Journal of Integrated Knowledge
Vol. 2 No. 7 (2026)

A Shared Endpoint Is Stricter Than a Shared Interval: Temporal Candidate Support in YAGO–WIKI

Owen Carter[1] · Liam Johnston[2] · Evan Anderson[3]

[1] Poland · [2] Hungary · [3] Croatia

Published: July 08, 2026 | Article 858

Abstract

An interval can overlap another without sharing either endpoint. Does that distinction materially change temporal candidate support? Across all 44,172 released YAGO–WIKI reference queries and 49,222 target entities, replacing shared endpoint labels with closed-span overlap retained 36 additional known counterparts, a gain of 0.081500 percentage points. It also added 87,737,943 candidate opportunities. At least one gold endpoint lacked usable finite evidence in 2,748 pairs; 228 further pairs remained unsupported despite evidence on both sides. Independent geometric and inverted-index calculations agreed exactly. These collection-specific results separate modest positive containment gains from substantial candidate expansion, without assigning negative labels to unannotated pairs. Keywords: temporal entity alignment; interval semantics; candidate construction; positive support; benchmark audit

A Shared Endpoint Is Stricter Than a Shared Interval: Temporal Candidate Support in YAGO-WIKI

Abstract

An interval can overlap another without sharing either endpoint. Does that distinction materially change temporal candidate support? Across all 44,172 released YAGO-WIKI reference queries and 49,222 target entities, replacing shared endpoint labels with closed-span overlap retained 36 additional known counterparts, a gain of 0.081500 percentage points. It also added 87,737,943 candidate opportunities. At least one gold endpoint lacked usable finite evidence in 2,748 pairs; 228 further pairs remained unsupported despite evidence on both sides. Independent geometric and inverted-index calculations agreed exactly. These collection-specific results separate modest positive containment gains from substantial candidate expansion, without assigning negative labels to unannotated pairs.

Keywords: temporal entity alignment; interval semantics; candidate construction; positive support; benchmark audit

1. The representation behind temporal exclusion

Candidate construction decides which cross-graph comparisons a later entity matcher can consider. Temporal evidence can be especially attractive because a shared time label appears less language dependent than a name. Yet an absence of shared labels has several possible meanings. The two entities may have incompatible evidence, one graph may omit relevant intervals, or a representation may have discarded interior overlap. This study asks how much that last distinction changes known-positive support and candidate workload in a released temporal alignment benchmark. The comparison holds the graph facts, entity identifiers, reference pairs and target catalogue fixed, changing only the logical relation used to retain a candidate.

Indexing research treats the number of comparisons and preservation of potential matches as linked design concerns (Christen, 2012). Surveys distinguish blocking and filtering from the subsequent matching decision, making that separation useful here (Papadakis et al., 2021). A candidate set is an opportunity to compare entities; membership alone supplies neither a match probability nor a ranked prediction. Comparative blocking evaluations also illustrate why retaining more potential matches must be considered alongside comparison volume (Papadakis et al., 2016). Accordingly, the present outcomes are known-positive containment and candidate counts. They do not reward a larger candidate set as if every additional comparison improved alignment quality.

Recent temporal alignment designs make the meaning of an early candidate exclusion consequential. The accessible publisher abstract and introduction of ATEA describe complementary candidate channels and consistency-aware evidence fusion, including further evidence acquisition when internal candidate evidence disagrees (Wu et al., 2026). That motivates examining whether a temporal disagreement originates in source support or in a chosen representation. The final T3A-LLM abstract places temporal and relational encoding before candidate restriction and later semantic scoring (Wang et al., 2026). Its relevance is the dependency of a later stage on the opportunities admitted earlier. Neither method is implemented or evaluated in this experiment, and neither is asserted to use the two rules compared below. The final six-author T3A publication is distinguished from its earlier four-author workshop text; detailed workshop procedures are not attributed to the final chapter.

The temporal distinction itself precedes contemporary graph models. Allen (1983) develops interval-based temporal reasoning, while Gadia (1988) embeds temporal information in a relational

model. These works motivate declaring the temporal objects and their operations rather than treating every encoded date as an interchangeable token. We use only closed overlap, without implementing a complete interval algebra or temporal database language. Jensen and Snodgrass (1999) discuss temporal data management more broadly, and Snodgrass (1999) provides a database application treatment. Their relevance is semantic discipline: a missing boundary, a point and a finite interval need explicit interpretations before any computation can support an empirical claim.

2. Released data and a shared finite clock

The analysis uses the complete YAGO-WIKI tables distributed with LightTEA at commit a441379b000dcfd90ba146060dc051ffc5e1c78b. LightTEA studies temporal entity alignment through label propagation across multiple views (Cai et al., 2023); its released tables provide the fixed empirical collection here. The archive was rehashed, checked for ZIP corruption and inventoried across all 17 benchmark tables. The present effect analysis uses only YAGO-WIKI because its five-column facts explicitly supply beginning and ending time identifiers. The archive's ICEWS tables were inspected for schema and integrity but are outside this interval question. This boundary was fixed before candidate support was computed.

The source side declares 49,629 entities and the target side 49,222. Every one of the 44,172 released reference pairs is a query, and the candidate universe contains every declared target entity. The 5,000 supplied training pairs serve only to check identifier membership, uniqueness and separation from reference pairs. There is no training, validation optimization or holdout selection: both candidate definitions were frozen before reference containment was evaluated. Facts from both graphs are available transductively. One-to-one known pair endpoints and zero overlap between the two supplied pair files were verified. These checks describe the release rather than establishing that its correspondences exhaust real identities, a distinction consistent with concerns about benchmark construction in alignment evaluation (Sun et al., 2020).

YAGO3 and Wikidata describe different knowledge-base construction settings (Mahdisoltani et al., 2015; Vrandečić and Krötzsch, 2014). Their publications establish source context, but the pinned benchmark tables determine the actual observations. No claim is made that those tables reproduce an entire historical upstream dump. The repository's software license also does not replace upstream data terms. The provenance record preserves repository attribution, YAGO's stated attribution requirement and Wikidata's structured-data licensing context. Raw upstream text is not republished with the manuscript; the derived query table contains numeric identifiers and support summaries.

Both benchmark graphs use one dictionary with 245 timestamp codes. Upstream TEA-GNN explicitly describes the first timestamp as unknown (Xu et al., 2021). Therefore code zero is treated as unknown even though the local dictionary prints an infinity-like string. A fact is eligible only when both endpoints are nonzero, map to finite integer labels, and have beginning no greater than ending. Equal endpoints remain valid point intervals. The 244 finite labels are ordered by their numeric values; no distance between arbitrary timestamp identifiers is interpreted as elapsed time. Labels such as 19,433 are retained literally. They are shared source labels, without independent historical validation or an assertion that such values represent credible observed dates.

Table 1 records the complete fact flow. A fact attaches its eligible interval to each actual head and tail entity. The resulting object is an incident-fact temporal footprint. It is not an entity lifespan, an activity estimate or a validated event chronology. Repeated facts and self-loops cannot enlarge a set merely through repetition. Unknown boundaries and inverted finite intervals are reported separately and excluded equally from both candidate rules. They are neither repaired nor converted into unbounded spans. Time

granularity can matter in temporal systems (Bettini et al., 2000), but this comparison introduces no tolerance window, finer calendar or semantic smoothing.

Table 1. Complete released YAGO-WIKI fact eligibility. Unknown-boundary rows and inverted finite rows are excluded from both policies; point rows are a subset of eligible rows.

Source state	Graph 1	Graph 2
All released fact rows	221,050	317,814
Finite, ordered intervals retained	146,094	296,763
Unknown beginning	0	178
Unknown ending	74,264	20,841
Both boundaries unknown	0	0
Finite inverted interval	692	32
Eligible point intervals	776	92,963

3. Two fixed logical candidate policies

For an entity e, let E(e) contain the beginning and ending labels of all eligible incident facts. Let I(e) be the union of the closed spans of those same facts. The endpoint policy retains target t for query q when E(q) and E(t) intersect. The interval policy retains t when some incident interval on q overlaps some incident interval on t, using max(beginning) ≤ min(ending). Endpoint labels are pooled without separately matching beginning to beginning or ending to ending. An empty footprint gives an empty candidate set under either definition and remains an explicit observed support state.

Proposition P1 states that the endpoint candidate set is contained in the interval candidate set. A shared endpoint lies in both corresponding closed spans, so every endpoint candidate must also be an interval candidate. This inclusion was checked separately for every query. Conversely, overlapping intervals need not share an endpoint. For example, the illustrative spans [2, 6] and [4, 8] overlap even though their endpoint sets are disjoint. This constructed example explains the operation and is not an empirical benchmark observation. Dignös et al. (2012) study temporal alignment in relational query processing, which is distinct from identifying equivalent graph entities. That distinction helps separate the temporal overlap operation used here from the identity claim a subsequent matcher would need to establish.

The computation uses the 244 finite dictionary values as an exact support vocabulary. For each entity it records the vocabulary values lying within its eligible spans. This preserves overlap exactly: when two closed intervals overlap, the larger beginning is itself a vocabulary endpoint inside their intersection. No dense enumeration of all integers up to 19,433 is required, and interior vocabulary membership is a computational representation of a span, not a claim of an additional observed event. Target postings and bitsets then recover the complete union of candidates. This follows the general dictionary-and-postings organization of inverted retrieval (Manning et al., 2008), without introducing a retrieval score or candidate budget.

The two primary effects are the mean paired increase in known-positive containment and the mean paired increase in candidate count across all queries. Each containment observation asks whether that query's released counterpart remains available. All other target labels stay unresolved. Work on positive and unlabeled learning explicitly distinguishes unlabeled examples from observed negatives (Elkan and Noto, 2008; Bekker and Davis, 2020). We adopt that distinction without fitting a classifier or assuming that

benchmark positives were selected at random. Precision, specificity, false-positive counts and general alignment accuracy are consequently not identified by this design. The larger candidate workload is reported directly rather than renamed as classification error.

The reference collection is evaluated in full, so numerical summaries describe this collection exactly. To expose dependence on any one query, we calculate every possible delete-one-query mean. If the paired values sum to S, deleting value d gives (S - d)/(N - 1); the smallest and largest such means define a deterministic sensitivity range. This range is not a confidence interval, does not quantify uncertainty about historical source truth and does not create independent graph replications. Medians and 95th percentiles use the inverse empirical cumulative distribution with no interpolation. A prespecified secondary summary restricts to pairs with eligible evidence on both gold endpoints, while retaining the full collection as the primary denominator.

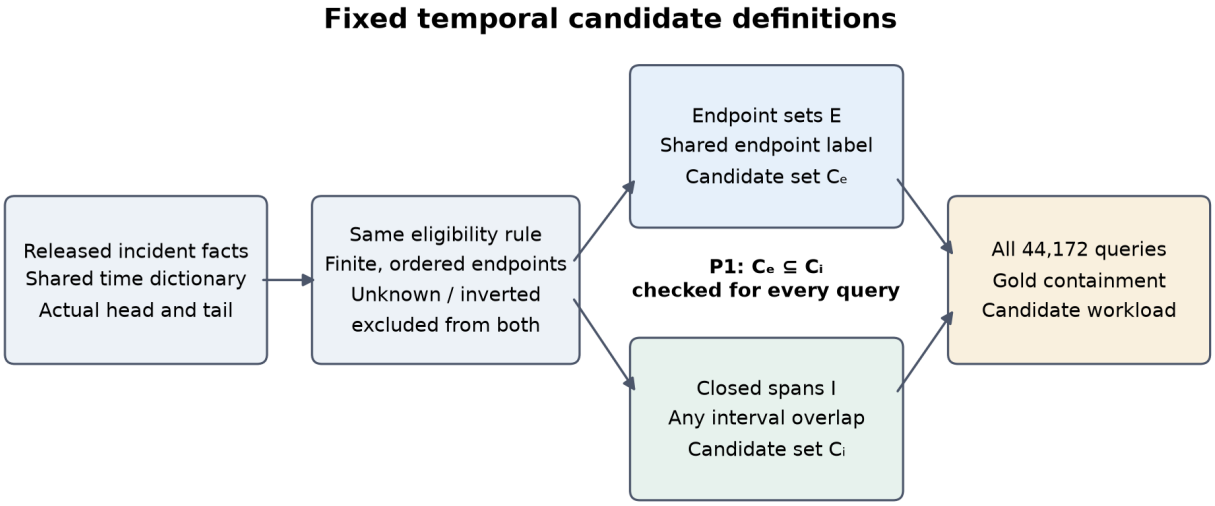

Figure 1. The original measurement model. Both branches receive exactly the same eligible incident facts. Arrows indicate computational operations, and candidate inclusion P1 is checked for each query. Labels on other candidates remain unknown.

4. Support gains and candidate expansion

Shared endpoints retained 41,160 known counterparts, whereas interval overlap retained 41,196 (Table 2). The difference was 36 queries, or 0.081500 percentage points across 44,172 queries. Every recovered pair arose through the broader fixed overlap condition; no query lost its known counterpart. The exact delete-one range for that increase was [0.079238, 0.081501] percentage points. Its narrowness reflects deletion of only one row from a large fixed collection. It offers no evidence that the same effect would recur in another benchmark, source version or temporal encoding.

Table 2. Full-reference positive containment and workload. Candidate counts use the complete 49,222-entity target catalogue. Percentages concern released known counterparts, not correctness of every candidate.

Policy	Known counterpart retained / N	Retention (%)	Known counterpart misses	Mean candidates/query	Median	95th percentile
Shared endpoint	41,160 / 44,172	93.181201	3,012	22,978.969324	27,522	36,876
Closed-span overlap	41,196 / 44,172	93.262700	2,976	24,965.248936	30,658	38,750

Table 3. Paired effects on all 44,172 queries. The final column gives the exact range of means after deleting any single query, in the same unit as the effect. These are deterministic sensitivity envelopes, not confidence intervals.

Outcome	Paired sum	Mean paired effect	Delete-one mean range
Known-positive containment increase	36	0.081500 percentage points	[0.079238, 0.081501]
Candidate count increase	87,737,943	1,986.279612 target entities/query	[1,985.655521, 1,986.324579]

The candidate increase was substantially larger in its own operational unit: 87,737,943 additional query-target opportunities, averaging 1,986.279612 per query. Mean workload rose from 22,978.969324 to 24,965.248936 target entities, respectively 46.684347% and 50.719696% of the complete target catalogue. The median increase was 1,573 and the 95th percentile 5,187; the largest increase was 29,553. Figure 2 retains the zero-footprint queries in both cumulative distributions. It shows that interval support is already broad before any name, relation-specific or learned similarity evidence is applied. These counts describe admissible comparisons rather than elapsed runtime or measured memory consumption.

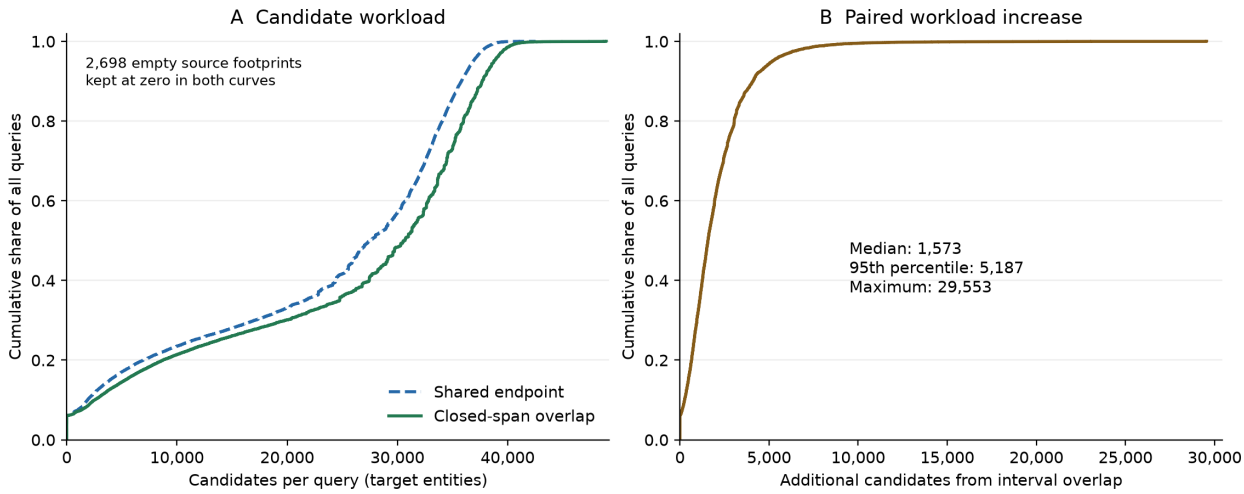

Figure 2. Empirical cumulative distributions over all 44,172 queries. Panel A compares full candidate counts; panel B shows their paired increase. Every zero value remains included. Curves are exact collection summaries and have no sampling confidence bands.

Missing evidence explains most remaining misses. There were 2,698 queries without an eligible source footprint and 254 whose known target lacked one; both conditions held for 204 pairs. Thus 2,748 pairs lacked finite support on at least one gold endpoint. Another 228 pairs had evidence on both sides but no overlapping eligible interval. Together these states account for all 2,976 interval misses. The common zero-candidate count, 2,698, matches empty source footprints; a missing gold-target footprint can still

coexist with many candidates among other targets. This separation prevents a populated candidate list from being mistaken for evidence that the known counterpart was available.

Among the 41,424 pairs with eligible evidence on both gold endpoints, shared endpoints retained 99.362688% and interval overlap 99.449594%. All 36 rescues remained, producing a secondary increase of 0.086906 percentage points. Mean candidate counts were 24,489.237085 and 26,601.655176, with mean increase 2,112.418091. The corresponding delete-one range for the workload increase was [2,111.755643, 2,112.469015]. Excluding unsupported pairs therefore raises conditional containment but does not remove candidate expansion or the 228 residual misses. This supportive subset describes a different denominator and cannot replace the primary result.

5. What a temporal candidate gate can justify

The observed gain demonstrates a concrete representational loss: endpoint-token agreement can exclude released counterparts whose incident spans overlap. Its magnitude in this collection was small in percentage points, while the added workload was large. Neither side of that comparison establishes a universally preferable gate. A system able to process many weak candidates may value the 36 recovered opportunities. Another system may require stronger joint evidence before accepting the additional workload. Dedoop describes a workflow separating candidate generation and matching in scalable deduplication (Kolb et al., 2012), but no Hadoop execution or performance comparison is made here. The result identifies a semantic and operational choice for a later system to evaluate.

Later matching evidence could behave quite differently from these temporal footprints. Pretrained language models have been used for entity matching (Li et al., 2020), but their results do not license assuming that every additional temporal candidate can be cheaply or correctly resolved. Similarly, ATEA's evidence fusion motivates preserving the distinction between absent and conflicting temporal support, while T3A's staged structure motivates checking what enters later scoring. The present calculation stops before either intervention. It neither estimates external retrieval benefits nor predicts the accuracy of semantic scoring on the 36 recovered pairs. Those questions require the corresponding implementations and an explicit evaluation protocol.

Several source properties can make broad temporal overlap unsurprising. An entity receives the union of intervals attached to all its incident facts, regardless of relation. Long spans or many incident relations can therefore connect it to a large fraction of the target catalogue. Knowledge graph surveys describe heterogeneous representations and construction practices (Hogan et al., 2022). Refinement research separately addresses missing knowledge and erroneous information (Paulheim, 2016). Our footprint policy does not distinguish those causes. It also does not reconcile conflicting source assertions, a task considered in data fusion research (Dong and Naumann, 2009). Keeping unusual labels and contradictory intervals visible is necessary to avoid implying that a deterministic overlap calculation has resolved such issues.

Unknown boundaries are especially consequential. Replacing an unknown ending with infinity would admit intervals that the frozen design excluded; deleting odd but finite labels would change a different aspect of the source. Either choice might be defensible for a new question, but this experiment supplies no validation for it. The 2,748 unsupported pairs and 228 finite-evidence misses show that interval expansion alone does not solve temporal incompleteness. They are substantive negative results, retained alongside the rescued pairs. The identity of unannotated targets also remains unresolved even when their temporal overlap is extensive. A candidate opportunity is weaker evidence than a correspondence annotation.

6. Reproducibility, scope and accountability

The execution record binds raw-file hashes, the frozen protocol, a derived row for every query, and exact aggregate sums. This follows provenance's distinction between source locations and the transformations producing a result (Buneman et al., 2001). Broader provenance uses include accountability, reproducibility and debugging (Herschel et al., 2017), all relevant to the separate source, protocol and result records here. The primary implementation uses integer support sets and NumPy summaries (Harris et al., 2020). A separate program reopens the raw files, merges intervals, recomputes gold support with geometric two-pointer comparisons, and rebuilds candidate counts through an independent inverted index using a reversed target namespace. All 574,236 saved integer fields and flags agreed; independently recomputed means, quantiles and deletion ranges had maximum absolute discrepancy zero.

Matplotlib generated the original model and empirical distribution graphics (Hunter, 2007), and both local PNGs were visually inspected. Scientific writing, analysis and visualization procedures from the Scientific Agent Skills library were used for the evidence and reporting workflow (Kassis et al., 2026). Software and procedural aids do not certify source truth or replace accountable scientific review. Providing executable artifacts rather than only a verbal method supports computational reproducibility as discussed by Peng (2011). The present check establishes agreement of two implementations on fixed bytes; it cannot validate every original fact, the representativeness of reference correspondences or the completeness of upstream knowledge.

The inference is confined to this release, these eligible intervals and these two untuned policies. No probability sample of graphs exists, and the query rows can share incident facts and time labels. We therefore make no population confidence or causal claim. The source reading record also preserves access limits: ATEA claims are confined to publisher-readable material, and final T3A claims to its accessible abstract and publication identity. The paid final chapter was not accessed. Human verification of citations, scientific interpretation, authorship, funding and conflicts remains pending. Within those limits, the experiment makes a narrow empirical finding reviewable: preserving closed-span overlap recovered 36 known opportunities, increased comparison volume sharply and left substantial support failures visible.

Data, code and declarations

The pinned public archive is available through the LightTEA repository at <https://github.com/lcai2/LightTEA/tree/a441379b000dcfd90ba146060dc051ffc5e1c78b>. Source hashes, schema checks, frozen protocol, derived query data, primary analysis and independent recalculation are supplied as companion artifacts. No personal linkage beyond the released numeric benchmark identifiers was added. Author approval and human source verification: **PENDING**. Submission readiness: **false**. No human author names, funding declarations or conflict statements are invented.

References

1. Buneman, Peter; Khanna, Sanjeev; Tan, Wang-Chiew. (2001). Why and Where: A Characterization of Data Provenance. Database Theory - ICDT 2001, 316-330. https://doi.org/10.1007/3-540-44503-X_20.
2. Vrandečić, Denny; Krötzsch, Markus. (2014). Wikidata: a free collaborative knowledgebase. Communications of the ACM. https://doi.org/10.1145/2629489
3. Paulheim, Heiko. (2016). Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web. https://doi.org/10.3233/sw-160218
4. Cai, Li; Mao, Xin; Xiao, Youshao; Wu, Changxu; Lan, Man. (2023). An Effective and Efficient Time-aware Entity Alignment Framework via Two-aspect Three-view Label Propagation. Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence. https://doi.org/10.24963/ijcai.2023/558

5. Wu, Wenjun; Xu, Kuangshan; Zhao, Runhao; Zhang, Shiqi; Zeng, Weixin; Tang, Jiuyang. (2026). Adaptive retrieval-augmented evidence-fusion for highly heterogeneous temporal entity alignment. Information Fusion. https://doi.org/10.1016/j.inffus.2026.104516.

6. Sun, Zequn; Zhang, Qingheng; Hu, Wei; Wang, Chengming; Chen, Muhao; Akrami, Farahnaz; Li, Chengkai. (2020). A benchmarking study of embedding-based entity alignment for knowledge graphs. Proceedings of the VLDB Endowment. https://doi.org/10.14778/3407790.3407828

7. Hunter, John D.. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering. https://doi.org/10.1109/mcse.2007.55

8. Hogan, Aidan; Blomqvist, Eva; Cochez, Michael; D'amato, Claudia; Melo, Gerard De; Gutierrez, Claudio; Kirrane, Sabrina; Gayo, José Emilio Labra; Navigli, Roberto; Neumaier, Sebastian; Ngomo, Axel-Cyrille Ngonga; Polleres, Axel; Rashid, Sabbir M.; Rula, Anisa; Schmelzeisen, Lukas; Sequeda, Juan; Staab, Steffen; Zimmermann, Antoine. (2022). Knowledge Graphs. ACM Computing Surveys. https://doi.org/10.1145/3447772

9. Snodgrass, Richard T.. (1999). Developing Time-Oriented Database Applications in SQL. Morgan Kaufmann. https://shop.elsevier.com/books/developing-time-oriented-database-applications-in-sql/snodgrass/978-0-08-050422-3

10. Kassis, Timothy; Agarwal, Vinayak; He, Yuhuan; Patel, Darshil; Brueckner, Aubrey M.. (2026). Scientific Agent Skills: A Library of Procedural Knowledge for Research Agents. arXiv preprint. https://arxiv.org/abs/2609.00065

11. Jensen, Christian S.; Snodgrass, Richard T.. (1999). Temporal data management. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/69.755613

12. Harris, Charles R.; Millman, K. Jarrod; van der Walt, Stéfan J.; Gommers, Ralf; Virtanen, Pauli; Cournapeau, David; Wieser, Eric; Taylor, Julian; Berg, Sebastian; Smith, Nathaniel J.; Kern, Robert; Picus, Matti; Hoyer, Stephan; van Kerkwijk, Marten H.; Brett, Matthew; Haldane, Allan; del Río, Jaime Fernández; Wiebe, Mark; Peterson, Pearu; Gérard-Marchant, Pierre; Sheppard, Kevin; Reddy, Tyler; Weckesser, Warren; Abbasi, Hameer; Gohlke, Christoph; Oliphant, Travis E.. (2020). Array programming with NumPy. Nature. https://doi.org/10.1038/s41586-020-2649-2

13. Li, Yuliang; Li, Jinfeng; Suhara, Yoshihiko; Doan, AnHai; Tan, Wang-Chiew. (2020). Deep entity matching with pre-trained language models. Proceedings of the VLDB Endowment. https://doi.org/10.14778/3421424.3421431

14. Kolb, Lars; Thor, Andreas; Rahm, Erhard. (2012). Dedoop: Efficient Deduplication with Hadoop. Proceedings of the VLDB Endowment. https://www.vldb.org/pvldb/vol5/p1878_larskolb_vldb2012.pdf

15. Christen, Peter. (2012). A Survey of Indexing Techniques for Scalable Record Linkage and Deduplication. IEEE Transactions on Knowledge and Data Engineering. https://doi.org/10.1109/tkde.2011.127

16. Bekker, Jessa; Davis, Jesse. (2020). Learning from positive and unlabeled data: a survey. Machine Learning. https://doi.org/10.1007/s10994-020-05877-5

17. Peng, Roger D.. (2011). Reproducible Research in Computational Science. Science. https://doi.org/10.1126/science.1213847

18. Bettini, Claudio; Jajodia, Sushil; Wang, X. Sean. (2000). Time Granularities in Databases, Data Mining, and Temporal Reasoning. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-04228-1

19. Mahdisoltani, Farzaneh; Biega, Joanna; Suchanek, Fabian M.. (2015). YAGO3: A Knowledge Base from Multilingual Wikipedias. Proceedings of the Seventh Biennial Conference on Innovative Data Systems Research (CIDR2015). https://www.cidrdb.org/cidr2015/Papers/CIDR15_Paper1.pdf

20. Gadia, Shashi K.. (1988). A homogeneous relational model and query languages for temporal databases. ACM Transactions on Database Systems. https://doi.org/10.1145/49346.50065

21. Papadakis, George; Svirsky, Jonathan; Gal, Avigdor; Palpanas, Themis. (2016). Comparative analysis of approximate blocking techniques for entity resolution. Proceedings of the VLDB Endowment. https://doi.org/10.14778/2947618.2947624

22. Herschel, Melanie; Diestelkämper, Ralf; Ben Lahmar, Houssem. (2017). A survey on provenance: What for? What form? What from?. The VLDB Journal. https://doi.org/10.1007/s00778-017-0486-1

23. Dong, Xin Luna; Naumann, Felix. (2009). Data fusion: resolving data conflicts for integration. Proceedings of the VLDB Endowment. https://doi.org/10.14778/1687553.1687620

24. Xu, Chengjin; Su, Fenglong; Lehmann, Jens. (2021). Time-aware Graph Neural Network for Entity Alignment between Temporal Knowledge Graphs. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. https://doi.org/10.18653/v1/2021.emnlp-main.709

25. Papadakis, George; Skoutas, Dimitrios; Thanos, Emmanouil; Palpanas, Themis. (2021). Blocking and Filtering Techniques for Entity Resolution: A Survey. ACM Computing Surveys. https://doi.org/10.1145/3377455

26. Manning, Christopher D.; Raghavan, Prabhakar; Schütze, Hinrich. (2008). Introduction to Information Retrieval. Cambridge University Press. https://doi.org/10.1017/cbo9780511809071.

27. Allen, James F.. (1983). Maintaining knowledge about temporal intervals. Communications of the ACM. https://doi.org/10.1145/182.358434

28. Elkan, Charles; Noto, Keith. (2008). Learning classifiers from only positive and unlabeled data. Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. https://doi.org/10.1145/1401890.1401920

29. Wang, Tao; Zhao, Runhao; Tang, Jiuyang; Wang, Xianghan; Liu, Xiaolu; He, Sijie. (2026). T3A-LLM: A Two-Stage Temporal Knowledge Graph Alignment Method Enhanced by LLM. Lecture Notes in Computer Science. https://doi.org/10.1007/978-3-032-15535-1_13.

30. Dignös, Anton; Böhlen, Michael H.; Gamper, Johann. (2012). Temporal alignment. Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data. https://doi.org/10.1145/2213836.2213886