent, trends, and explanations," *Journal of Economic Literature*, vol. 55, p. 789, 2017.
[13] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: Interactive visual specification of data transformation scripts," in *SIGCHI*, 2011, p. 3363.
[14] M. Stonebraker, D. Bruckner, I. F. Ilyas, G. Beskales, M. Cherniack, S. B. Zdonik, A. Pagan, and S. Xu, "Data curation at scale: The data tamer system." in *CIDR*, vol. 2013, 2013.
[15] R. Verborgh and M. De Wilde, *Using OpenRefine*. Packt Publishing, 2013.
[16] M. Stonebraker and I. F. Ilyas, "Data integration: The current status and the way forward." *IEEE Data Eng. Bull.*, vol. 41, no. 2, pp. 3–9, 2018.
[17] X. L. Dong and T. Rekatsinas, "Data integration and machine learning: A natural synergy," in *International conference on management of data*, 2018, p. 1645.
[18] S. Krishnan, J. Wang, E. Wu, M. J. Franklin, and K. Goldberg, "Activeclean: Interactive data cleaning for statistical modeling," *VLDB*, vol. 9, no. 12, pp. 948–959, 2016.
[19] I. F. Ilyas and X. Chu, *Data cleaning*. Morgan & Claypool, 2019.
[20] S. Krishnan, J. Wang, M. J. Franklin, K. Goldberg, T. Kraska, T. Milo, and E. Wu, "Sampleclean: Fast and reliable analytics on dirty data." *IEEE Data Eng. Bull.*, vol. 38, no. 3, pp. 59–75, 2015.
[21] S. Krishnan, M. J. Franklin, K. Goldberg, and E. Wu, "Boostclean: Automated error detection and repair for machine learning," *arXiv:1711.01299*, 2017.
[22] L. Berti-Equille, "Learn2clean: Optimizing the sequence of tasks for web data preparation," in *The World Wide Web Conference*, 2019, pp. 2580–2586.
[23] A. Doan and A. Y. Halevy, "Semantic integration research in the database community: A brief survey," *AI magazine*, vol. 26, no. 1, pp. 83–83, 2005.
[24] A. Elmagarmid, P. Ipeirotis, and V. Verykios, "Duplicate record detection: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 19, p. 1, 2007.
[25] W. E. Winkler, "The state of record linkage and current research problems," in *Statistical Research Division, US Census Bureau*. Citeseer, 1999.
[26] P. Cerda, G. Varoquaux, and B. Kégl, "Similarity encoding for learning with dirty categorical variables," *Machine Learning*, vol. 107, no. 8, p. 1477, 2018.
[27] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.
[28] M. Bilenko and R. J. Mooney, "Adaptive duplicate detection using learnable string similarity measures," in *KDD*, 2003, p. 48.
[29] A. McCallum, K. Bellare, and F. Pereira, "A conditional random field for discriminatively-trained finite-state string edit distance," *UAI*, p. 388, 2005.
[30] S. Mudgal, H. Li, T. Rekatsinas, A. Doan, Y. Park, G. Krishnan, R. Deep, E. Arcaute, and V. Raghavendra, "Deep learning for entity matching: A design space exploration," in *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 19–34.
[31] S. Sarawagi and A. Bhamidipaty, "Interactive deduplication using active learning," *KDD*, 2002.
[32] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, p. 135, 2017.
[33] M. Ebraheem, S. Thirumuruganathan, S. Joty, M. Ouzzani, and N. Tang, "Distributed representations of tuples for entity resolution," *VLDB*, vol. 11, p. 1454, 2018.
[34] J. Kasai, K. Qian, S. Gurajada, Y. Li, and L. Popa, "Low-resource deep entity resolution with transfer and active learning," in *Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5851–5861.
[35] B. Efron and C. Morris, "Stein's paradox in statistics," *Scientific American - SCI AMER*, vol. 236, pp. 119–127, May 1977.
[36] R. Bordawekar and O. Shmueli, "Using word embedding to enable semantic queries in relational databases," in *Proceedings of the 1st Workshop on Data Management for End-to-End Machine Learning*, ser. DEEMâĂŹ17, 2017.
[37] C. M. Bishop, *Pattern recognition and machine learning*. Springer New York, NY, 2006.
[38] P. Cerda and G. Varoquaux, "Encoding high-cardinality string categorical variables," *IEEE Trans. Knowl. Data Eng.*, 2019.
[39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
[40] J. Josse, N. Prost, E. Scornet, and G. Varoquaux, "On the consistency of supervised learning with missing values," 2020.
[41] M. Le Morvan, J. Josse, T. Moreau, E. Scornet, and G. Varoquaux, "Neumiss networks: differentiable programming for supervised learning with missing values." *Advances in Neural Information Processing Systems*, vol. 33, 2020.
[42] T. Gneiting, "Making and evaluating point forecasts," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 746–762, 2011.
[43] R. Koenker and G. Bassett Jr, "Regression quantiles," *Econometrica*, p. 33, 1978.

[44] A. S. Blinder, "Wage discrimination: Reduced form and structural estimates," *The Journal of Human Resources*, vol. 8, no. 4, pp. 436–455, 1973.

[45] R. Oaxaca, "Male-female wage differentials in urban labor markets," *International Economic Review*, vol. 14, no. 3, pp. 693–709, 1973.

[46] D. Rubin, "Estimating causal effects of treatments in randomized and nonrandomized studies," *Journal of Educational Psychology*, vol. 66, no. 5, p. 688, 1974.

[47] G. W. Imbens and D. B. Rubin, *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

[48] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 625–632.

[49] M. J. Funk, D. Westreich, C. Wiesen, T. Stürmer, M. A. Brookhart, and M. Davidian, "Doubly robust estimation of causal effects," *American journal of epidemiology*, vol. 173, p. 761, 2011.

[50] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, and J. Robins, "Double/debiased machine learning for treatment and structural parameters," *The Econometrics Journal*, vol. 21, pp. C1–C68, 2018.

[51] T. Blakely, J. Lynch, K. Simons, R. Bentley, and S. Rose, "Reflection on modern methods: when worlds collideâĂŤprediction, machine learning and causal inference," *International Journal of Epidemiology*, 2019.

[52] A. N. Glynn and K. M. Quinn, "An introduction to the augmented inverse propensity weighted estimator," *Political analysis*, vol. 18, p. 36, 2010.

[53] K. Rawson and T. MuÃśoz, "Against cleaning," in *Debates in the Digital Humanities*. University of Minnesota Press, 2019, pp. 279–292.

[54] M. Boumans and S. Leonelli, "From dirty data to tidy facts: Clustering practices in plant phenomics and business cycle analysis," in *Data Journeys in the Sciences*. Springer, Cham (Switzerland), 2020, pp. 79–101.

[55] X. L. Dong, A. Halevy, and C. Yu, "Data integration with uncertainty," *The VLDB Journal*, vol. 18, no. 2, pp. 469–500, 2009.

[56] A. Kimmig, A. Memory, R. J. Miller, and L. Getoor, "A collective, probabilistic approach to schema mapping using diverse noisy evidence," *IEEE Trans. Knowl. Data Eng.*, vol. 31, p. 1426, 2018.

[57] T. Rekatsinas, X. Chu, I. F. Ilyas, and C. Ré, "Holoclean: Holistic data repairs with probabilistic inference," *VLDB*, vol. 10, no. 11, 2017.

[58] D. Suciu, D. Olteanu, C. Ré, and C. Koch, "Probabilistic databases," *Synthesis lectures on data management*, vol. 3, pp. 1–180, 2011.

[59] P. Domingos and D. Lowd, "Markov logic: An interface layer for artificial intelligence." Morgan & Claypool Publishers, 2009, vol. 3.

[60] S. H. Bach, M. Broecheler, B. Huang, and L. Getoor, "Hinge-loss markov random fields and probabilistic soft logic," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 3846–3912, 2017.

[61] F. E. Petry, *Fuzzy databases: principles and applications*. Springer Science & Business Media (Germany), 2012, vol. 5.

[62] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press Cambridge, MA, USA, 2016.

[63] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 29, p. 2724, 2017.

[64] R. Cappuzzo, P. Papotti, and S. Thirumuruganathan, "Creating embeddings of heterogeneous relational datasets for data integration tasks," in *SIGMOD*, 2020, pp. 1335–1349.

[65] Y. N. Silva, W. G. Aref, and M. H. Ali, "The similarity join database operator," in *International Conference on Data Engineering (ICDE)*. IEEE, 2010, p. 892.

[66] M. Yu, G. Li, D. Deng, and J. Feng, "String similarity search and join: a survey," *Frontiers of Computer Science*, vol. 10, p. 399, 2016.

[67] M. Qu and J. Tang, "Probabilistic logic neural networks for reasoning," in *Advances in Neural Information Processing Systems*, 2019, p. 7712.

[68] E. Choi, Z. Xu, Y. Li, M. W. Dusenberry, G. Flores, Y. Xue, and A. M. Dai, "Graph convolutional transformer: Learning the graphical structure of electronic health records," in *AAAI*, 2019.

[69] C. Molnar, *Interpretable Machine Learning*. Lulu.com, 2020.

[70] A. Ghazimatin, O. Balalau, R. Saha Roy, and G. Weikum, "Prince: Provider-side interpretability with counterfactual explanations in recommender systems," in *International Conference on Web Search and Data Mining*, 2020, pp. 196–204.

[71] A. Altmann, L. Tolosi, O. Sander, and T. Lengauer, "Permutation importance: a corrected feature importance measure," *Bioinformatics*, vol. 26, p. 1340, 2010.

[72] I. G. Terrizzano, P. M. Schwarz, M. Roth, and J. E. Colino, "Data wrangling: The challenging yourney from the wild to the lake." in *CIDR*, 2015.

[73] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," in *Semantic services, interoperability and web applications: emerging concepts*. IGI Global, 2011, pp. 205–227.

Alexis Cvetkov-Iliev received an Engineering degree in Optics from the Institut d'Optique Théorique et Appliquée (Palaiseau, France) in 2019 and a M.S degree in Automation, Signal and Image Processing from Paris-Saclay University (Gif-sur-Yvette, France) in 2019.

Since 2019, he is pursuing a Ph.D degree in machine learning at Inria (French National Institute for Computer Science Research). His research focuses on embedding methods for relational data, with the aim of facilitating data integration and analysis across databases.

Alexandre Allauzen Since 2019, Alexandre Allauzen is professor at ESPCI (École Supérieure de Physique et de Chimie Industrielles de la Ville de Paris). His research affiliation is in the MILES project (Machine Intelligence and Learning Systems) of the LAMSADE (Laboratoire d'Analyse et de Modélisation de Systèmes pour l'Aide à la Décision) in Paris-Dauphine University and PSL (Paris Sciences & Lettres). His research activities focus on deep learning applied to natural language processing and more recently to Physics and different kind of structured data.

Gaël Varoquaux is a research director working on data science and health at Inria (French National Institute for Computer Science Research). His research focuses on statistical-learning tools for data science and scientific inference, with an eye on applications in health and social science. He develops tools to make machine learning easier, with statistical models suited for real-life, uncurated data, and software for data science. For example, since 2008, he has been exploring data-intensive approaches to understand brain function and mental health. He co-funded scikit-learn, one of the reference machine-learning toolboxes, and helped build various central tools for data analysis in Python. He has a PhD in quantum physics and is a graduate from Ecole Normale Superieure, Paris.

VII. APPENDIX: IMPLEMENTATION DETAILS

We describe here the estimation methods used in our empirical evaluation. Subsections A, B and C correspond to the analytical tasks of Table 3: the evolution of salary with experience, salary quantiles across jobs, and the proportion of women across jobs. Subsection D focuses on the causal inference problem of Figure 4: estimating the effect of gender on salaries, accounting for confounding factors.

A. SALARY EVOLUTION AS A FUNCTION OF EXPERIENCE

For a given job, we aim to estimate the mean salary as a function of work experience. This amounts to estimating the conditional expectation $\tau = \mathbb{E}[\text{Salary} \mid \text{Job, Experience}]$.

1) Matching & averaging

We form a group $G(j,e)$ of employees with the job j and experience level e of interest, and compute the empirical mean of the employee salaries y_i.

$$\hat{\tau}_{\text{matching}}(j,e) = \frac{1}{|G(j,e)|} \sum_{\substack{1\leq i\leq n \\ i\in G(j,e)}} y_i$$

Note that in our experiments, we estimate τ for job titles in the test set. Some of them have no equivalent in the training set and thus cannot be matched, meaning that G would be empty. In this case, we include in G all employees with the desired experience level, regardless of their jobs.

2) Embeddings & fuzzy matching

Matching and averaging provides noisy estimates when the group $G(j,e)$ contains few employees. To obtain reliable estimates in these cases, fuzzy matching averages manual matching estimates $\hat{\tau}_{\text{matching}}(j',e)$ over several jobs j', giving more weight to jobs j' that are similar to the job j of interest:

$$\hat{\tau}_{\text{fuzzy}}(j,e) = \frac{\sum_{j'\in J} \hat{\tau}_{\text{matching}}(j',e) \cdot \text{sim}(j',j)}{\sum_{j'\in J} \text{sim}(j',j)}$$

with J the set of all job titles and $\text{sim}(j',j) \geq 0$ the string similarity between the job j' and the job j of interest.

To define the string similarity $\text{sim}(j_1,j_2)$ between job titles, we encode them into vectors $\mathbf{j_1}, \mathbf{j_2}$ using a pretrained fastText model [4] and compute their cosine similarity:

$$c(j_1,j_2) = \frac{\mathbf{j_1}\cdot\mathbf{j_2}}{||\mathbf{j_1}||\,||\mathbf{j_2}||} \in [-1,1]$$

We finally obtain the similarity score by rescaling the cosine similarity into $[0,1]$, based on a threshold t that we tune to minimize cross-validation errors:

$$\text{sim}(j_1,j_2) = \begin{cases} \frac{c(j_1,j_2)-t}{1-t} & \text{if } c(j_1,j_2) \geq t \\ 0 & \text{otherwise} \end{cases}$$

We select the threshold in the following range of values: $t \in \{0.9, 0.8, 0.7, 0.6, 0.5\}$.

[4]The fastText model for english words can be downloaded here: https://fasttext.cc/docs/en/crawl-vectors.html.

3) Embedding & learning

We can estimate τ by training a machine-learning model f_θ to predict the salary of an employee given its job and experience level. Importantly, we optimize model parameters θ to minimize the mean squared error:

$$\hat{\theta} = \arg\min_\theta \left(\frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(j_i,e_i))^2 \right) \quad (1)$$

where y_i, j_i and e_i are the salary, job title and experience level of the i^{th} employee. Indeed, minimizing the mean squared error leads to estimates of the conditional expectation [37, section 1.5.5].

Once trained, we can directly query the model to estimate the mean salary for the desired job j and experience level e:

$$\hat{\tau}_{\text{learning}}(j,e) = f_{\hat{\theta}}(j,e)$$

In our experiments, we use gradient boosted regression trees[5] as machine-learning model f_θ. We also use vector representations $\mathbf{j_i}$ of the job titles as features (obtained from a pretrained fastText model) to implicitly account for entity-matching.

We also tune the learning rate $\alpha \in \{0.01, 0.03, 0.1, 0.3\}$ of the model to minimize cross-validation errors.

B. SALARY QUANTILES

We are also interested in the distribution of salaries among employees with job j. More precisely, we aim to estimate the 0.75-quantile, *i.e.* the salary $\tau(j)$ so that 75% of employees with job j earn less than $\tau(j)$.

1) Matching & averaging

To estimate this quantity, we group employees based on their jobs, and then compute the empirical 0.75-quantile of salaries $\hat{\tau}_{\text{matching}}(j)$ for each group.

2) Embeddings & fuzzy matching

We follow the same procedure that we used to estimate the mean salary given the job and experience level (see Section VII-A2), with $\hat{\tau}_{\text{fuzzy}}(j)$ being a weighted average of the "matching & averaging" estimates.

3) Embedding & learning

We can estimate $\tau(j)$ by training a machine-learning model f_θ to predict the salary of an employee given its job. To estimate quantiles, we optimize model parameters θ to minimize the **quantile loss**, instead of the mean squared error:

$$\hat{\theta} = \arg\min_\theta \left(\frac{1}{n} \sum_{i=1}^{n} \rho_\alpha(y_i - f_\theta(j_i)) \right) \quad (2)$$

$$\text{where } \rho_\alpha(x) = \begin{cases} -x\,(1-\alpha), & \text{if } x \leq 0 \\ \alpha\, x, & \text{otherwise} \end{cases}$$

with $\alpha = 0.75$ the quantile to estimate.

[5]See https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html

As before, we can directly query the model to estimate the 0.75-quantile of salaries among employees with job j:

$$\hat{\tau}_{\text{learning}}(j) = f_{\hat{\theta}}(j)$$

Again, we use gradient boosted regression trees for prediction and fastText embeddings to encode job titles. We also tune the learning rate $\alpha \in \{0.01, 0.03, 0.1, 0.3\}$ of the model to minimize cross-validation errors.

C. PROPORTION OF MEN ACROSS JOBS

We aim here to estimate the percentage $\tau(j)$ of men among employees with job j.

1) Matching & averaging

As before, we simply group employees by job titles, and compute the empirical frequency of men in each job.

2) Embeddings & fuzzy matching

We apply the same procedure as in the previous subsections.

3) Embedding & learning

We can estimate $\tau(j)$ by training a classification model f_θ to predict the gender W of an employee given its job j. The model output $f_\theta(j) = \hat{\tau}_{\text{learning}}(j)$ then estimates the probability that an employee with job j is a man. Model parameters are optimized to minimize the logistic loss:

$$\hat{\theta} = \arg\min_\theta \left(\frac{1}{n} \sum_{i=1}^{n} -W_i \log(f_\theta(j_i)) - (1 - W_i) \log(1 - f_\theta(j_i)) \right) \quad (3)$$

where W_i and j_i are the gender and job of the ith employee.

As before, we use gradient boosted trees as classification model [6] and use pretrained fastText embeddings to encode job titles. The learning rate of the model is also tuned to minimize cross-validation errors.

D. CAUSAL EFFECT OF GENDER ON SALARY

As described in Section III-B, we are interested in the *average treatment effect* (ATE) $\tau = \mathbb{E}[y(W = 1) - y(W = 0)]$: the average salary gap between a man and a woman, all else being equal. In our experiments we use the following features: job title, experience level, ethnicity and the type of employer (city, county, university, hospital). Including these features allows to compare salaries between similar employees and isolate the effect of gender.

Note that the ethnicity feature is also non-normalized: multiple variants for each ethnicity exist in the data (*e.g.* "Black", "BLK", "Black or African American"). When estimating the ATE with manual or fuzzy matching techniques, we thus had to group similar ethnicities into 7 categories. When using machine-learning models for estimation, we simply encoded ethnicities into vectors of dimension 10, using a Gamma-Poisson factorization[7] [38]. Besides, these vectors can capture nuances that would have been lost in the matching process otherwise: for instance when grouping "Mexican" with "Hispanic or Latino".

We could easily estimate the ATE if for each employee we had access to $y(1)$ and $y(0)$, *i.e.* salaries under scenario $W = 1$ (employee is a man) and $W = 0$ (employee is a woman):

$$\widehat{\tau} = \frac{1}{n} \sum_{i=1}^{n} y_i(1) - y_i(0) \quad (4)$$

Unfortunately, we either observe $y_i = y_i(1)$ or $y_i = y_i(0)$ in the data. To be able to apply Eq. 4, we can replace the unobserved salary y_i^{unobs} by an estimate.

1) Matching & averaging

A simple way to estimate the unobserved salary y_i^{unobs} is to consider the set O_i of employees with the same features as employee i but of opposite sex, and take their average salary. However, this is not always possible: some employees may have no counterpart of the opposite sex in the data. We thus consider only the set M of employees for which O_i is not empty.

$$\widehat{\tau}_{\text{matching}} = \frac{1}{|M|} \sum_{i \in M} W_i(y_i - \widehat{y_i}^{\text{unobs}}) + (1 - W_i)(\widehat{y_i}^{\text{unobs}} - y_i) \quad (5)$$

with

$$\widehat{y_i}^{\text{unobs}} = \frac{1}{|O_i|} \sum_{k \in O_i} y_k \quad (6)$$

2) Embeddings & fuzzy matching

Dismissing employees that have no counterpart of the opposite sex in the data can bias the results. To avoid this, we allow our estimate of y_i^{unobs} to include employees from similar, but non-identical jobs. For an employee i so that $O_i = \emptyset$, we consider instead the sets $O_i^{(j)}$ of employees with the same features, **except for their job title** $j \neq j_i$, and of opposite sex.

For each set $O_i^{(j)}$ we compute the average salary $\bar{y}(O_i^{(j)})$. Finally, we estimate y_i^{unobs} as a weighted average over the different $\bar{y}(O_i^{(j)})$, based on the similarity between j and j_i.

$$\widehat{\tau}_{\text{fuzzy}} = \frac{1}{n} \sum_{i=1}^{n} W_i(y_i - \widehat{y_i}^{\text{unobs}}) + (1 - W_i)(\widehat{y_i}^{\text{unobs}} - y_i) \quad (7)$$

[6] See https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingClassifier.html

[7] An implementation of this approach is available in the dirty-cat package: https://dirty-cat.github.io/stable/ (see GapEncoder)

with

$$\widehat{y}_i^{\text{unobs}} = \begin{cases} \frac{1}{|O_i|} \sum\limits_{k \in O_i} y_k & \text{if } O_i \neq \emptyset \\ \frac{\sum\limits_{j \in J} \bar{y}(O_i^{(j)}) \, \text{sim}(j, j_i)}{\sum\limits_{j \in J} \text{sim}(j, j_i)} & \text{otherwise} \end{cases} \tag{8}$$

We use the same similarity score as in section VII-A2, with a threshold $t = 0.8$.

3) Embeddings & learning

Modern causal inference tools rely on machine-learning models, for instance to estimate y_i^{unobs}. Typically, a function f_θ is trained to predict y_i given the employee covariates/features X_i (job, experience level, ...) and its gender W_i. As before, parameters θ are optimized to minimize the mean squared error. We obtain the following estimate:

$$\widehat{\tau} = \frac{1}{n} \sum_{i=1}^{n} \big[W_i (y_i - f_{\hat{\theta}}(X_i, W = 0)) + (1 - W_i)(f_{\hat{\theta}}(X_i, W = 1) - y_i) \big] \tag{9}$$

Other approaches are based on *inverse propensity weighting*. They rely on estimates of the propensity score $e(X_i) = P(W_i = 1 | X_i)$ – the probability of being a man given features X_i – to account for imbalances between men and women covariates in the ATE:

$$\widehat{\tau} = \frac{1}{n} \sum_{i=1}^{n} \frac{W_i \, y_i}{\widehat{e}(X_i)} - \frac{(1 - W_i) \, y_i}{1 - \widehat{e}(X_i)} \tag{10}$$

A machine-learning model f_θ trained to predict the gender of an employee from its covariates provides estimates of the propensity score: $f_{\hat{\theta}}(X_i) = \widehat{e}(X_i)$.

Powerful methods combines both approaches for more robustness [49]. We use such techniques in our experiments to estimate the ATE:

$$\widehat{\tau}_{\text{learning}} = \frac{1}{n} \sum_{i=1}^{n} \big[\widehat{y}_{i,1} - \widehat{y}_{i,0} + \frac{W_i}{\widehat{e}_i}(y_i - \widehat{y}_{i,1}) - \frac{1 - W_i}{1 - \widehat{e}_i}(y_i - \widehat{y}_{i,0}) \big] \tag{11}$$

Salary estimates $\widehat{y}_{i,0/1} = f_{\hat{\theta}}^{(y)}(X_i, W = 0/1)$ are obtained from the machine-learning model $f_{\hat{\theta}}^{(y)}$, trained to predict the salary y from covariates X and gender W. Similarly, propensity-score estimates $\widehat{e}_i = f_{\hat{\theta}}^{(w)}(X_i)$ are obtained from the machine-learning model $f_{\hat{\theta}}^{(w)}$, trained to predict the gender W from covariates X.

A technical subtlety is that we use a cross-fitting procedure to estimate salaries and propensity-scores [50]. Instead of fitting machine-learning models on all the data and then taking models output as estimates, we split samples in K folds and obtain estimates for each fold using models fitted on the $K - 1$ remaining folds.

As before, we use gradient boosted trees models. Their learning rates $\in$ [0.1, 0.3, 0.5] and maximum depths $\in$ [8, 12, None] are tuned to minimize cross-validation errors.