

COVER-KBC: Evidence-Centric Adaptive Inference for Closed-Book Knowledge Base Construction

Quoc-Linh Vo

Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
vqlinh2433@apcs.fitus.edu.vn

Abstract

Large Language Models (LLMs) encode substantial factual knowledge, but reliably extracting complete knowledge-base entries remains challenging when relations differ in cardinality, semantics, and missingness. We present COVER-KBC, our system for the AKBC Shared Task 2026, which formulates closed-book knowledge extraction as relation-typed, evidence-centric inference. Each relation is compiled into an inference program that controls elicitation, verification, stopping, and final selection. Model observations are accumulated in a provenance-aware evidence state and scored using independent support, calibrated verification, contradiction, and disagreement. To distinguish candidate correctness from answer-space completeness, COVER-KBC estimates residual search need and uses a budget-aware controller to adaptively acquire further evidence when necessary. Relation-specific finalization handles heterogeneous output structures and difficult numeric or nullable cases. Using a single Mistral-Small-3.2-24B checkpoint without additional training or external retrieval, COVER-KBC achieves a Macro-F1 of 0.5878 on the hidden test set. Our implementation is available at <https://github.com/vquclinh/cover-kbc>.

1 Introduction

Pretrained language models encode substantial relational knowledge in their parameters, making them attractive sources for closed-book knowledge base construction (KBC) (Petroni et al., 2019). Yet factual knowledge that can be elicited from a model is not always stable across prompts (Cao et al., 2021), and constructing a knowledge-base entry is harder than recalling a single fact. A system must recover the appropriate object set, reject unsupported candidates, and determine whether valid objects may still be missing. This becomes especially challenging when relations range from nullable and numerical

attributes to small closed sets and large open-ended sets.

We argue that heterogeneous closed-book KBC is therefore not merely a generation problem, but an active evidence-acquisition problem. The objective is to recover an object set that is both precise and sufficiently complete under a bounded inference budget. Importantly, confidence that an individual candidate is correct does not imply that the answer set is complete (Razniewski et al., 2024); deciding whether to trust a candidate and whether to continue searching are distinct inference problems.

We propose COVER-KBC, a relation-typed, evidence-centric framework built around this distinction. First, each relation is compiled into an executable inference contract that governs elicitation, verification, stopping, and final selection. Second, model observations are accumulated as independence-aware atomic evidence, preserving provenance, support, contradiction, and verification signals rather than collapsing repeated generations into a flat vote. Third, COVER-KBC separately models candidate confidence and residual search need, allowing a coverage-aware controller to decide whether to acquire further evidence, verify unresolved candidates, or stop. Thus, both *what to ask next* and *when enough evidence has been collected* become relation-dependent decisions.

AKBC Shared Task 2026. This shared task (Kalo et al., 2026) evaluates closed-book object-set prediction. For each query $q = (s, r)$, where s is a subject entity and r is a relation, the system must predict an object set $\hat{O}_{s,r}$ using only the parametric knowledge of a language model. The benchmark covers six heterogeneous relations: land borders, city of death, venue capacity, award recipients, stock exchanges, and geographic area¹. Predictions may be empty, singleton, or multi-valued; external factual retrieval and additional model train-

¹<https://www.akbc.ws/2026/shared-task.html>

ing are prohibited, while multi-step inference is allowed under a 32B neural-parameter budget. Using a single Mistral-Small-3.2-24B checkpoint without additional training or external retrieval, COVER-KBC achieves 0.5878 Macro-F1 on the hidden test set.

2 Related Work

2.1 Closed-Book Knowledge Construction

Prior LM-KBC systems use multi-step or relation-specific prompting, relation-wise self-consistency, and relation-dependent decomposition (Alivanistos et al., 2022; Das et al., 2024; Albert-Roulhac and Zouaq, 2025; He and Razniewski, 2025). Multi-stage language-model programs similarly organize multiple model calls into structured inference procedures (Opsahl-Ong et al., 2024). COVER-KBC instead represents each relation as an executable contract governing elicitation, verification, stopping, and finalization.

2.2 Evidence, Verification, and Calibration

Self-consistency aggregates repeated generations (Wang et al., 2023), while Atomic Self-Consistency combines reliable subcomponents across samples (Thirukovalluru et al., 2024). Knowledge-fusion methods preserve source-specific evidence when estimating factual reliability (Dong et al., 2014). Chain-of-Verification separates verification from initial generation (Dhuliawala et al., 2024), and contextual calibration reduces systematic label bias (Zhao et al., 2021). COVER-KBC combines these ideas in an independence-aware evidence state with candidate-level blind verification.

2.3 Adaptive and Coverage-Guided Inference

Adaptive-Consistency stops sampling when additional generations are unlikely to change the decision (Aggarwal et al., 2023), while FLARE and Self-RAG condition further information acquisition on uncertainty or self-reflection (Jiang et al., 2023; Asai et al., 2024). Knowledge-base completeness work distinguishes correctness of observed facts from whether relevant facts remain missing (Razniewski et al., 2024). COVER-KBC operationalizes these ideas in a closed-book setting by separating candidate confidence from residual search need and using relation-specific evidence to determine further acquisition or stopping.

Relation	Type
countryLand	SMALLSET
BordersCountry	
companyTradesAt	
StockExchange	SMALLSET
personHas	
CityOfDeath	NULLSINGLE
hasArea	NUMERIC
hasCapacity	NUMERIC
awardWonBy	LARGEOPENSET

Table 1: The six relations are compiled into typed inference programs.

3 Methodology

We formulate closed-book KBC as relation-typed active evidence acquisition. Given the task setting above, COVER-KBC does not map a prompt response directly to a final object set. Instead, each model call is parsed as an observation and accumulated into an explicit inference state containing candidate objects, evidence provenance, verification signals, residual search need, and remaining budget. The system has four components: relation-typed inference contracts, independence-aware evidence aggregation, blind candidate verification, and budget-aware control with relation-specific finalization.

3.1 Relation-Typed Inference Programs

We define each relation through an executable inference contract

$$C_r = (\tau_r, \mathcal{V}_r, \mathcal{P}_r, \mathcal{G}_r, F_r),$$

where τ_r is the relation type, $\mathcal{V}_r$ is the set of eligible evidence-acquisition views, $\mathcal{P}_r$ defines parsing and normalization, $\mathcal{G}_r$ defines verification and stopping behavior, and F_r is the final selector.

This contract is our mechanism for relation conditioning: it specifies the semantic definition of the relation, positive cases, hard negatives, cardinality, mandatory views, optional views, verifier policy, stopping policy, and output selector. Relation conditioning therefore affects the program executed by the system, not only the wording of the prompt. Table 1 shows how the six task relations are mapped to COVER-KBC program types.

The four program types correspond to different inference problems. SMALLSET relations require high-precision set closure. NULLSINGLE relations require distinguishing true empty answers from failed recall. NUMERIC relations require parsing and clustering scalar values. LARGEOPENSET re-

lations require continued discovery while avoiding unbounded tail expansion.

3.2 Independence-Aware Evidence State

Each model call produces a parsed observation. COVER-KBC stores these observations in an evidence state

$$S_t = (O_t, R_t, B_t),$$

where O_t is the current set of candidate objects, R_t is the set of evidence records, and B_t stores budget, gate, and controller state. Each record attaches to one candidate object and carries the source view, prompt provenance, parser result, evidence type, and an independence group.

We do not count repeated samples from the same view as independent corroboration. Evidence from different acquisition mechanisms, such as direct recall, structural decomposition, contrastive questioning, missingness search, or reverse checking, can provide separate support. This separates repeated wording effects from genuinely distinct evidence.

For a candidate object o , let $m(o)$ be the set of acquisition groups that could express o under the relation contract, and let $g(o) \subseteq m(o)$ be the groups that support it. We define the independent support of candidate o as

$$F(o) = \frac{|g(o)|}{|m(o)|}.$$

Verifier agreement is represented separately because verifier calls are shown the candidate.

3.3 Blind Verification and Candidate Confidence

COVER-KBC combines the evidence attached to a candidate into a single score:

$$S(o) = \alpha F(o) + \beta L(o) + \gamma X(o) - \delta C(o) - \eta U(o),$$

where $F(o)$ is the independent acquisition support of Section 3.2, $L(o)$ is the calibrated verifier signal defined below, and $X(o)$ is optional heterogeneous recall from a second model family. Because our submitted system uses one Mistral checkpoint for every neural role, $X(o)$ is inactive throughout the results we report.

The two penalties capture evidence against a candidate rather than absence of evidence for it. $C(o)$ is the normalized strength of explicit contradiction, where the denominator covers the acquisition mechanisms together with the blind verifier; a view

that simply does not mention the candidate, an optional view that was never run, and an UNKNOWN verifier label are all excluded, since none of them is a signed judgement against o . $U(o)$ is the normalized disagreement among verifier label distributions across the prompt templates of a relation, so a candidate whose verifier judgement changes with wording is discounted. All five weights are constants of the released configuration; their values are listed in Appendix A.

Verification is deliberately blind. The verifier receives the subject, relation definition, hard-negative rules, and one candidate; it sees neither the generator prompt and rationale nor the support count or competing candidates, so its judgement cannot inherit the generator’s confidence. It predicts one of three labels: VALID, INVALID, or UNKNOWN. We calibrate the verifier logits against a content-free baseline:

$$\tilde{p}_j = \frac{\exp((z_j - b_j)/T)}{\sum_k \exp((z_k - b_k)/T)},$$

where z_j is the raw label logit, b_j is the content-free baseline logit for the same template, and T is the temperature. The verifier score is the log-odds of VALID against the remaining mass, rescaled to a bounded signal:

$$L(o) = \text{clip}_{[-1,1]} \left(\frac{\log \frac{\tilde{p}_{\text{VALID}} + \epsilon}{\tilde{p}_{\text{INVALID}} + \tilde{p}_{\text{UNKNOWN}} + \epsilon}}{3.0} \right),$$

with ϵ a small constant for numerical stability. Placing UNKNOWN in the denominator makes abstention count against a candidate rather than leaving it neutral: under closed-book inference an unverifiable candidate is one we prefer not to emit, and the conservative treatment is what keeps precision high on the small-set relations.

3.4 Residual Search Need and Adaptive Control

Candidate confidence estimates whether an observed candidate should be trusted. Completeness is handled separately through a residual search signal q_{res} , which estimates whether another action is likely to add useful verified information.

The signal aggregates three groups of relation-dependent gap indicators:

$$q_{\text{res}} = \text{clip}_{[0,1]}(r_t),$$

$$r_t = \max(\bar{\phi}_w, \phi_{\text{mandatory}}, \phi_{\text{blocking}}),$$

where $\bar{\phi}_w$ is the weighted mean of the soft gap signals admitted by the relation contract, $\phi_{\text{mandatory}}$ is one whenever a mandatory view has not yet been executed, and ϕ_{blocking} is a program-typed blocking condition. We take the maximum rather than a weighted sum so that a single unsatisfied mandatory view or blocking gap forces continuation regardless of how settled the remaining signals are.

The blocking term is used only where a hard gap is not already captured by mandatory views. `SMALLSET` and `LARGEOPENSET` relations set $\phi_{\text{blocking}} = 0$. `NUMERIC` relations set it to the normalized support of the strongest rival numeric cluster relative to the dominant one, so a competitive second cluster keeps the relation open. `NULLSINGLE` relations set it to the maximum of an unresolved existence gate and locality competition between the top two candidates, except that a confidently negative existence gate sets it to zero: the empty set is a valid answer, not a failure to recall.

The relation type also determines which soft signals enter $\bar{\phi}_w$: `SMALLSET` contracts use set instability and unresolved mass, `NUMERIC` contracts use numeric dispersion, and `LARGEOPENSET` contracts use marginal yield and facet coverage. `NULLSINGLE` contracts rely on the blocking term instead, since existence and locality already determine whether the relation may stop.

The controller stops when any of a fixed sequence of conditions holds: the call or token budget is exhausted, every mandatory view has run and the typed program declares the relation closed, no verification is still pending, and finally q_{res} has fallen below the relation’s stopping threshold. Acquisition is bounded in any case by a maximum number of steps per query. The threshold is part of the contract’s stopping policy; the component weights inside $\bar{\phi}_w$ are shared configuration constants. The scalar and its components play different roles: q_{res} is the last gate, deciding whether an otherwise-closable relation is in fact settled, while the individual gap signals it aggregates determine which action is worth taking next.

At each step the controller selects the highest-scoring legal action among running an untried view, querying a facet, verifying an unresolved candidate, performing a reverse check, resampling with a different template, and stopping:

$$A_t(a) = \lambda_y Y_t(a) + \lambda_g G_t(a) + \lambda_i I_t(a) - \lambda_c \text{cost}(a) - \lambda_d D_t(a),$$

where $Y_t(a)$ is the expected yield of a , $G_t(a)$

scores how directly a closes the specific gaps behind q_{res} , such as unresolved candidate mass, mechanism gaps, and facet gaps, $I_t(a)$ is the expected uncertainty reduction, $\text{cost}(a)$ is model-call cost, and $D_t(a)$ penalizes redundancy with actions already taken. The relation contract determines which actions are legal and which gaps count, while the utility coefficients are global constants of the frozen configuration; both are fixed in advance, so the policy is deterministic and budget-aware.

3.5 Relation-Aware Finalization

After acquisition stops, the relation contract maps the evidence state to the final object set. The principle shared by all four selectors is that finalization operates on the evidence state rather than on generated text: a string that was never resolved into supported evidence is not eligible for output. `SMALLSET` selectors emit validated entities and drop unresolved candidates; `NULLSINGLE` selectors emit at most one object and otherwise preserve the empty set; `LARGEOPENSET` selectors retain accepted candidates after normalization and deduplication; `NUMERIC` selectors operate on value clusters rather than raw strings.

Numeric finalization clusters parsed values by relative distance,

$$d(a, b) = \frac{|a - b|}{\max(|a|, |b|)},$$

$$a \sim b \iff d(a, b) \leq \tau,$$

which absorbs rounding and unit-conversion variation while keeping genuinely different reported values apart. The core selector uses $\tau = 0.025$, deliberately stricter than the 5% relative tolerance of the official evaluator, so that two values merge only when they agree well within the scoring margin. When no cluster reaches sufficient independent support, a second round of numeric acquisition arbitrates among the competing values at $\tau = 0.05$, the granularity the metric actually rewards. In either case a source-blind numeric judge compares the values without treating the upstream prediction as an additional vote, so a confidently generated but poorly supported number cannot win by repetition.

The same separation applies to existence. For city-of-death queries the system first resolves whether the subject is deceased, and only then asks for the city, with the response checked against a strict city parser. Treating existence as a prior gate rather than a post-filter is what lets the system return a principled empty set instead of a failed recall.

Relation	Macro Precision	Macro Recall	Macro F1
awardWonBy	0.3255	0.3707	0.3105
companyTradesAtStockExchange	0.8992	0.7963	0.7385
countryLandBordersCountry	0.9712	0.9295	0.9291
hasArea	0.6700	0.6700	0.6700
hasCapacity	0.2449	0.1633	0.1633
personHasCityOfDeath	0.9600	0.5900	0.5700
All relations	0.7268	0.6055	0.5878
Zero-object cases	0.6038	0.9412	0.7356

Table 2: Official hidden-test results. Zero-object cases denote rows whose gold object set is empty.

Relation	Direct	COVER-KBC
awardWonBy	0.079	0.250
companyTradesAtStockExchange	0.640	0.650
countryLandBordersCountry	0.941	0.959
hasArea	0.460	0.580
hasCapacity	0.103	0.250
personHasCityOfDeath	0.490	0.510
All relations	0.492	0.560

Table 3: Macro-F1 on the validation set. Direct denotes a single-pass 5-shot baseline using the same Mistral-Small-3.2-24B checkpoint as COVER-KBC. COVER-KBC values are those reported for the submitted system.

A related pattern appears in stock-exchange queries. An empty prediction triggers candidate-blind views targeting direct listing evidence, primary listing, ordinary-share listing, and anti-prior verification, since an empty output more often reflects failed elicitation than a genuine absence. Structural validation then rejects ticker symbols, indexes, regulators, market segments, country names, and company names, and an exchange is emitted only with sufficient independent support.

Together these selectors close the loop opened in Section 3.1: one contract fixes both which evidence a relation may acquire and how that evidence becomes an answer, so relation conditioning shapes the whole inference path rather than only the prompt.

4 Experimental Setup

We use the official AKBC Shared Task 2026 benchmark and evaluator.² Hidden-test labels are not released, so test results are reported by the official hidden-test evaluation service.

All neural inference uses a single frozen Mistral-Small-3.2-24B checkpoint.³ The model has 24,011,361,280 published parameters, below the

²<https://github.com/lm-kbc/dataset2026>

³Hugging Face model id: [mistralai/Mistral-Small-3.2-24B-Instruct-2506](https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506).

32B task limit, and is reused for all logical roles in the system. We run inference on Google Colab with one NVIDIA A100 GPU using the Hugging Face backend, mistral-common tokenizer, nf4 quantization, and bfloat16 dtype.

COVER-KBC is evaluated in a closed-book setting: no external retrieval, web search, external knowledge base, manual answer table, fine-tuning, LoRA, or additional neural model is used. All generation uses greedy decoding rather than sampling. Downstream parsing, aggregation, clustering, and final selection are deterministic given the model outputs.

5 Results

5.1 Hidden-Test Performance

Table 2 reports the official hidden-test results for COVER-KBC. The system achieves 0.5878 Macro-F1 overall. Performance is strongest on countryLandBordersCountry, where the relation has a bounded small-set structure and benefits from conservative evidence aggregation. companyTradesAtStockExchange also obtains strong performance, with high precision and improved recall from the escalation views that revisit predictions.

The main weakness is hasCapacity. The task defines capacity as the highest published spectator capacity, so the difficulty is not that venues have several reported figures but that the system does not reliably target the maximum among them: precision and recall are both low, indicating that the accepted cluster is often centred on a seated or pre-renovation figure rather than the intended one. The open-set awardWonBy relation is also challenging, mainly because many awards have long recipient lists and recall is difficult under closed-book inference; with only ten test subjects, its scores are also the noisiest in the table.

5.2 Comparison with a Same-Backbone Direct Baseline

Table 3 compares COVER-KBC with a single-pass 5-shot baseline using the same Mistral-Small-3.2-24B checkpoint; the direct baseline achieves 0.492 Macro-F1, while COVER-KBC reports 0.560, with the largest gains on `awardWonBy`, `hasArea`, and `hasCapacity`.

5.3 Relation-Level Observations

The results support the central design choice of COVER-KBC: different relation types require different inference and finalization behavior. Small-set relations benefit from strict validation and conservative set closure. Nullable relations require preserving empty outputs unless the evidence supports a singleton. Numeric relations benefit from multi-view scalar clustering, but remain sensitive to ambiguity in the target definition. Large open-set relations remain recall-limited because the system must discover many valid objects without external retrieval.

6 Conclusion

We presented COVER-KBC, a closed-book KBC system that treats object-set prediction as relation-typed active evidence acquisition. Instead of mapping one prompt response directly to a final answer, COVER-KBC accumulates structured evidence, separates candidate confidence from residual search need, verifies candidates blindly, and finalizes outputs with relation-specific selectors.

Using a single frozen Mistral-Small-3.2-24B checkpoint under the 32B parameter budget, COVER-KBC achieves 0.5878 Macro-F1 on the official hidden test set. The results show that heterogeneous KBC relations benefit from specialized inference programs: small-set relations benefit from conservative set closure, nullable relations require explicit empty-answer handling, numeric relations benefit from multi-view clustering, and open-set relations require continued discovery. These findings support relation-aware evidence acquisition as a practical direction for closed-book knowledge-base construction.

Limitations

COVER-KBC is designed for the six relations in the shared task. Its relation contracts, evidence views, parsers, and final selectors are manually specified, so extending the system to new relations

would require additional relation-level engineering. The approach is therefore not a universal relation-agnostic KBC method.

The system is also limited by the factual knowledge stored in the frozen language model. Because no retrieval, external knowledge base, or additional training is used, COVER-KBC can miss facts that are absent, stale, or weakly represented in the model parameters. This is most visible for `awardWonBy`, which remains recall-limited, and for `hasCapacity`, where the model rarely commits to the maximum published figure that the task definition requires.

Finally, multi-step evidence acquisition improves control but increases inference cost compared with a single-prompt baseline. Hidden-test labels are not released, so detailed error analysis on the official test set is limited to official aggregate scores and relation-level aggregates.

A Hyperparameters

Table 4 lists the candidate-scoring weights, verifier calibration parameters, and per-query budgets of the released configuration. Most values are constants of that configuration; the action-planner coefficients are calibrated once on the provided training split and then frozen. No neural parameters are updated at any point, and no validation or test data is used for calibration.

Component	Parameter	Value
Candidate score	α (support)	1.0
	β (verifier)	0.6
	γ (cross-model)	0.5 [†]
	δ (contradiction)	1.5
	η (disagreement)	1.0
Verifier	T	1.0
	ϵ	10 ⁻⁶
	log-odds clip divisor	3.0
Budget	call budget per query	12 [‡]
	token budget per query	6000 [‡]
Stopping	residual threshold	0.15 [§]

Table 4: Configuration of the submitted system. Scoring and verifier values are global constants; the stopping threshold is set per relation by the contract. [†]Configured but inactive: $X(o) = 0$ when all neural roles share one checkpoint. [‡]Core controller cap; relation-specific escalation calls are separately capped in the released configuration. [§]Contract default.

Acknowledgments

This research is partially supported by the research funding from the Faculty of Information Technology, University of Science, Ho Chi Minh City, Vietnam.

References

- Pranjal Aggarwal, Aman Madaan, Yiming Yang, and Mausam. 2023. [Let’s sample step by step: Adaptive-consistency for efficient reasoning and coding with LLMs](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12375–12396, Singapore. Association for Computational Linguistics.
- Edouard Albert-Roulhac and Amal Zouaq. 2025. [Re-WiSe: Relation-wise self-consistency for LLM probing](#). In *Joint Proceedings of the 3rd Workshop on Knowledge Base Construction from Pre-Trained Language Models and the 4th Challenge on Language Models for Knowledge Base Construction (KBC-LM+LM-KBC 2025) co-located with the 24th International Semantic Web Conference (ISWC 2025)*, volume 4041 of *CEUR Workshop Proceedings*, Nara, Japan. CEUR-WS.org.
- Dimitrios Alivanistos, Selene Báez Santamaría, Michael Cochez, Jan-Christoph Kalo, Emile van Krieken, and Thiviyan Thanapalasingam. 2022. [Prompting as probing: Using language models for knowledge base construction](#). In *ISWC 2022 Posters, Demos and Industry Tracks*, CEUR Workshop Proceedings. CEUR-WS.org.
- Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. [Self-RAG: Learning to retrieve, generate, and critique through self-reflection](#). In *The Twelfth International Conference on Learning Representations*.
- Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun, Lingyong Yan, Meng Liao, Tong Xue, and Jin Xu. 2021. [Knowledgeable or educated guess? revisiting language models as knowledge bases](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1860–1874, Online. Association for Computational Linguistics.
- Arunav Das, Nadeen Fathallah, and Nicole Obretincheva. 2024. [Navigating nulls, numbers and numerous entities: Robust knowledge base construction from large language models](#). In *Joint Proceedings of the 2nd Workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM 2024) and the 3rd Challenge on Language Models for Knowledge Base Construction (LM-KBC 2024) co-located with the 23rd International Semantic Web Conference (ISWC 2024)*, volume 3853 of *CEUR Workshop Proceedings*, Baltimore, USA. CEUR-WS.org.
- Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. [Chain-of-verification reduces hallucination in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand. Association for Computational Linguistics.
- Xin Dong, Evgeniy Gabrilovich, Jeremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. [Knowledge vault: A web-scale approach to probabilistic knowledge fusion](#). In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’14*, pages 601–610, New York, NY, USA. Association for Computing Machinery.
- Jingbo He and Simon Razniewski. 2025. [Combining self-retrieval-augmented generation with divide-and-conquer for language model-based knowledge base construction](#). In *Joint Proceedings of the 3rd Workshop on Knowledge Base Construction from Pre-Trained Language Models and the 4th Challenge on Language Models for Knowledge Base Construction (KBC-LM+LM-KBC 2025) co-located with the 24th International Semantic Web Conference (ISWC 2025)*, volume 4041 of *CEUR Workshop Proceedings*, Nara, Japan. CEUR-WS.org.
- Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. [Active retrieval augmented generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.
- Jan-Christoph Kalo, Tuan-Phong Nguyen, Simon Razniewski, and Bohui Zhang. 2026. [AKBC Shared Task 2026: Predicting complete knowledge base entries from language models](#). In *Proceedings of the 10th Workshop on Automated Knowledge Base Construction (AKBC) at EMNLP 2026*. Association for Computational Linguistics.
- Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. 2024. [Optimizing instructions and demonstrations for multi-stage language model programs](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA. Association for Computational Linguistics.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. [Language models as knowledge bases?](#) In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

- Simon Razniewski, Hiba Arnaout, Shrestha Ghosh, and Fabian M. Suchanek. 2024. [Completeness, recall, and negation in open-world knowledge bases: A survey](#). *ACM Computing Surveys*, 56(6):1–42.
- Raghuveer Thirukovalluru, Yukun Huang, and Bhuwan Dhingra. 2024. [Atomic self-consistency for better long form generations](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12681–12694, Miami, Florida, USA. Association for Computational Linguistics.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. [Calibrate before use: Improving few-shot performance of language models](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.