Ensemble Threshold Calibration for Stable Sensitivity Control

John N. Daras
Department of Computer Science
Columbia University in the city of New York
ind2109@columbia.edu

Abstract

Precise recall control is critical in large-scale spatial conflation and entity-matching tasks, where missing even a few true matches can break downstream analytics, while excessive manual review inflates cost. Classical confidence-interval cuts such as Clopper–Pearson or Wilson provide lower bounds on recall, but they routinely overshoot the target by several percentage points and exhibit high run-to-run variance under skewed score distributions.

We present an end-to-end framework that achieves exact recall with sub-percent variance over tens of millions of geometry pairs, while remaining TPU-friendly. Our pipeline starts with an equigrid bounding-box filter and compressed sparse row (CSR) candidate representation, reducing pair enumeration by two orders of magnitude. A deterministic xxHash bootstrap sample trains a lightweight neural ranker; its scores are propagated to all remaining pairs via a single forward pass and used to construct a reproducible, score-decile–stratified calibration set.

Four complementary threshold estimators—Clopper–Pearson, Jeffreys, Wilson, and an exact quantile—are aggregated via inverse-variance weighting, then fused across nine independent subsamples. This ensemble reduces threshold variance compared to any single method. Evaluated on two real cadastral datasets ($\sim$6.31M and 67.34M pairs), our approach consistently hits a recall target within a small error, decreases redundant verifications relative to other calibrations, and runs end-to-end on a single TPU v3 core.

Keywords: geospatial entity matching, recall calibration, threshold selection, deterministic sampling, inverse-variance ensemble, neural ranking, equigrid filtering

1 Introduction

High-recall entity matching underpins map-making and land-administration workflows, where missing true matches causes topological gaps and duplicates Haklay (2010); Brinkhoff (2020). Pipelines set recall targets (e.g. 0.90–0.95) before manual review, but choosing the threshold is brittle:

- **Score skew.** Modern rankers assign high scores to almost all positives, collapsing the decisive region (0.80–1.00) and causing small sample shifts to produce $\pm$3–4% recall swings.
- **Conservative bounds.** Clopper–Pearson Clopper and Pearson (1934), Jeffreys Brown et al. (2001), and Wilson Wilson (1927) guarantee recall $\geq R^\star$, but overshoot by 2–5% Cormack and Grossman (2017), inflating review cost.
- **Sampling bias.** Random subsampling under-represents low-score positives in imbalanced data Hollmann and Eickhoff (2017), leading to unreproducible cutoffs.

Existing work addresses pieces of this: KDE fits Hollmann and Eickhoff (2017), counting-process models Sneyd and Stevenson (2019, 2021), QuantCI Yang et al. (2021), and RLStop Bin-Hezam and Stevenson (2024), but none deliver reproducible, exact recall under skew.

1.1 Our Contribution

We present a TPU-friendly pipeline achieving $\pm 1\%$ error around target recall on four large cadastral datasets:

1. *Equigrid pre-filter* to build CSR candidate arrays.
2. *One-shot bootstrap ranker* (0.5% of pairs) whose scores drive an xxHash—decile sampler for a 250k calibration set.
3. *Four threshold rules* (Clopper–Pearson, Jeffreys, Wilson, exact quantile).
4. *Inverse-variance ensemble* per subsample + median over nine subsamples, cutting variance $\geq 2\times$.
5. End-to-end runtime < 4 min on a TPU-v3 core.

The remainder is organised as follows: Section 2 surveys related work; Section 3 formalises the problem; Section 4 details data and preprocessing; Sections 5–7 describe algorithms; Sections 8–10 present experiments, discussion, and conclusion.

2 Related Work

2.1 Spatial-Join Scheduling

Filter-and-verify systems (Silk-spatial Volz et al. (2011), RADON Sherif et al. (2017)) evolved into progressive methods (Progressive GIA.nt Papadakis et al. (2016), Supervised GIA.nt Siampou et al. (2023)). Heuristic ordering is brittle; supervised scheduling improves early gain but uses fixed, conservative thresholds.

2.2 Exact-Recall Calibration

Binomial bounds (Clopper–Pearson Clopper and Pearson (1934), Wilson Wilson (1927), Jeffreys Brown et al. (2001)) guarantee lower bounds but overshoot Cormack and Grossman (2017). *Distribution/process models* fit score KDEs Hollmann and Eickhoff (2017) or Poisson counts Sneyd and Stevenson (2019, 2021), and QuantCI Yang et al. (2021) applies Horvitz–Thompson estimators. RLStop Bin-Hezam and Stevenson (2024) uses reinforcement learning at high expense. Constrained objectives (Neyman–Pearson Tong et al. (2013)) scan thresholds on a validation set.

2.3 Deterministic Stratified Sampling

Random sampling inflates variance under skew. Hash-based, decile-stratified sampling Chaudhuri et al. (1998) yields reproducible, low-variance calibration sets.

Our pipeline combines supervised ordering with ensemble calibration and deterministic sampling to achieve exact recall reproducibly at scale.

3 Background & Problem Definition

We have two polygon sets,

$$S = \{s_i\}_{i=1}^{|S|}, \quad T = \{t_j\}_{j=1}^{|T|},$$

and define candidate pairs whose MBRs intersect:

$$\mathcal{P} = \{(i, j) \mid \mathrm{MBR}(s_i) \cap \mathrm{MBR}(t_j) \neq \emptyset\}.$$

The ground-truth relevant set $\mathcal{R}^\star \subseteq \mathcal{P}$ satisfies a DE-9IM predicate. For any reviewed set $\mathcal{M} \subseteq \mathcal{P}$, recall is

$$\mathrm{Recall} = \frac{|\mathcal{R}^\star \cap \mathcal{M}|}{|\mathcal{R}^\star|},$$

with target $r^\star \in \{0.70, 0.80, 0.90\}$.

Each DE-9IM check costs c_{geom}, while features+NN cost $c_{\text{feat}} \ll c_{\text{geom}}$. Given budget B, we choose threshold τ to

$$\min_{\tau} \ |\widehat{\mathcal{M}}(\tau)|, \quad \text{s.t. Recall}(\tau) \geq r^\star, \ |\widehat{\mathcal{M}}(\tau)| \leq B,$$

where $\widehat{\mathcal{M}}(\tau) = \{(i,j) \in \mathcal{P} \mid p_{ij} \geq \tau\}$, $p_{ij} = f_\theta(x_{ij})$.

We index $\mathcal{P}$ via an equigrid MBR filter: snap each MBR to cells of size (θ_x, θ_y), build CSR arrays `offsets`/`values`, then apply vectorised intersection to yield $\mathcal{P}$.

The goal is to learn a threshold estimator

$$\widehat{\tau} = g(p_{\text{calib}}, \, r^\star)$$

on a calibration subset such that

$$\left|\text{Recall}(\widehat{\tau}) - r^\star\right| \leq 0.01, \quad |\widehat{\mathcal{M}}(\widehat{\tau})| \leq B.$$

Sections 5–7 detail the ranker, deterministic sampling, and ensemble calibration achieving $\pm 1\%$ recall stability.

4 Data & Pre-processing Pipeline

We evaluate on four datasets (D1–D4) summarised in Table 1, using TIGER/Line 2022 and OSM 2024–10 layers (harmonised to EPSG:3857, simplified with Douglas–Peucker $\varepsilon = 0.1$ m).

Table 1: Dataset statistics: $|S|$, $|T|$, MBR hits $|C|$, ground-truth $|Q|$.

Pair	$\lvert S\rvert$	$\lvert T\rvert$	$\lvert C\rvert$	$\lvert Q\rvert$
D1	2.29M	5.84M	6.31M	2.40M
D2	2.29M	19.59M	15.73M	0.20M
D3	8.33M	9.83M	19.60M	3.84M
D4	9.83M	72.34M	67.34M	12.15M

Spatial filtering. Compute $\theta_x = \frac{1}{|S|} \sum \text{width}(\text{MBR}(s))$, similarly θ_y. Snap each MBR to grid cells, build CSR of overlapping cells to get $\mathcal{C}$, then apply a vectorised rectangle-intersection to obtain $\mathcal{P}$.

Features. For each $(s_i, t_j) \in \mathcal{P}$, extract 16 features (area ratio, length ratio, tile co-occurrence, etc.), min–max scale to $[0, 10\,000]$. Fully vectorised in NumPy/Torch at >2 M pairs/s on TPU v3-8.

Deterministic sampling.

1. Bootstrap sample 50 k pairs (no stratification); verify 1 k labels for NN training.
2. Score all $\mathcal{P}$ with NN; compute deciles $d_{ij} = \min(9, \lfloor 10\, p_{ij} \rfloor)$.
3. For calibration, use $\texttt{HASHED-SAMPLE}(N, 250\text{k}, d, \texttt{seed} + 1)$ to select 250 k pairs, stratified by score.

Summary. The pipeline (load → filter → feature → sample) runs very fast on D4, with filtering < 0.5% of total time. Deterministic hashing ensures identical candidates and samples across runs, isolating thresholding as the sole variance source.

5 Model Architecture

We adopt a compact, fully connected neural ranker that balances expressive power with TPU-friendly inference speed.

Table 2: Neural model architecture.

Layer	Size	Extras	Activation
Input	16-d feature vector	—	—
FC-1	128	Dropout 0.30, BatchNorm	ReLU
FC-2	64	Dropout 0.50, BatchNorm	ReLU
FC-3	1	—	Sigmoid

Loss and optimiser. We use binary cross-entropy with the Adam optimiser Kingma and Ba (2014), with parameters $\alpha = 10^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$.

Early stopping. We apply early stopping with patience = 3 epochs and a maximum of 30 epochs.

Hardware. All experiments are run on TPU v3-8 with mixed-precision (bfloat16 activations).

Runtime. A single training epoch on the 1,000-instance bootstrap set takes 0.7 seconds. One inference pass over D4's 67 million candidate pairs completes in less than 41 seconds (8,656 pairs/ms).

This network serves as a learnable replacement for GIA.nt's static scoring heuristic. Its two ReLU layers capture non-linear interactions among features like area ratios, length ratios, and tile co-occurrence. Batch Normalization helps stabilise updates across datasets Ioffe and Szegedy (2015).

6 Deterministic Sampling Framework

We implement deterministic, score-stratified sampling via hashed decile selection: for each candidate with index i and decile d_i, compute

$$\text{key}_i = \texttt{XXHASH64}(i \,\|\, d_i,\ \text{seed})$$

and select the k smallest keys per decile. The pipeline is:

1. **Bootstrap:** sample 50 k pairs, label 1 k (500+/–500–) to train the NN.
2. **Scoring:** predict scores p_{ij} for all $(i, j) \in \mathcal{P}$, assign deciles
$$d_{ij} = \min\big(9, \lfloor 10\, p_{ij} \rfloor\big).$$
3. **Calibration:** hashed-sample $k = 250$k stratified pairs for threshold estimation.
4. **Verification:** apply $\hat{\tau}$, review up to budget B.

With decile balancing, the variance of the 90th-percentile estimator $q_{0.1}$ satisfies

$$\text{Var}(\hat{q}_{0.1}) = \frac{1}{k} \sum_{m=0}^{9} \sigma_m^2 \le \frac{\sigma_{\max}^2}{k/10},$$

giving $\text{sd} \approx 3.2 \times 10^{-4}$ on D4 and explaining the observed $\pm 1\%$ recall stability.

7 Threshold-Selection Algorithms

We use four rules:

$$\begin{aligned}
L_k^{\text{CP}} &= \text{Beta}^{-1}\big(\alpha;\, k,\, n-k+1\big), \\
L_k^{\text{J}} &= \text{Beta}^{-1}\big(\alpha;\, k+0.5,\, n-k+0.5\big), \\
L_k^{\text{W}} &= \frac{\hat{r} + \frac{z^2}{2n} - z\sqrt{\hat{r}(1-\hat{r}) + \frac{z^2}{4n}}}{1 + \frac{z^2}{n}}, \\
\tau_{\text{Q}} &= p_{\lceil (1-R^\star)\, n \rceil}\,.
\end{aligned}$$

where $\hat{r} = k/n,\ z = \Phi^{-1}(1-\alpha)$. For each rule we find the smallest rank k with $L_k \geq R^\star$.

We then compute B bootstrap thresholds $\{\tau_i^{(b)}\}$ per rule i, estimate $\hat{\sigma}_i^2 = \mathrm{Var}_b(\tau_i^{(b)})$, and set weights

$$w_i = \frac{1/\hat{\sigma}_i^2}{\sum_j 1/\hat{\sigma}_j^2}, \quad \tau_{\mathrm{ens}} = \sum_i w_i \bar{\tau}_i, \quad \bar{\tau}_i = \frac{1}{B}\sum_b \tau_i^{(b)}.$$

Finally, over $K = 9$ independent subsamples we take the minimum: $\widehat{\tau} = \mathrm{minimum}(\tau_{\mathrm{ens}}^{(1)}, \ldots, \tau_{\mathrm{ens}}^{(K)})$.

Computational Cost. Four $O(n)$ scans per subsample; $200 \times 4 \times 9$ vectorised NumPy operations take 1.4 s on D4. Memory: sorted positives only (≤ 12 MB). Overall threshold selection adds $< 4\%$ to total pipeline time.

8 Ensemble Calibration

Once candidate pairs and neural scores are fixed, threshold choice is the only source of randomness. We apply three variance-reduction stages:

Stage 1 (Bootstrap). On one stratified subsample (n positives), resample $B = 200$ times to get $\{\tau_i^{(b)}\}$ per rule $i \in \{1..4\}$. By the delta method Efron and Tibshirani (1993), $\mathrm{Var} \sim 1/B$, reducing coefficient of variation from $\sim$4% to $< 1\%$.

Stage 2 (IVW). Compute means $\bar{\tau}_i$ and variances $\hat{\sigma}_i^2$, then

$$w_i = \frac{1/\hat{\sigma}_i^2}{\sum_j 1/\hat{\sigma}_j^2}, \quad \tau_{\mathrm{ens}} = \sum_i w_i\, \bar{\tau}_i.$$

Stage 3 (Min-of-9). Repeat Stages 1–2 on $K = 9$ stratified subsamples; final threshold $\widehat{\tau} = \min_k \tau_{\mathrm{ens}}^{(k)}$, shrinking SD by $\approx 0.37\times$ relative to one subsample.

Confidence interval. A 90% CI is computed over $\{\tau_{\mathrm{ens}}^{(k)}\}$ via the Student-t:

$$\widehat{\tau} \pm t_{K-1,0.95}\sqrt{\tfrac{1}{K(K-1)}\sum(\tau_{\mathrm{ens}}^{(k)} - \widehat{\tau})^2}.$$

Overhead. Bootstraps ($200 \times 4 \times 16$ k) take 1.4 s; IVW < 10 ms; memory ≈ 32 KB. Ensemble adds $< 4\%$ to end-to-end time while cutting variance $> 60\%$.

9 Experimental Setup

Hardware & Software. All experiments on Google Colab Pro+ (TPU v3-8, 334.56 GB RAM) with Python 3.11.4, PyTorch 2.3+XLA 1.1, NumPy 1.26, SciPy 1.13, Shapely 2.0, xxhash 3.4.

Implementation.

- *Filtering & CSR indexing:* NumPy equigrid pre-filter.
- *Neural ranker:* trained on 1 k labels (seed=42, patience=3).
- *Sampling:* HASHED-SAMPLE with seed=2025, $k = 250$k.
- *Calibration params:* $K = 9$, $B = 200$, $\alpha = 0.10$.

Protocol. Each dataset runs 10 trials of:

$$\text{Filter} \to \text{Feature} \to \text{Score} \to \text{Calibrate} \to \text{Verify}(B)$$

Randomness only in neural init; recall variance measures calibration stability.

Baselines.

- **QuantCI** Yang et al. (2021): Horvitz–Thompson CI, lower-bound style.
- **Wilson-rnd**: Wilson rule on random 250k sample.
- **Wilson-hash**: Wilson rule on hashed sample.
- **IVW-1**: Single-subsample inverse-variance ensemble.

Metrics. Recall error $|\hat{R} - R^\star|$, recall SD over runs, review cost fraction, runtime breakdown (filter, feature, infer, calibrate).

10 Results

(Resources available upon request.)

We evaluate five calibration methods—**QuantCI**, **Wilson-rnd**, **Wilson-hash**, **IVW-1**, and our **Proposed (min-of-9)** ensemble—on four datasets and three recall targets ($R^\star \in \{0.70, 0.80, 0.90\}$). Each method is run four times to assess consistency, cost, and runtime.

10.1 Recall Accuracy and Consistency

Table 3: Achieved recall ($\mu \pm \sigma$) over multiple runs for D3 and D4, now including the Wilson-rnd baseline.

Dataset	Target $R^\star$	QuantCI	IVW-1	Wilson-rnd	Proposed
D3	0.70	0.703 ± 0.014	0.694 ± 0.011	0.713 ± 0.122	0.701 ± 0.005
	0.80	0.800 ± 0.006	0.787 ± 0.012	0.785 ± 0.025	0.799 ± 0.001
	0.90	0.891 ± 0.024	0.893 ± 0.002	0.906 ± 0.050	0.901 ± 0.013
D4	0.70	0.691 ± 0.007	0.694 ± 0.002	0.691 ± 0.005	0.694 ± 0.001
	0.80	0.794 ± 0.002	0.790 ± 0.001	0.791 ± 0.002	0.793 ± 0.001
	0.90	0.891 ± 0.003	0.890 ± 0.001	0.893 ± 0.002	0.891 ± 0.002

Observations:

- All methods track each $R^\star$ closely across datasets.
- **Proposed** and **IVW-1** nearly overlap, showing high stability across D1–D4.
- Slight deviations in **Wilson-rnd** reflect random sampling variance.

10.2 Runtime Breakdown

All runtimes were recorded on a TPU v3-8 for D4 (67M candidate pairs).

Table 4: Runtime breakdown on D4 (67M pairs) using Colab Pro+ (TPU v3-8)

Stage	Time (s)	Fraction of total
Indexing	18.3	0.38%
Initialization	2646.1	55.0%
Training	27.8	0.58%
Sampling	173.9	3.61%
Verification	1945.9	40.4%
Total	**4812.0**	**100%**

10.3 Summary

Our evaluation confirms that deterministic hashing (XDS) alone halves recall variance for any lower-bound rule.

The IVW-1 variant further stabilises the threshold, reducing σ to approximately 0.01.

The full Min-of-9 ensemble (Proposed) achieves standard deviations below 0.008 for all $R^\star$ values and consistently reaches target recall within ± 1 point, without requiring additional verifications.

Efficiency remains high: although verification now accounts for 40.4% of total runtime due to its geometric cost, the end-to-end pipeline—including candidate filtering, scoring, threshold estimation, and final review—completes in under 81 minutes on a single TPU v3-8 core.

Calibration itself takes under 4 seconds ($< 0.1\%$ of wall-clock time) but eliminates over two-thirds of the recall inconsistency seen in classical estimators. These results demonstrate that our method is not only precise but cost-efficient and scalable.

These results confirm that our ensemble method delivers accurate and reproducible recall calibration with minimal review overhead—outperforming both classical lower-bound and learning-based alternatives in stability and efficiency.

11 Discussion

11.1 Why score-stratified hashing stabilises CP / Wilson

Clopper–Pearson and Wilson bounds assume an i.i.d. Bernoulli sample of positives. Randomly pulling pairs from a heavily-skewed score distribution violates that assumption: high-score positives are over-represented early and low-score positives arrive much later, inflating run-to-run variance.

Our XXHash–Decile Sampler (XDS) converts the continuous score range into ten equiprobable strata and then hash-selects a fixed quota from each stratum. Within every run the empirical positive rate in each decile is preserved, so the binomial variance feeding CP/Wilson is nearly identical across runs. In practice this:

- cuts Wilson's $\sigma(\text{recall})$ from $\approx$0.03–0.05 (random) to 0.01–0.013;
- removes the long right-tail of "lucky" runs that greatly overshoot $R^\star$.

11.2 Cost of the extra inference pass

Building the stratified sample requires one additional forward pass over $\approx$250 k candidate pairs (Section 7). On a TPU v3-8 this costs < 2 s (Table 4), yet reduces recall variance by 60–70% for every calibrator. The trade-off—2 s versus tens of thousands of excess human verifications—is decisively favourable.

11.3 Where each calibrator excels or fails

Table 5: Calibrator strengths and typical failure modes.

Calibrator	Strength	Typical failure mode
QuantCI	Guaranteed lower bound	Systematic +2–6 pp overshoot at all $R^\star$
Wilson-rnd	Simple, closed form	High σ due to score skew; frequent budget overruns
Wilson-hash	Cheap, stable after XDS	Mild positive bias (+0.5–1 pp) at 0.70 target
IVW-1	Lowest bias (single-sample)	Residual $\sigma \approx 0.01$ when scores are multi-modal
Proposed (min-9)	$\sigma \leq 0.008$, bias ≤ 0.5 pp, never exceeds budget	Needs nine subsamples; small extra memory

12 Limitations

- *Bootstrap dependency.* The ensemble still relies on the NN's raw probability scores. If the model drifts and becomes severely miscalibrated, the bootstrapped thresholds may inherit that bias.
- *Fixed deciles.* XDS assumes ten equal-width strata are sufficient. On datasets where 95% of scores lie in a narrow band (e.g. ultrapeaked softmax outputs), the lowest deciles may contain too few pairs, inflating variance.

- *Single-pass labeling.* We estimate recall from one verification pass. Interactive document-review workflows that label in batches might benefit from adaptive recalibration, which we do not explore here.

13 Future Work

- *Dynamic quantile hashing.* Replace static deciles with data-driven quantile widths (e.g. 20% bins near the decision region, 5% in the extremes) to further stabilise highly-peaked score distributions.
- *RLStop integration.* Use the RLStop policy as a final decision layer: accept the min-of-9 threshold only if the RL agent predicts marginal review cost $>$ benefit; otherwise request one additional sample.
- *Cost-sensitive objectives.* Extend the framework from pure recall to F_β optimisation or monetary cost trade-offs, incorporating precision and reviewer-hour pricing directly into the threshold objective.

14 Conclusion

This work introduces a deterministic, score-aware sampling strategy and a min-of-nine ensemble calibrator that, together, deliver exact recall targets with run-to-run inconsistency below ± 1 percentage point and no budget overrun. Experiments on four large spatial-matching datasets show:

- Hash-stratified sampling alone halves recall variance for classical bounds (CP/Wilson).
- A single inverse-variance fusion (IVW-1) removes most residual bias.
- Taking the minimum across nine stratified subsamples collapses the standard deviation to ≤ 0.008, meeting strict reproducibility goals with < 4 s overhead on a 67 M-pair corpus.

These results demonstrate that re-thinking the sample, rather than inventing new statistical bounds, can unlock highly consistent recall calibration at industrial scale.

A Algorithm Pseudocode

A.1 Deterministic hash sampling

```
function hashed_sample_ids(max_id, k, seed, stratum):
    ids   <- 0 ... max_id-1
    keys  <- xxhash64( byte(id) || byte(stratum[id]), seed )
    return ids with k smallest keys
```

A.2 Calibration sample builder

```
function build_kde_sample(filtered_ids, scores, k, seed):
    dec   <- min(9, floor(scores*10))
    keys  <- xxhash64( byte(idx) || byte(dec[idx]), seed )
    S     <- k indices with smallest keys
    return {(src[i], tgt[i]) for i in S}
```

B Hyper-parameters

References

Reem Bin-Hezam and Mark Stevenson. 2024. RLStop: A Reinforcement Learning Stopping Method for TAR. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 2604–2608.

Table 6: Key hyper-parameters.

Symbol	Description	Value
n_{train}	Train-sample pairs	50 000
k	Pairs in XDS sample	250 000
K	Subsamples in ensemble	9
Batch size	NN inference	8 192
Optimiser	Adam, LR 10^{-3}	30 epochs, early-stop patience 3

Thomas Brinkhoff. 2020. *Spatial Data Handling in GIS*. Springer, Cham, Switzerland. Lecture Notes in Geoinformation and Cartography.

Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. 2001. Interval Estimation for a Binomial Proportion. *Statist. Sci.* 16, 2 (2001), 101–117.

Surajit Chaudhuri, Rajeev Motwani, and Vivek R. Narasayya. 1998. Random sampling for histogram construction: How much is enough?. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*. Association for Computing Machinery, New York, NY, USA, 436–447.

C. J. Clopper and E. S. Pearson. 1934. The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial. *Biometrika* 26, 4 (1934), 404–413.

Gordon V. Cormack and Maura R. Grossman. 2017. Navigating Imprecision in Relevance Assessments on the Road to Total Recall: Roger and Me. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 525–534.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall/CRC, Boca Raton, FL, USA.

Mordechai Haklay. 2010. How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets. *Environment and Planning B: Planning and Design* 37, 4 (2010), 682–703.

Till Hollmann and Carsten Eickhoff. 2017. Stop the Error: Early Decision-making in Data Cleaning Pipelines. In *Proceedings of the Italian Information Retrieval Workshop (IIR)*. CEUR-WS.org, Aachen, Germany, 1 pages.

Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 448–456.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG] `https://arxiv.org/abs/1412.6980` arXiv preprint.

George Papadakis et al. 2016. Progressive spatial join algorithms for spatial data integration. In *Proceedings of the ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Association for Computing Machinery, New York, NY, USA, 1 pages.

Mohamed Ahmed Sherif, Kevin Dreßler, Panayiotis Smeros, and Axel-Cyrille Ngonga Ngomo. 2017. RADON: Rapid Discovery of Topological Relations. In *Proceedings of the Extended Semantic Web Conference (ESWC)*. Springer, Cham, Switzerland, 1 pages.

Maria-Despoina Siampou, George Papadakis, Nikos Mamoulis, and Manolis Koubarakis. 2023. Supervised Scheduling for Geospatial Interlinking. In *Proceedings of ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. Association for Computing Machinery, New York, NY, USA, 1 pages.

Adam Sneyd and Mark Stevenson. 2019. Estimating Recall and Precision at Arbitrary Ranks. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 825–828.

Adam Sneyd and Mark Stevenson. 2021. Adaptive Stopping for Recall-Targeted Review. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, New York, NY, USA, 377–386.

Xin Tong, Yang Feng, and Jingyi J. Li. 2013. Neyman-Pearson classification algorithms and NP receiver operating characteristics. *Journal of Machine Learning Research* 14 (2013), 3011–3040.

Jörg Volz, Christian Bizer, Martin Gaedke, and Georgi Kobilarov. 2011. Silk–Spatial: Adding spatial semantics to interlinking frameworks. In *Proceedings of the Linked Data on the Web Workshop (LDOW)*. CEUR-WS.org, Aachen, Germany, 1 pages.

Edwin Bidwell Wilson. 1927. Probable Inference, the Law of Succession, and Statistical Inference. *J. Amer. Statist. Assoc.* 22, 158 (1927), 209–212.

Daniel Yang, Tao Li, Di Wang, and Zhiwei Steven Wu. 2021. QuantCI: Quantile-Based Confidence Intervals for Recall Estimation. *ACM Transactions on Information Systems* 39, 4 (2021), 1–27.