

Classifying Multilingual Stemmers by Pareto Strata: Error Regimes in Pairwise Lexical Conflation

Leo Galambos

Abstract—Scalar stemmer rankings cannot distinguish unconditional inferiority from a genuine exchange between missed and false conflations. We introduce a matrix-conditional taxonomy in the two-objective space of Paice-style under-conflation and over-conflation rates. The taxonomy separates aggregate error-vector equivalence, recursive Pareto-dominance strata, within-stratum conflation regimes, raw pair-cost support intervals, and multi-output trajectories. A checked 314-scenario snapshot covers 20 predominantly inflectional, UniMorph-derived reference resources and two nested processing modes. Its 190 deterministic implementation rows yield 174 distinct aggregate error vectors and four Pareto strata containing 40, 97, 27, and 10 vectors. In the primary complete-resource view, Radixor is the sole first-stratum system in all 19 comparative languages and strictly dominates 75 comparator implementation rows representing 69 distinct comparator error vectors; the nested lowercase-only view reproduces the same conclusion. The second stratum contains recall edges, specificity edges, linearly supported compromises, and efficient points unsupported by any linear pair-cost ratio. Pareto depth is explicitly relative to the candidate set: after deleting one non-Radixor error vector, 551 of 592 retained-vector assignments remain unchanged and 41 improve by one stratum. Nested-mode stratum identity is preserved for 88 of 95 implementation-language units. Because all-candidate relations contain the primary relation, candidate expansion can only reduce under-conflation while preserving or increasing over-conflation; empirically it yields 14 Pareto improvements and 17 trade-offs. The framework classifies measured agreement with declared lexical policies rather than universal linguistic quality.

Index Terms—stemming, information retrieval, lexical conflation, Pareto dominance, Pareto depth, under-stemming, over-stemming, multi-output morphology

I. INTRODUCTION

STEMMING determines which word forms a retrieval or text-processing system treats as equivalent. Two primary relation errors follow. *Under-conflation* leaves forms apart although a reference policy relates them; *over-conflation* joins forms that the policy keeps separate. A scalar score necessarily encodes a preference between these risks.

This scalarization hides a fundamental distinction. If system a has no more under-conflation and no more over-conflation than system b , with a strict advantage on at least one axis, then b is inferior for every monotone loss over those axes. If each system is better on one axis, however, the choice is policy-dependent. A leaderboard alone does not reveal which case occurred.

Paice’s under-stemming index (UI), over-stemming index (OI), stemming weight, and error-rate measures established a two-dimensional view of stemming errors [1], [2]. Later work studied algorithm strength and output similarity [3]. Strength

is not quality: two systems can have the same OI/UI ratio while one has lower UI and lower OI. We therefore classify measured stemmer error profiles through Pareto dominance before applying any scalar preference.

Pareto-front evaluation is already used in NLP when effectiveness and efficiency are competing objectives [4], and recursive non-dominated sorting is known as Pareto depth analysis in other classification settings [5]. Our contribution is narrower: to the best of our knowledge, stemming evaluation has not previously used recursive UI–OI Pareto depth to define matrix-conditional strata and conflation regimes across a multilingual comparison.

For a fixed language, reference resource, processing mode, implementation version, and output policy, the proposed class is

$$\mathcal{C}_M(s) = (Q_M(s), G_M(s), I_M(s)), \quad (1)$$

where Q_M is Pareto stratum, G_M is within-stratum regime, and I_M is the interval of pair-cost ratios supporting the point under a linear policy. The class is deliberately conditional on matrix M ; it is not an immutable label for an algorithm across all languages and tasks.

The contributions are:

- 1) an error-vector and recursive Pareto-stratum taxonomy that separates dominance depth from conflation orientation;
- 2) pair-cost support intervals with a direct interpretation as the relative cost of one missed versus one false reference pair;
- 3) a 20-resource map of strata, regimes, implementation-category profiles, dominance margins, and candidate-set sensitivity; and
- 4) a monotonic trajectory analysis for deterministic versus all-candidate output policies.

The evaluation uses reference relations derived from the same predominantly inflectional resource family that trains Radixor. Its Radixor result is therefore *within-reference*; it does not establish universal superiority under independently defined derivational, contextual, or domain-specific policies.

II. BACKGROUND AND RELATED WORK

A. Stemming as Pairwise Conflation

Stemming has long been viewed as a conflation or clustering operation rather than only string editing [6]–[8]. Rule-based, dictionary-based, and learned approaches implement different normalization scopes and inductive biases [9]–[11]. Hull showed why aggregate retrieval results can conceal stemming trade-offs [12]; CISTEM later demonstrated that even one language can admit multiple defensible reference policies [13].

We use *stemming* in the information-retrieval sense of context-free lexical normalization. The UniMorph-derived rows provide lemma-like canonical families, but the evaluated systems do not perform sentence-conditioned lemma disambiguation or return morphosyntactic analyses. The present reference relations are predominantly inflectional; they are not a complete model of derivational or semantic relatedness.

For a set of distinct forms W , the reference relation asks whether two forms share at least one declared family. A deterministic stemmer predicts a relation by output equality. UI is the false-negative rate among reference-positive pairs; OI is the false-positive rate among reference-negative pairs. These are type-level structural diagnostics, not direct estimates of retrieval precision or recall without frequencies, collections, queries, and relevance judgments.

B. Strength, Dominance, and Pareto Depth

Paice's stemming weight characterizes the relative balance between over- and under-stemming. Frakes and Fox measure strength and output similarity [3]. Neither determines Pareto dominance. For example,

$$a = (UI = 0.05, OI = 0.001), \quad (2)$$

$$b = (UI = 0.10, OI = 0.002) \quad (3)$$

have the same OI/UI ratio, but a is better on both axes.

Non-dominated sorting recursively removes the current Pareto frontier and assigns deeper ranks [14]. Pareto depth has also been used to classify multicriteria observations outside optimization [5]. Here the measured outputs are fixed; sorting is an analytical taxonomy, not a training algorithm.

A deterministic stemmer induces a partition of a signed reference graph. Standard correlation clustering minimizes raw disagreements $FP + FN$ [15]. Balanced accuracy instead corresponds to

$$L_{\text{bal}} = \frac{1}{2}UI + \frac{1}{2}OI = 1 - BA, \quad (4)$$

which class-balances positive and negative errors. Pareto dominance precedes either scalarization.

III. PAIRWISE REFERENCE FRAMEWORK

A. Reference and Predicted Relations

Let $\mathcal{F} = \{F_1, \dots, F_m\}$ be resource-defined reference families and

$$W = \bigcup_{j=1}^m F_j \quad (5)$$

the distinct surface-form population. A form is represented once even if it belongs to several families. For distinct $u, v \in W$,

$$R_{\mathcal{F}}(u, v) = \mathbf{1}[G(u) \cap G(v) \neq \emptyset], \quad (6)$$

where $G(w)$ is the set of families containing w . A deterministic stemmer s predicts

$$R_s(u, v) = \mathbf{1}[s(u) = s(v)]. \quad (7)$$

The confusion counts are

$$TP = |\{(u, v) : R_{\mathcal{F}} = 1 \wedge R_s = 1\}|, \quad (8)$$

$$FN = |\{(u, v) : R_{\mathcal{F}} = 1 \wedge R_s = 0\}|, \quad (9)$$

$$FP = |\{(u, v) : R_{\mathcal{F}} = 0 \wedge R_s = 1\}|, \quad (10)$$

$$TN = |\{(u, v) : R_{\mathcal{F}} = 0 \wedge R_s = 0\}|. \quad (11)$$

We use

$$UI = \frac{FN}{TP + FN}, \quad OI = \frac{FP}{TN + FP}. \quad (12)$$

Reference negatives use a closed-world convention: no recorded shared family means “separate” for this evaluation. An FP is therefore disagreement with the resource policy, not necessarily a universal linguistic error.

B. Output Policies and Exact Counting

PRIMARY_OUTPUT uses one deterministic production output and defines a partition. ALL_CANDIDATES relates two forms when their returned candidate sets intersect; this relation can be non-transitive. ANY_CANDIDATE is a pair-specific optimistic bound whose candidate choice may differ between pairs, so it does not represent one globally realizable, reference-independent system output.

The counters aggregate output frequencies globally and within families, then subtract reference relations repeated across several families. Candidate-aware evaluation canonicalizes candidate-set signatures and uses an inverted candidate-to-signature index. All pair arithmetic is checked, and the optimized implementation agrees with an independent quadratic oracle on 150 fixed-seed randomized non-exclusive covers.

Proposition 1 (Candidate-expansion monotonicity). *Assume every candidate set contains its deterministic primary output. Then*

$$R_{\text{PRIMARY}} \subseteq R_{\text{ALL}}, \quad (13)$$

and consequently

$$UI_{\text{ALL}} \leq UI_{\text{PRIMARY}}, \quad OI_{\text{ALL}} \geq OI_{\text{PRIMARY}}. \quad (14)$$

Proof. If two primary outputs are equal, that shared output belongs to both candidate sets, so their intersection is non-empty. Expanding the predicted-positive relation cannot create a new FN and cannot remove an existing FP. $\square$

Thus candidate activation can yield a Pareto improvement only by lowering UI while leaving OI unchanged; otherwise a UI improvement is purchased with additional over-conflation.

IV. PARETO-STRATUM TAXONOMY

A. Error-Vector Equivalence and Dominance

For fixed matrix M , define the aggregate error vector

$$\mathbf{e}_M(s) = (UI_M(s), OI_M(s)). \quad (15)$$

Two implementations are *error-vector equivalent* when

$$s \stackrel{e}{\sim}_M t \iff \mathbf{e}_M(s) = \mathbf{e}_M(t). \quad (16)$$

This is equivalence only with respect to aggregate (FN, FP) counts. The implementations may make different errors on

individual pairs; the snapshot does not retain pair identities needed to claim output or error-assignment equivalence.

Within one matrix all systems share the same positive and negative denominators. Equality and dominance can therefore be decided exactly from integer (FN, FP) counts. For minimization, s dominates t , written $s \prec_M t$, if

$$UI_M(s) \leq UI_M(t), \quad OI_M(s) \leq OI_M(t), \quad (17)$$

with at least one strict inequality.

Proposition 2 (Policy-independent inferiority). *If $s \prec_M t$, every loss strictly increasing under the product order on (UI, OI) prefers s to t .*

Proof. Both coordinates of s are no larger and at least one is smaller. Strict monotonicity under the product order preserves the strict preference. $\square$

B. Recursive Pareto Strata

Let $ND(A)$ denote the non-dominated points of set A . Define

$$Q_1(M) = ND(S_M) \quad (18)$$

and recursively

$$Q_k(M) = ND\left(S_M \setminus \bigcup_{j < k} Q_j(M)\right). \quad (19)$$

The stratum number is a *candidate-set-relative dominance depth*, not an absolute quality grade. Adding or removing systems can change lower strata even when no stemmer output changes.

Figure 1 shows the complete-resource English matrix. Three implementations have one aggregate error vector, but this does not prove that their individual pair errors are identical.

C. Conflation Regimes and Pair-Cost Support

Within one stratum, the point with minimum FN is the *recall edge*; the point with minimum FP is the *specificity edge*. A single-point front is *joint-dominant*. Other points are compromises.

For a linear pair-error policy,

$$C_\rho(s) = \rho FN(s) + FP(s), \quad \rho \geq 0, \quad (20)$$

ρ is the cost assigned to one missed reference-positive pair relative to one false reference-negative pair. The exact interval $I_M(s)$ contains the ρ values for which s minimizes C_ρ within its own stratum. A non-edge point with a non-empty interval is a *supported compromise*; a non-dominated point without such an interval is *unsupported efficient* and requires a nonlinear, threshold, or lexicographic policy.

The earlier rate-weighted form $\lambda UI + OI$ is equivalent within one matrix under $\rho = \lambda N/P$, where $P = TP + FN$ and $N = TN + FP$. Equation (20) avoids confusing a class-rate weight with the cost of one erroneous pair. These pair costs are still type-level policy parameters, not empirical query economics.

TABLE I
MULTILINGUAL PARETO-STRATUM CENSUS

Stratum	Impl. rows	Error vectors	Matrices containing stratum
Q_1	40	40	40
Q_2	111	97	38
Q_3	29	27	23
Q_4	10	10	9

V. EXPERIMENTAL DESIGN

The checked snapshot at commit 0c3b13f contains 314 scenarios over 20 default language-resource configurations, all UniMorph-derived in this release [16], [17]. Two modes are retained: `ALL_WORDS`, the primary complete-resource view, and nested `LOWERCASE_GROUPS_ONLY`, which removes a complete family when any member contains uppercase or titlecase code points. The latter is a sensitivity view, not an independent sample.

Each mode contains 95 deterministic implementation rows. The complete-resource view has 89 distinct aggregate error vectors: 20 Radixor vectors and 69 comparator vectors represented by 75 comparator implementation rows across 19 comparative languages. The nested view contains 85 vectors, including 65 distinct comparator vectors represented by the same 75 implementation rows. Hebrew has no configured comparator.

Implementation categories are explicit descriptive groupings and are not used in the Pareto computation. In particular, Snowball language stemmers are separated from English Porter-family implementations; minimal, light, and narrow filters remain distinct because they intentionally implement different normalization scopes.

All results are deterministic measurements of the checked matrices. No sampling p -values are reported because languages, resources, and implementations are not random draws from a defined population. The analysis asks:

- RQ1 What Pareto-dominance strata occur in the multilingual matrices?
- RQ2 Which conflation regimes and implementation-category profiles occupy the lower strata?
- RQ3 How sensitive are strata to the nested mode and to removal of one comparator error vector?
- RQ4 How does all-candidate activation move multi-output systems in the UI–OI plane?

VI. RESULTS

A. Four Pareto-Dominance Strata

Across both modes, 190 implementation rows yield 174 aggregate error vectors and four recursive strata (Table I, Fig. 2). Sixteen rows are count-equivalent to another row in the same matrix; no claim is made that their pairwise error assignments are identical.

B. Radixor Moves the Within-Reference Frontier

In the primary `ALL_WORDS` view, Radixor is the sole Q_1 vector in all 19 comparative languages and has both fewer FN pairs and fewer FP pairs than all 75 comparator implementation rows representing 69 distinct comparator error vectors. The

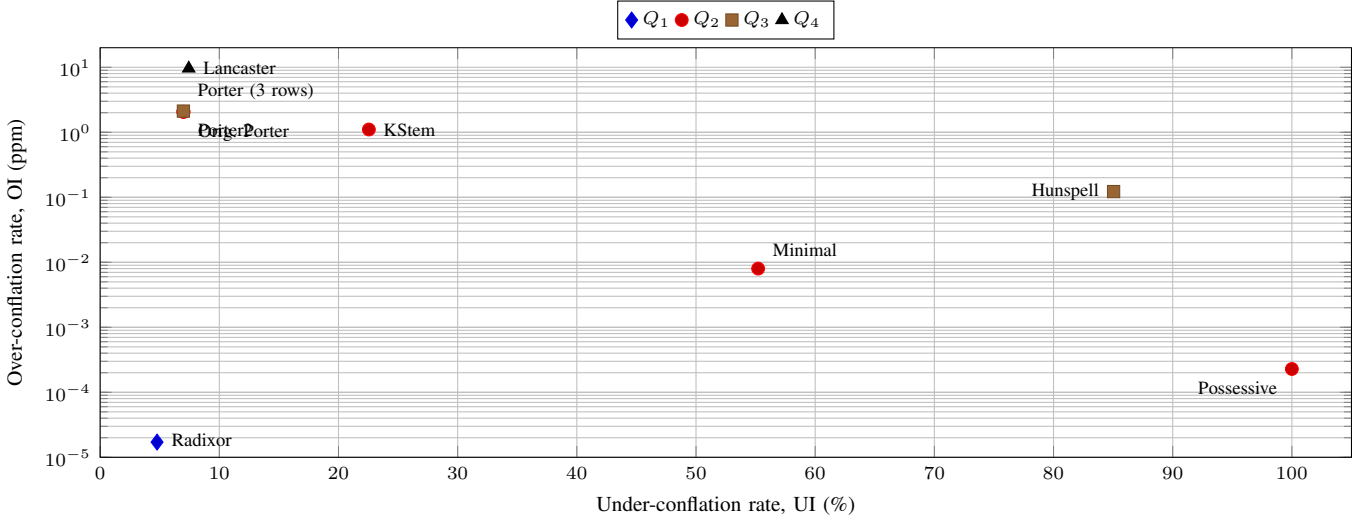


Fig. 1. Recursive Pareto strata for English ALL_WORDS. OI is shown in parts per million of reference-negative pairs. Marker shape denotes stratum; coincident rows share one aggregate (FN, FP) error vector.

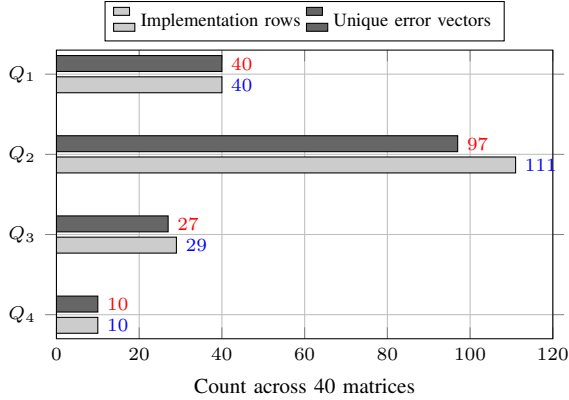


Fig. 2. Pareto-stratum census over 40 matrices.

nested mode also reproduces the same component-wise ordering for its 75 comparator rows and 65 comparator vectors.

Strict dominance does not imply equal robustness on both axes. Table II reports, for each comparative complete-resource matrix, the gap to the nearest comparator on each axis and then summarizes those 19 matrix-level minima. The smallest UI advantage is 2.197 percentage points and 1,237 FN pairs. The smallest FP advantage is only three pairs, corresponding to 0.000211 ppm; the median nearest-comparator FP gap is 4,263 pairs. Thus the UI dominance is uniformly substantial in rate terms, whereas the strict OI ordering is numerically thin in a few matrices and should not be detached from reference annotation uncertainty.

The conclusion remains within-reference. Radixor was induced from the lexical-resource family that defines the evaluated relation, whereas most comparators were not. The result demonstrates adaptation to the declared inflectional policy, not universal dominance on independent derivational, contextual, or task-specific references.

TABLE II
COMPLETE-RESOURCE RADIXOR DOMINANCE MARGINS OVER 19
COMPARATIVE MATRICES

Nearest-comparator gap	Minimum	Median
FN pairs	1,237	85,841
UI percentage points	2.197	21.992
FP pairs	3	4,263
OI parts per million	0.000211	0.863

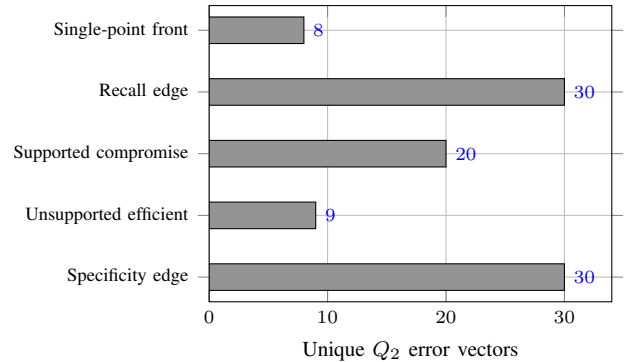


Fig. 3. Regimes among the 97 Q_2 aggregate error vectors.

C. The Second Stratum Contains Distinct Regimes

The 97 Q_2 error vectors comprise 30 recall edges, 30 specificity edges, 20 supported compromises, nine unsupported-efficient points, and eight single-point fronts (Fig. 3). Thus Q_2 is not one homogeneous runner-up class.

German illustrates the pair-cost interpretation (Fig. 4, Table III). After removing the dominating Q_1 point, CISTEM is the recall edge, German Minimal the specificity edge, and direct Snowball a supported compromise. If one missed reference-positive pair is assigned less than 0.0659 times the cost of one false reference-negative pair, Minimal is preferred within Q_2 ; Snowball is supported between 0.0659 and 0.700; CISTEM is supported above 0.700.

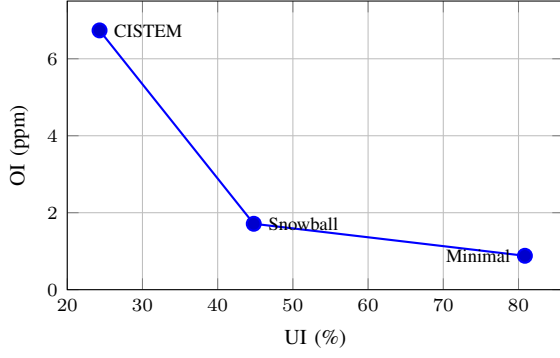


Fig. 4. German Q_2 operating regimes. The connected points form the linearly supported lower frontier within the stratum.

TABLE III
GERMAN Q_2 SUPPORT FOR $C_p = \rho FN + FP$

System	Regime	Conditional ρ interval
CISTEM	Recall edge	$[0.700, \infty)$
Snowball direct	Supported compromise	$[0.0659, 0.700]$
German Minimal	Specificity edge	$[0, 0.0659]$

D. Implementation Categories Have Different Profiles

Figure 5 reports complete-resource error vectors only. All 20 Radixor vectors are Q_1 . All seven minimal-filter vectors are Q_2 , six at the specificity edge. Snowball language stemmers contribute 28 vectors: 20 in Q_2 , seven in Q_3 , and one in Q_4 ; English Porter-family implementations contribute two Q_2 vectors and one Q_3 vector. Light filters span Q_2 – Q_4 , Hunspell places six of eight vectors in Q_2 , and Morfologik places two of three in Q_2 .

These profiles are consistent with known scope differences: minimal filters are predominantly conservative, while many Snowball language stemmers occupy recall edges or supported compromises. A deeper stratum does not mean that a narrow filter is badly implemented; it means that its measured relation is dominated under the broader reference policy used here.

E. Strata Are Relative but Measurably Stable

Pareto depth depends on which alternatives are present. Removing Radixor necessarily moves every remaining point upward by one stratum because Radixor alone forms Q_1 and dominates all comparators. A less drastic leave-one-comparator analysis deletes each non-Radixor error vector in turn and recomputes the strata. Across 592 retained-vector assignments, 551 (93.1%) remain unchanged, 41 improve by one stratum, and none changes by more than one. This quantifies candidate-set sensitivity rather than hiding it.

The nested mode gives a separate sensitivity view. Of 95 implementation-language units, 88 retain the same stratum. Table IV shows that all changes occur in the lower strata; Q_1 and Q_2 are unchanged for every paired unit.

Scalar rankings remain metric-dependent among non-dominated alternatives. Relative to balanced accuracy, the best comparator set is disjoint in 28 of 38 matrices for precision, 15 for $F_{0.5}$, and 12 for raw pair agreement, F_1 , and MCC;

TABLE IV
STRATUM TRANSITIONS FROM ALL_WORDS TO NESTED LOWERCASE-ONLY MODE

	Q_1	Q_2	Q_3	Q_4
Q_1	20	0	0	0
Q_2	0	53	0	0
Q_3	0	5	11	1
Q_4	0	0	1	4

recall never selects a disjoint set. F_1 and Jaccard are monotonic transforms and are not counted as independent evidence. These disagreements are projections of the same measured frontier, not contradictory experiments.

F. Multi-output Activation Follows a Constrained Trajectory

Proposition 1 limits all-candidate movement to two useful classes in this snapshot: lower UI with unchanged OI, or lower UI with higher OI. Across 31 complete-resource language-system units, 14 are Pareto improvements and 17 are trade-offs (Fig. 6). All 14 improvements are Radixor units; six additional Radixor units, all eight Hunspell units, and all three Morfologik units are trade-offs.

Magnitude matters in addition to class count. Median UI reductions are 2.219 percentage points for Radixor, 3.503 for Hunspell, and 6.624 for Morfologik. Median OI increases are 0 ppm for Radixor, 0.531 ppm for Hunspell, and 0.310 ppm for Morfologik. Radixor’s upper-quartile OI increase is only 0.000306 ppm, but its six nonzero cases remain genuine trade-offs. Candidate sets have different semantics across the systems, so these trajectories describe measured relation changes rather than one universal multi-output operation.

VII. DISCUSSION

A. What the Classes Mean

The taxonomy separates three questions:

- 1) *error-vector equivalence*: do implementations have the same aggregate FN and FP counts in this matrix?
- 2) *Pareto stratum*: how deep is the vector in the candidate-set-relative dominance order?
- 3) *conflation regime*: where does the vector lie on its stratum and for which pair-cost ratios is it linearly supported?

The class does not establish identical outputs, identical error pairs, or universal linguistic quality. It is a reproducible description of measured agreement with one declared reference policy.

A practical workflow follows. First eliminate dominated vectors. If several systems remain on the best available stratum, inspect recall and specificity edges, supported compromises, or application-specific thresholds. Only then apply a scalar score or downstream validation. This ordering prevents a preference function from obscuring unconditional inferiority.

B. NLP Interpretation of the Radixor Result

Within the evaluated resources, Radixor moves the measured frontier rather than offering another UI–OI compromise. This is consistent with a lexicon-trained system being able to compile

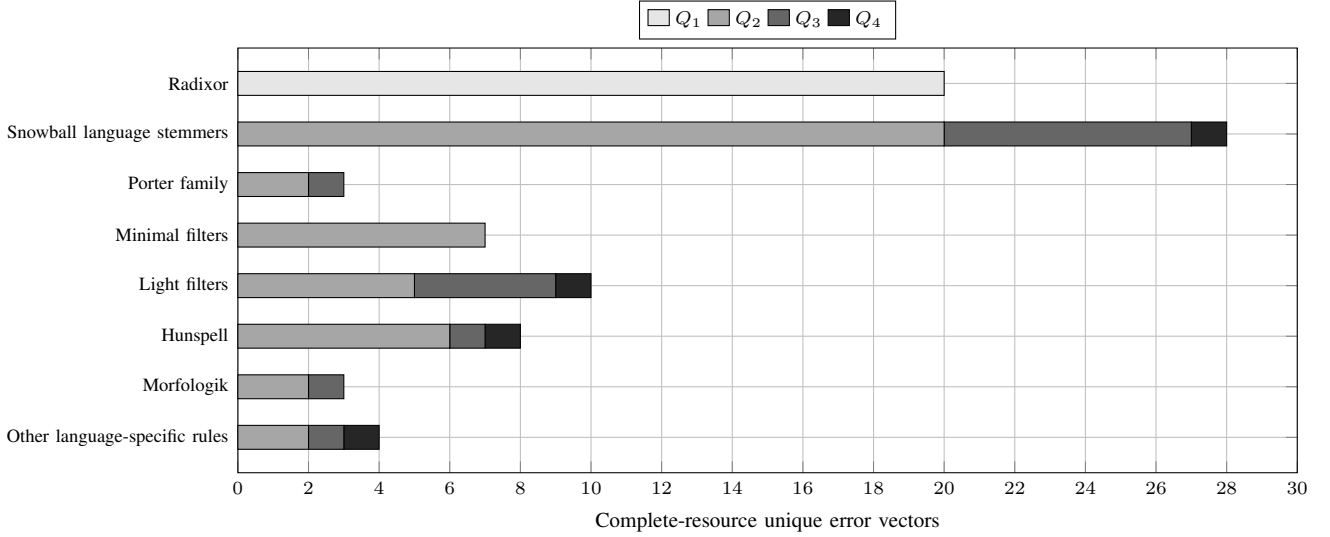


Fig. 5. Complete-resource Pareto-stratum profiles for selected explicit implementation categories. Categories are descriptive and do not determine the strata.

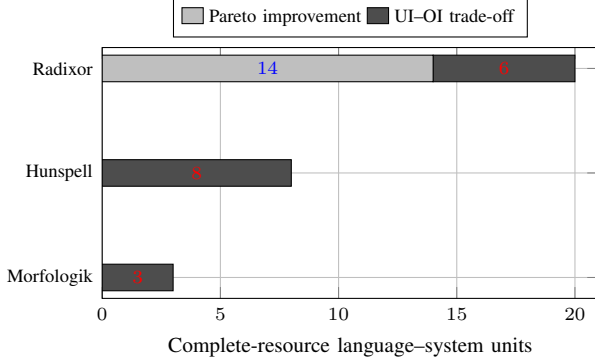


Fig. 6. Movement from deterministic primary output to all-active candidates.

the same kind of family evidence used by the reference policy. It is also why the result must not be generalized beyond that setting without independent references.

The comparators have intentionally heterogeneous scopes. Minimal, light, plural, and possessive filters are designed to perform less normalization; morphological analyzers may return several lemmas; rule stemmers often produce artificial stems rather than reference canonical strings. The Pareto classification measures the relation each tested implementation actually induces. It does not declare one algorithmic philosophy intrinsically superior for every search task.

Alternative German GS1/GS2 policies, external domain lexicons, or downstream retrieval collections would test whether stratum and regime assignments persist when the reference policy changes. Such validation is a distinct experiment rather than a missing algebraic step in the present framework.

C. Limitations

The taxonomy uses only UI and OI. Memory, latency, root preservation, downstream retrieval, robustness, or fairness across family sizes would produce a higher-dimensional Pareto problem. Pairwise weighting grows quadratically with family

size, and singleton families contribute no positive pairs while enlarging the negative population. The aggregate snapshot lacks cover-native family-size and membership histograms required for a family-macro sensitivity analysis.

Reference negatives follow a closed-world assumption, and UniMorph-derived families predominantly encode inflectional evidence. Unrecorded derivational or semantic relations can therefore appear as false conflation. Stratum depth depends on the evaluated candidate set, and strict OI dominance can rest on a small number of pair differences in some matrices. The two modes are nested, not independent. Implementation categories are explicit but descriptive, and cross-language counts are not population estimates.

VIII. REPRODUCIBILITY

All values refer to public commit 0c3b13f. The package contains the frozen 314-row count projection, regenerated relation metrics, 174 aggregate error vectors, Pareto assignments, pair-cost intervals, explicit category mappings, dominance margins, deletion sensitivity, nested-mode transitions, candidate trajectories, figures, and verification scripts. The verifier recomputes UI and OI from integer counts, independently reconstructs non-dominated layers, checks the 75 complete-resource and 75 nested comparator dominance relations, validates German pair-cost intervals, confirms all 592 deletion perturbations and 95 mode transitions, and verifies candidate-expansion monotonicity. Rounded documentation tables are not analytical inputs.

IX. CONCLUSION

Classifying stemmers only by one score or by aggressiveness conflates qualitatively different situations. Aggregate UI-OI vectors form a candidate-set-relative Pareto structure: dominance identifies policy-independent inferiority, while edge position and pair-cost support distinguish recall-oriented, specificity-oriented, supported, and nonlinear-policy alternatives.

Across 20 predominantly inflectional reference resources and two nested modes, 174 aggregate error vectors form four strata. In the primary complete-resource view, Radixor alone occupies Q_1 in every comparative language and dominates 75 comparator implementation rows on both axes, although the nearest OI margin is thin in a few matrices. Lower strata contain interpretable regimes and are mostly stable under nested-mode and single-comparator perturbations. Multi-output expansion is constrained to either an under-conflation improvement with unchanged OI or an explicit UI–OI trade-off. The resulting classes are auditable descriptions of measured reference-policy agreement, not universal labels detached from language, resource, implementation, or task.

REFERENCES

- [1] C. D. Paice, “An evaluation method for stemming algorithms,” in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994, pp. 42–50.
- [2] —, “Method for evaluation of stemming algorithms based on error counting,” *Journal of the American Society for Information Science*, vol. 47, no. 8, pp. 632–649, 1996.
- [3] W. B. Frakes and C. J. Fox, “Strength and similarity of affix removal stemming algorithms,” *ACM SIGIR Forum*, vol. 37, no. 1, pp. 26–30, 2003.
- [4] X. Liu, T. Sun, J. He, J. Wu, L. Wu, X. Zhang, H. Jiang, Z. Cao, X. Huang, and X. Qiu, “Towards efficient NLP: A standard evaluation and a strong baseline,” in *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2022, pp. 3288–3303.
- [5] B. Abbasi, J. Calder, and A. M. Oberman, “Anomaly detection and classification for streaming data using PDEs,” *SIAM Journal on Applied Mathematics*, vol. 78, no. 2, pp. 921–941, 2018.
- [6] J. B. Lovins, “Development of a stemming algorithm,” *Mechanical Translation and Computational Linguistics*, vol. 11, pp. 22–31, 1968.
- [7] —, “Error evaluation for stemming algorithms as clustering algorithms,” *Journal of the American Society for Information Science*, vol. 22, no. 1, pp. 28–40, 1971.
- [8] M. Lennon, D. S. Peirce, B. D. Tarry, and P. Willett, “An evaluation of some conflation algorithms for information retrieval,” *Journal of Information Science*, vol. 3, no. 4, pp. 177–183, 1981.
- [9] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.
- [10] L. Galambos, “Lemmatizer for document information retrieval systems in JAVA,” in *SOFSEM 2001: Theory and Practice of Informatics*, ser. Lecture Notes in Computer Science, vol. 2234. Springer, 2001, pp. 243–252.
- [11] T. Brychcín and M. Konopík, “HPS: High precision stemmer,” *Information Processing & Management*, vol. 51, no. 1, pp. 68–91, 2015.
- [12] D. A. Hull, “Stemming algorithms: A case study for detailed evaluation,” *Journal of the American Society for Information Science*, vol. 47, no. 1, pp. 70–84, 1996.
- [13] L. Weissweiler and A. Fraser, “Developing a stemmer for german based on a comparative analysis of publicly available stemmers,” in *Language Technologies for the Challenges of the Digital Age*, ser. Lecture Notes in Computer Science. Springer, 2018, vol. 10713, pp. 81–94.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [15] N. Bansal, A. Blum, and S. Chawla, “Correlation clustering,” *Machine Learning*, vol. 56, no. 1–3, pp. 89–113, 2004.
- [16] K. Batsuren, O. Goldman, S. Khalifa, N. Habash, W. Kieraś, G. Bella, B. Leonard, G. Nicolai, K. Gorman *et al.*, “UniMorph 4.0: Universal morphology,” in *Proceedings of the 13th Conference on Language Resources and Evaluation*. European Language Resources Association, 2022, pp. 840–855.
- [17] L. Galambos, “Radixor multilingual stemming-quality snapshot,” GitHub repository, commit 0c3b13f485a9ad0b460c0931e4497ea95bed66a1, 2026, checked-in 314-scenario quality matrix and methodology; accessed Aug. 26, 2026. [Online]. Available: <https://github.com/leogalambos/Radixor>