LM-KBC 2025: 4th Challenge on Knowledge Base Construction from Pre-trained Language Models

Jan-Christoph Kalo[1], Simon Razniewski[2], Bohui Zhang[3] and Tuan-Phong Nguyen[4]

[1] *University of Amsterdam*
[2] *ScaDS.AI & TU Dresden*
[3] *King's College London*
[4] *VNU University of Engineering and Technology*

Abstract

Pretrained language models (LMs) have significantly advanced a variety of semantic tasks and have shown promise as sources of knowledge elicitation. While prior work has studied this ability through probing or prompting, the potential of LMs for large-scale knowledge base construction remains underexplored. The fourth edition of the LM-KBC Challenge invited participants to build knowledge bases directly from LMs, given specific subjects and relations. Unlike existing probing benchmarks, the challenge imposed no simplifying assumptions on relation cardinality—allowing a subject entity to be linked to zero, one, or multiple object entities. To ensure accessibility, the challenge featured a single track based the same LLM to be used by all participants. Five submissions were received, which explored a variety of ideas from self-consistency, self-RAG, reasoning, and prompt optimization.

1. Introduction

Large language models (LLMs) such as Qwen [1], Llama [2], and ChatGPT [3] are optimized for masked language modeling or text completion and have achieved remarkable success on a wide range of downstream NLP tasks, including question answering, information retrieval, and machine translation. More recently, LLMs have attracted attention for their potential to directly produce structured knowledge from their parameters. This is promising, as current knowledge bases (KBs) such as Wikidata [4] and ConceptNet [5], though central to the Semantic Web ecosystem, remain inherently incomplete [6]. KB construction is particularly challenging due to the optional nature of many relations (e.g., *place of death, cause of death, parent organization*) and the presence of multiple correct objects for a single subject–relation pair (e.g., *shares border, employer, speaks language*). Moreover, KBs must be materialized for trustworthy and consistent downstream usage [7].

Traditional approaches to **knowledge base construction (KBC)** have leveraged unstructured text [8, 9], crowdsourcing [10, 11], and semi-structured resources [12, 13, 14]. Automated KBC has long been a core topic in the Semantic Web community, spanning decades of research on knowledge extraction, consolidation, and schema matching. The seminal LAMA paper [15] demonstrated that language models could rank correct object tokens highly when prompted with a subject–relation query. While subsequent work has reported both progress [16, 17] and criticism [18, 19, 20], the potential of LMs for KBC remains underexplored. Importantly, the LAMA benchmark and its variants are not designed for true KB construction. Although LLMs are increasingly studied in Semantic Web tasks such as entity recognition, relation extraction, and reasoning, most evaluations of factual knowledge extraction remain rooted in NLP-style benchmarks with simplified assumptions.

This challenge seizes the opportunity to bridge the gap by exploring how LLMs can contribute to practical KB construction. Continuing previous efforts [21, 22, 23, 24], the 4th edition focuses on leveraging a single locally runnable LLM to construct KBs without prior knowledge of relation

Joint proceedings of KBC-LM and LM-KBC @ ISWC 2025

j.c.kalo@uva.nl (J. Kalo); simon.razniewski@tu-dresden.de (S. Razniewski); bohui.zhang@kcl.ac.uk (B. Zhang); tuanphong@vnu.edu.vn (T. Nguyen)
0000-0002-5492-2292 (J. Kalo); 0000-0002-5410-218X (S. Razniewski); 0000-0001-5430-1624 (B. Zhang); 0000-0002-1674-2741 (T. Nguyen)

cardinalities. Specifically, given a subject–relation pair, participants were asked to design an LLM-based system that generates candidate subject–relation–object triples, and decides whether to accept or reject each one. The predictions were evaluated using F1-score.

2. Task Description

In the LM-KBC Challenge, the knowledge base construction task is defined as follows: given a subject entity s and a relation r, the goal is to generate all correct object entities $[o_1, o_2, \ldots, o_k]$ by probing language models. For example, given the tuple (Greece, `countryLandBorderCountry`), a participant might query the language model with a prompt such as "Greece shares a border with [MASK]". The system should then output country entities like [Albania, North Macedonia, Bulgaria, Turkey], in any order. Similarly, for numeric-answer relations, such as (Wembley Stadium in London, `hasCapacity`), the expected output would be ["90000"].

Participants are required to build LM-based systems that produce entity labels without relying on external resources (e.g., web search engines, retrieval-augmented generation), i.e., submitted systems had to be fully self-contained. For comparison, we released a baseline method based on prompt templates, covering both question-style prompts and fill-in-the-blank templates.

3. Dataset Construction

Relation	Train	Validation	Test	Special features
`awardWonBy`	10	10	10	Many objects per subject
`companyTradesAtStockExchange`	100	100	100	Null values possible
`countryLandBordersCountry`	68	68	67	Null values possible
`hasArea`	100	100	100	Object is numeric (unit: km^2)
`hasCapacity`	100	100	100	Object is numeric
`personHasCityOfDeath`	100	100	100	Null values possible

Table 1
Dataset statistics. The numbers indicate the count of subject-relation pairs.

The dataset was built by querying Wikidata and manually refining the results to reduce errors and improve quality. Compared with previous editions [21, 22], the 2025 version focuses on six challenging relations with distinctive characteristics, enabling participants to design approaches tailored to specific problem types. These relations fall into the following categories:

1. **Relations with many missing objects** (e.g., a person's place of death, or the stock exchange where a company is listed).
2. **Relations with long object lists** (e.g., the list of award winners in a given field).
3. **Standard relations** carried over from the previous edition.

For each relation, up to 100 subject entities are provided for the training, validation, and hidden test sets used in challenge evaluation. The relations were carefully selected to ensure diversity, with subject entities spanning different types such as persons, countries, and organizations. The subject–object pairs were automatically sampled from Wikidata under the following constraints:

1. **Balanced object list lengths:** Longer lists were oversampled to avoid dominance by single-object examples.
2. **Balanced subject popularity:** Using proxies such as total Wikidata statements or web hits, we ensured roughly a 50/50 split between popular and long-tail subject entities for each relation.
3. **Balanced object complexity:** Both single-token and multi-token object entities were included.

Rank	CodaLab Username	Method	Average F1-score	Cite
1	edarsem	Relation-Wise Self-consistency	0.4439	Albert-Roulhac and Zouaq, 2025
2	JingboHe	Self-RAG and DaC	0.4052	He and Razniewski, 2025
3	acmc	Soft Thinking	0.3977	Creo et al., 2025
4	isam	LLM-as-a-Judge	0.2406	Sam, 2025
5	aclay	Prompt optimization	0.2159	Clay et al., 2025

Table 2
Leaderboard ranking of participating systems, including CodaLab usernames, methods, and citations.

The 2025 dataset is publicly available on GitHub[1]. Participating systems submit their predictions on the CodaLab platform[2] [25], where final scores are computed on the hidden test set.

4. Comparison with Previous Editions

Compared with previous editions of the LM-KBC challenge [21, 22], the 4th edition introduced the following characteristics:

1. **More diverse relations:** As in the 3rd edition, we continue to use a smaller, representative set of relations grouped into topical categories, enabling participants to better tailor their approaches. Compared with the 3rd edition, the numeric-answer relations were replaced by two new ones.
2. **Single parameter-bounded track:** The challenge continues to follow a single-track format. Building on last year's fixed limit of 10 billion parameters for LLMs, we further standardized the setting by requiring participants to use Qwen3-8B [1].
3. **High data quality:** As before, substantial manual effort was invested to ensure the dataset meets the highest quality standards.

5. Results of Submissions

Table 2 presents the final ranking of the five participating systems, all of which outperformed the baseline pipeline.

The top-performing system, Relation-Wise Self-consistency (ReWiSe) by Albert-Roulhac and Zouaq, achieved the highest average F1-score of 0.4439. Their method generates **synthetic chain-of-thought reasoning paths** and applies **relation-wise self-consistency** to aggregate multiple LLM outputs. The other systems employed distinct strategies:

He and Razniewski proposed a hybrid system that combines two specialized pipelines. It processes general relations using a **Self-RAG** approach with a description-first, extraction-second design. For the complex `awardWonBy` relation, it uses a **divide-and-conquer** module to aggregate and filter candidates from decomposed subqueries.

Creo et al. proposed Soft Thinking, which inserts **soft prompt embeddings within the chain-of-thought reasoning** section.

Clay et al. implemented a multi-stage pipeline where the LLM acts as both as **generator and judge**. Their method generates multiple candidate triples, iteratively judges and re-runs low-scoring ones, filters the results by consensus, and applies a final multi-pass quality judge to select the final outputs.

Sam used a mixed-strategy prompting approach, employing three distinct system prompts based on relation type and **six unique, detailed user prompts for each individual relation**, with no additional post-processing of the model's output. Detailed descriptions of all systems are provided in the corresponding papers included in the proceedings.

[1] https://github.com/lm-kbc/dataset2025
[2] https://codalab.lisn.upsaclay.fr/competitions/23218

From the detailed per-relation results in Table 3, we observe that no single method consistently outperforms the others across all relations. The ReWiSe system proposed by Albert-Roulhac and Zouaq achieved the highest F1-score in three out of six relations. The Self-RAG and DaC, Soft Thinking, and LLM-as-a-Judge methods achieved the best performance on `awardWonBy`, `companyTradesAtStockExchange`, and `hasCapacity`, respectively. Overall, the relations `awardWonBy`, `hasArea`, and `hasCapacity` remain particularly challenging. Compared with the baseline, system improvements on these relations were marginal, and their absolute F1-scores stayed relatively low.

CodaLab Username	Method	Precision	Recall	F1-score
`awardWonBy`				
edarsem	Relation-Wise Self-consistency	0.0540	0.1929	0.0825
JingboHe	Self-RAG and DaC	0.1639	0.2052	**0.1759**
acmc	Soft Thinking	0.1609	0.0370	0.0573
isam	LLM-as-a-Judge	0.1910	0.1591	0.1390
aclay	Prompt optimization	0.7000	0.0000	0.0000
baseline	-	0.2399	0.0900	0.1170
`companyTradesAtStockExchange`				
edarsem	Relation-Wise Self-consistency	0.6700	0.6290	0.5427
JingboHe	Self-RAG and DaC	0.6950	0.5440	0.5057
acmc	Soft Thinking	0.6667	0.6470	**0.5547**
isam	LLM-as-a-Judge	0.1783	0.6043	0.1708
aclay	Prompt optimization	0.8300	0.4300	0.3900
baseline	-	0.1850	0.5907	0.1670
`countryLandBordersCountry`				
edarsem	Relation-Wise Self-consistency	0.9240	0.8942	**0.8900**
JingboHe	Self-RAG and DaC	0.9480	0.8279	0.8649
acmc	Soft Thinking	0.8173	0.7909	0.7711
isam	LLM-as-a-Judge	0.7377	0.8172	0.6909
aclay	Prompt optimization	0.5821	0.1493	0.1343
baseline	-	0.7684	0.8125	0.7025
`hasArea`				
edarsem	Relation-Wise Self-consistency	0.3200	0.3200	**0.3200**
JingboHe	Self-RAG and DaC	0.3100	0.3100	0.3100
acmc	Soft Thinking	0.1900	0.1900	0.1900
isam	LLM-as-a-Judge	0.2600	0.2600	0.2600
aclay	Prompt optimization	0.9800	0.0300	0.0300
baseline	-	0.2400	0.2400	0.2400
`hasCapacity`				
edarsem	Relation-Wise Self-consistency	0.1400	0.0900	0.0900
JingboHe	Self-RAG and DaC	0.1900	0.1100	0.1100
acmc	Soft Thinking	0.0900	0.0900	0.0900
isam	LLM-as-a-Judge	0.1500	0.1500	**0.1500**
aclay	Prompt optimization	0.8200	0.0000	0.0000
baseline	-	0.0400	0.0400	0.0400
`personHasCityOfDeath`				
edarsem	Relation-Wise Self-consistency	0.8500	0.6400	**0.5600**
JingboHe	Self-RAG and DaC	0.6500	0.6500	0.4100
acmc	Soft Thinking	0.9300	0.6000	0.5400
isam	LLM-as-a-Judge	0.0900	0.6600	0.0900
aclay	Prompt optimization	0.9100	0.5800	0.5200
baseline	-	0.0800	0.6500	0.0800

Table 3
Per-relation macro-averaged results. Systems are listed with their CodaLab usernames and methods, ranked consistently with Table 2.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] A. Yang, et al., Qwen3 Technical Report, 2025. URL: https://arxiv.org/abs/2505.09388. arXiv:2505.09388.

[2] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, LLaMA: Open and Efficient Foundation Language Models, 2023. URL: https://arxiv.org/abs/2302.13971. arXiv:2302.13971.

[3] O. team, GPT-4 Technical Report, 2024. URL: https://arxiv.org/abs/2303.08774. arXiv:2303.08774.

[4] D. Vrandečić, M. Krötzsch, Wikidata: A Free Collaborative Knowledgebase, Commun. ACM (2014). URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.

[5] R. Speer, J. Chin, C. Havasi, ConceptNet 5.5: An Open Multilingual Graph of General Knowledge, in: AAAI, 2017.

[6] S. Razniewski, H. Arnaout, S. Ghosh, F. Suchanek, Completeness, recall, and negation in open-world knowledge bases: A survey, ACM Computing Surveys (2024).

[7] B. Zhang, E. Koutsiana, Y. Zhao, A. Meroño-Peñuela, E. Simperl, Trustworthy Knowledge Graphs: Practices and Approaches, in: Handbook on Neurosymbolic AI and Knowledge Graphs, IOS Press, 2025.

[8] N. Nakashole, M. Theobald, G. Weikum, Scalable Knowledge Harvesting with High Precision and High Recall, in: WSDM, 2011. URL: https://doi.org/10.1145/1935826.1935869. doi:10.1145/1935826.1935869.

[9] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge Vault: A Web-Scale Approach to Probabilistic Knowledge Fusion, in: KDD, 2014. URL: https://doi.org/10.1145/2623330.2623623. doi:10.1145/2623330.2623623.

[10] M. Acosta, A. Zaveri, E. Simperl, D. Kontokostas, S. Auer, J. Lehmann, Crowdsourcing Linked Data Quality Assessment, in: H. Alani, L. Kagal, A. Fokoue, P. Groth, C. Biemann, J. X. Parreira, L. Aroyo, N. Noy, C. Welty, K. Janowicz (Eds.), ISWC, 2013. URL: https://link.springer.com/chapter/10.1007/978-3-642-41338-4_17. doi:10.1007/978-3-642-41338-4_17.

[11] A. Kobren, T. Logan, S. Sampangi, A. McCallum, Domain Specific Knowledge Base Construction via Crowdsourcing, in: AKBC, 2014.

[12] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge, in: SIGMOD, Association for Computing Machinery, 2008. URL: https://doi.org/10.1145/1376616.1376746. doi:10.1145/1376616.1376746.

[13] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, DBpedia: A Nucleus for a Web of Open Data, in: ISWC, 2007. URL: https://doi.org/10.1007/978-3-540-76298-0_52. doi:10.1007/978-3-540-76298-0_52.

[14] J. Hoffart, F. M. Suchanek, K. Berberich, E. Lewis-Kelham, G. de Melo, G. Weikum, YAGO2: Exploring and Querying World Knowledge in Time, Space, Context, and Many Languages, in: WWW, 2011. URL: https://doi.org/10.1145/1963192.1963296. doi:10.1145/1963192.1963296.

[15] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language Models as Knowledge Bases?, 2019. URL: https://arxiv.org/abs/1909.01066. arXiv:1909.01066.

[16] K. Guu, K. Lee, Z. Tung, P. Pasupat, M.-W. Chang, REALM: Retrieval-Augmented Language Model Pre-Training, 2020. URL: https://arxiv.org/abs/2002.08909. arXiv:2002.08909.

[17] A. Roberts, C. Raffel, N. Shazeer, How Much Knowledge Can You Pack Into the Parameters of a Language Model?, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), EMNLP, 2020. URL: https://aclanthology.org/2020.emnlp-main.437/. doi:10.18653/v1/2020.emnlp-main.437.

[18] R. T. McCoy, E. Pavlick, T. Linzen, Right for the Wrong Reasons: Diagnosing Syntactic Heuristics

in Natural Language Inference, in: ACL, 2019. URL: https://aclanthology.org/P19-1334/. doi:10.18653/v1/P19-1334.

[19] N. Kassner, H. Schütze, Negated and Misprimed Probes for Pretrained Language Models: Birds Can Talk, But Cannot Fly, in: ACL, 2020. URL: https://aclanthology.org/2020.acl-main.698/. doi:10.18653/v1/2020.acl-main.698.

[20] B. Cao, H. Lin, X. Han, L. Sun, L. Yan, M. Liao, T. Xue, J. Xu, Knowledgeable or Educated Guess? Revisiting Language Models as Knowledge Bases, in: C. Zong, F. Xia, W. Li, R. Navigli (Eds.), ACL, Online, 2021. URL: https://aclanthology.org/2021.acl-long.146/. doi:10.18653/v1/2021.acl-long.146.

[21] J.-C. Kalo, S. Singhania, S. Razniewski, J. Z. Pan, et al., LM-KBC 2023: 2nd Challenge on Knowledge Base Construction from Pre-trained Language Models, in: Joint proceedings of 1st workshop on Knowledge Base Construction from Pre-Trained Language Models (KBC-LM) and the 2nd challenge on Language Models for Knowledge Base Construction (LM-KBC), 2023. URL: https://ceur-ws.org/Vol-3577/paper0.pdf.

[22] J.-C. Kalo, T.-P. Nguyen, S. Razniewski, B. Zhang, Preface: LM-KBC Challenge 2024, in: Joint Proceedings of the KBC-LM Workshop and the LM-KBC Challenge 2024, 2024. URL: https://ceur-ws.org/Vol-3853/paper0.pdf.

[23] D. Alivanistos, S. B. Santamaría, M. Cochez, J.-C. Kalo, E. van Krieken, T. Thanapalasingam, Prompting as Probing: Using Language Models for Knowledge Base Construction, 2023. URL: https://arxiv.org/abs/2208.11057. arXiv:2208.11057.

[24] B. Zhang, I. Reklos, N. Jain, A. M. Peñuela, E. Simperl, Using Large Language Models for Knowledge Engineering (LLMKE): A Case Study on Wikidata, 2023. URL: https://arxiv.org/abs/2309.08491. arXiv:2309.08491.

[25] A. Pavao, I. Guyon, A.-C. Letournel, D.-T. Tran, X. Baro, H. J. Escalante, S. Escalera, T. Thomas, Z. Xu, CodaLab Competitions: An Open Source Platform to Organize Scientific Challenges, Journal of Machine Learning Research (2023). URL: http://jmlr.org/papers/v24/21-1436.html.