Knowledge-Rich Embeddings for Tabular Learning

Félix Lefebvre
SODA Team, Inria Saclay
felix.lefebvre@inria.fr

Myung Jun Kim
SODA Team, Inria Saclay

Gaël Varoquaux
SODA Team, Inria Saclay
Probabl, France

Abstract

Tables have their own structure, calling for dedicated tabular learning methods with the right inductive bias. These methods outperform direct applications of language models, which struggle with the heterogeneous features typical in tables, such as numerical data or dates. Yet, many tables contain text that refers to real-world entities, and most tabular learning methods ignore the external knowledge that such strings could unlock. Which knowledge-rich representations should tabular learning leverage? While large language models (LLMs) encode implicit factual knowledge, knowledge graphs (KGs) share the relational structure of tables and come with the promise of better-controlled knowledge. Studying tables in the wild, we assemble 105 tabular learning datasets comprising text. We find that knowledge-rich representations from LLMs or KGs boost prediction and, combined with simple linear models, markedly outperform strong tabular baselines. Larger LLMs and larger KGs both provide greater gains. On datasets where all entities are linked to a KG, LLMs and KG models of similar size perform similarly, suggesting that the benefit of LLMs over KGs is to solve the entity linking problem. Our results highlight that external knowledge is a powerful but underused ingredient for advancing tabular learning.

1 Introduction: background knowledge for tabular learning

Tabular data is prominent in machine learning, often containing text entries that refer to real-world entities such as company names, drugs, or locations. While methods like gradient-boosted decision trees and modern table foundation models excel with numerical data [Chen and Guestrin, 2016, Hollmann et al., 2025], the text processing is relegated to a preprocessing step that typically relies on superficial encodings (e.g., character n-grams), discarding the rich, real-world knowledge embedded in these strings. This is a missed opportunity, as external knowledge could significantly boost predictive power, especially in low-data regimes.

A scalable way to inject this knowledge is to use vector representations pretrained on large-scale sources [Cvetkov-Iliev et al., 2023, Grinsztajn et al., 2023, Lefebvre and Varoquaux, 2025]. Two main paradigms exist for this: knowledge graphs (KGs) and large language models (LLMs). General-purpose KGs [Bollacker et al., 2008, Vrandečić and Krötzsch, 2014, Suchanek et al., 2024] offer structured, curated facts, but their use is hampered by incompleteness and the difficult "symbol grounding" problem of linking messy text to canonical entities [Mendes et al., 2011]. In contrast, LLMs implicitly encode vast world knowledge from web-scale text and can embed any string, effectively sidestepping the entity linking challenge. However, this comes at the cost of reliability, as their knowledge is statistical, not factual, and prone to hallucination [Ji et al., 2023].

This raises a critical question: which source of knowledge is more effective for tabular learning? To investigate this, we conduct a large-scale empirical study on 105 tabular datasets, assembled from three diverse sources. We compare knowledge-rich representations from LLMs and KG models, evaluating their impact on downstream prediction tasks. Our key findings are:

AI for Tabular Data workshop at EurIPS 2025.

1. **Good representations matter more than sophisticated downstream tabular learners:** Both LLM and KG embeddings, when paired with a simple linear model, outperform strong tabular baselines that use superficial text encodings.
2. **Scale brings gains:** Performance improves with the size of both LLMs and KGs.
3. **Entity linking is the key bottleneck:** On a subset of tables where entities are pre-linked to a KG, KG embeddings perform on par with LLMs of similar size. This suggests that the primary advantage of LLMs is not superior knowledge, but their ability to implicitly solve the entity linking problem.

2 Related work

Tabular learning with text features Tabular learning has long been dominated by gradient-boosted decision trees [Chen and Guestrin, 2016], but recent deep learning approaches, including table foundation models [Hollmann et al., 2025, Ma et al., 2024, Qu et al., 2025], now often outperform them [Erickson et al., 2025]. A shared limitation, however, is the absence of specific handling of text features, which are typically vectorized using superficial methods like TF-IDF that ignore the semantics and external knowledge embedded in the strings.

To address this, recent work has explored using external knowledge sources. One approach leverages LLMs by serializing table rows into text to be processed by an LLM [Hegselmann et al., 2023, Gardner et al., 2024]. An alternative paradigm uses KGs to pretrain tabular models [Kim et al., 2024, 2025]. Prior comparative studies have shown that embeddings from language models outperform traditional substring-based encoders, especially in diverse-entry and low-data regimes [Grinsztajn et al., 2023, Kasneci and Kasneci, 2024]. However, these works do not provide a direct comparison between knowledge sourced from LLMs versus structured KGs, a gap our study aims to fill.

Learning representations from knowledge sources Knowledge can be extracted from KGs by learning embeddings from the graph structure [Bordes et al., 2013, Yang et al., 2014, Trouillon et al., 2016, Sun et al., 2019] or by using language models to encode the textual descriptions of entities and relations [Wang et al., 2021, Saxena et al., 2022]. A related line of work refines general-purpose LLMs on knowledge-base data to improve their factual grounding and performance on knowledge-intensive tasks [Sun et al., 2020, Feng et al., 2023]. Our work evaluates and contrasts these different representation strategies in the context of tabular learning.

3 Methodology: a benchmark for table background knowledge

105 tabular datasets We assemble a diverse benchmark from three sources: TextTabBench [Mráz et al., 2025], CARTE [Kim et al., 2024], and WikiDBs [Vogel et al., 2024], covering regression, binary, and multi-class classification tasks (Table 1). To focus on text-based knowledge, we remove all numerical columns and apply standard preprocessing (see Appendix A), ensuring each dataset has at least 1,050 rows. For a controlled comparison between LLMs and KGs, we identify a subset of 15 tables where entries are unambiguously linked to Wikidata5M [Wang et al., 2021]. This allows us to evaluate pure KG models in a setting where the entity linking problem is solved. We also create smaller KG versions by filtering out low-degree entities and retaining the largest connected component, to study the impact of KG size (Table 2).

Table 1: Task distribution across sources of tables.

Source	b-clf	m-clf	reg	Total
TextTabBench	5	2	10	**17**
CARTE	11	0	40	**51**
WikiDBs	1	21	15	**37**
Total	**17**	**23**	**65**	**105**

Table 2: Knowledge graph datasets. All graphs use the same 822 relations.

	# entities	# triples	deg.
Wikidata5M	4.6M	20.6M	-
Wikidata3M	3.2M	15.5M	3
Wikidata2M	2.1M	11.5M	4
Wikidata1M	1.1M	6.8M	6
Wikidata500k	0.5M	3.1M	9

Evaluation pipeline Figure 4 shows our pipeline to evaluate representations on downstream tasks. For each dataset, we sample 1,024 training rows, and 1,024 testing rows (or all remaining rows if fewer are available), and average results over 10 random seeds.

Embedding models We compare a wide array of representations:

- **Non-pretrained baseline:** TF-IDF followed by SVD, via the `Skrub` library.
- **Pure LLMs:** A diverse set including Llama 3 [Dubey et al., 2024], Qwen3 [Zhang et al., 2025], RoBERTa [Liu et al., 2019], and others to study the effect of scale and architecture.
- **Hybrid LLM+KG models:** Models that refine LLMs on relational data, such as ERNIE 2.0, KGT5, and TabuLa-8B, each time compared against their corresponding base LLM.
- **Pure KG models:** For linked tables, we use classic KG embeddings (DistMult, TransE, ComplEX, and RotatE) trained on Wikidata5M and its subsets.

Table serialization and downstream estimators We serialize each table row into a natural language prompt (*e.g.*, "The `<col_a>` is `<val_a>`. The `<col_b>` is `<val_b>`.", details in Appendix A), enabling LLMs to generate context-aware embeddings. These embeddings are then used to train three representative tabular learners: Ridge regression, XGBoost, and the TabPFNv2 foundation model. For the latter two, we use PCA to reduce dimensionality to $d = 300$ and $d = 500$, respectively.

4 Results: knowledge representations for tabular learning

4.1 Knowledge-rich representations boost tabular learning

Good representations matter more than advanced tabular models Our central finding is that text representation quality is paramount. As shown in Figure 1a, a simple linear model (Ridge) fed with high-quality embeddings from modern LLMs consistently outperforms sophisticated tabular learners like XGBoost and TabPFNv2 that use superficial text encodings. This suggests that for tables

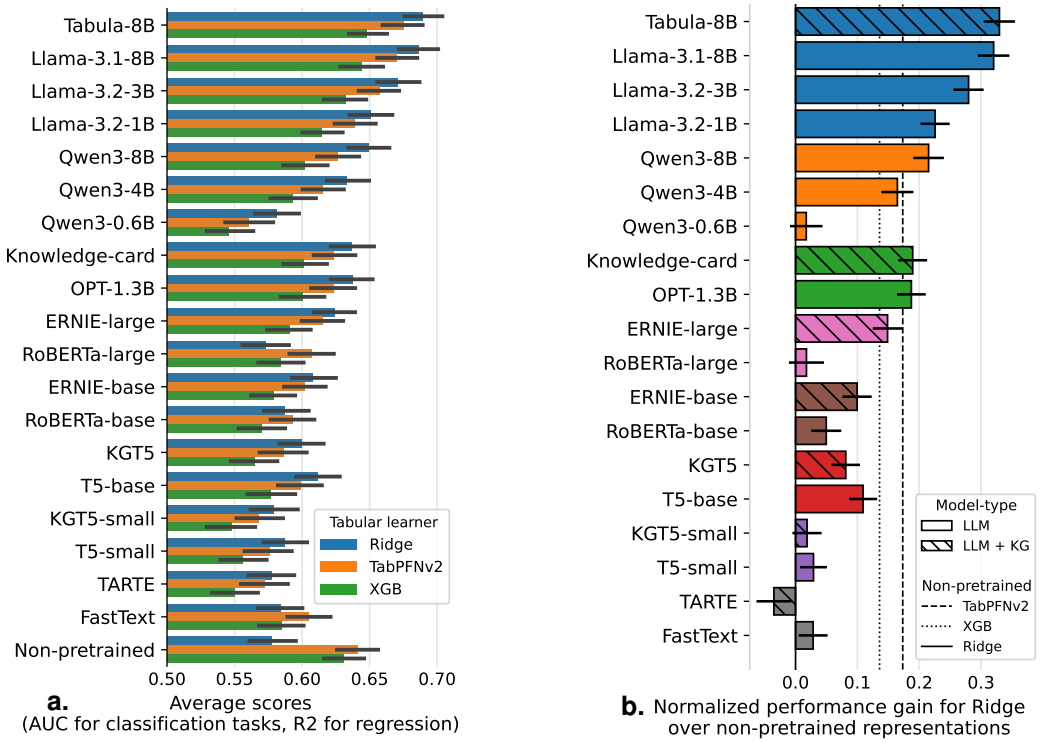

Figure 1: Performance gain of the various knowledge-rich representations compared to a non-pretrained baseline – **a.** Comparisons including three tabular learners: Ridge, XGBoost, and TabPFNv2; absolute scores. – **b.** Relative improvements to non-pretrained string representations, when using a ridge model as a tabular learner; normalized scores (0 is 10% worse, 1 is the best score observed). – Appendix Figure 5 gives critical difference diagrams across all methods and datasets.

rich in text, the primary performance bottleneck is not the downstream learning algorithm but the semantic poverty of the input features.

Modern tabular models struggle with high-dimensional embeddings Interestingly, the performance benefit of advanced tabular learners disappears when they are given knowledge-rich embeddings. XGBoost and TabPFNv2, when paired with LLM embeddings, underperform the simple Ridge model (Figure 1a). A likely cause is the need for aggressive dimensionality reduction (PCA) to make these embeddings compatible with the tabular learners, which may discard valuable information. This could also come from a mismatch between the inductive biases of current tabular models and the dense, rotationally-invariant nature of modern text representations [Grinsztajn et al., 2022].

The benefit of refining LLMs on relational data is unclear We find no consistent evidence that refining LLMs on relational data (hybrid models) improves performance. As seen in Figure 1b, while ERNIE 2.0 shows a clear gain over its RoBERTa base, KGT5 does not improve upon T5, and TabuLa-8B is on par with its Llama base. The effectiveness of such refinement appears highly dependent on the specific model and pretraining strategy, warranting further investigation.

4.2 LLMs vs. KGs: entity linking is the bottleneck

To isolate the value of knowledge from the challenge of entity linking, we evaluate performance on a subset of 15 tables where all text entries are pre-linked to Wikidata5M entities. This allows a direct comparison between pure LLM embeddings and pure KG embeddings.

Scale is key Figure 2 shows a clear trend for both LLMs and KGs: bigger is better. Performance improves steadily with the size of the LLM and the size of the KG used for training embeddings. For KGs, smaller graphs cover fewer entities, leading to a sharp performance drop as entity coverage decreases. LLMs exhibit a softer degradation, as they can still generate a (less informed) embedding for any string.

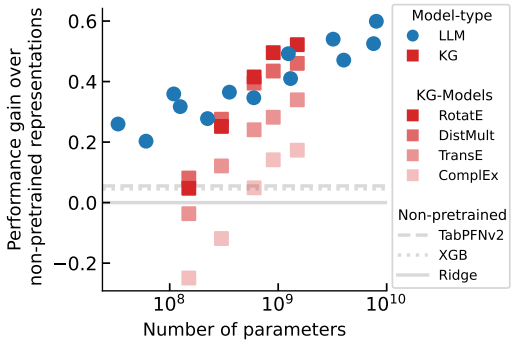

Figure 2: Comparing pure-KG to pure-LLM approaches on matched tables. Performance scales with model size for both. On fully linked data, KG embeddings match LLMs of similar size.

When linking is solved, KGs match LLMs Crucially, on these fully linked tables, the best KG embedding models perform on par with LLMs of a similar parameter count (Figure 2). This suggests that the primary advantage of LLMs in the general setting is not that they encode superior knowledge, but that they implicitly solve the difficult "symbol grounding" problem of linking messy, real-world text to canonical entities. When this problem is removed, the structured knowledge from KGs is just as effective. This also implies that for these tasks, the advanced language understanding capabilities of LLMs beyond entity recognition provide little additional benefit.

5 Conclusion

Our large-scale study demonstrates that for tabular learning with text, the quality of text representations is crucial. Knowledge-rich embeddings from LLMs or KGs bring more gains than complex downstream models. Thus, the focus should shift from the learning algorithm to the quality of the input features. When entity linking is provided, pure KG models are just as powerful as LLMs of similar size, suggesting that the main advantage of LLMs for tabular data is not superior knowledge, but their ability to bridge the gap between unstructured text and canonical entities. The clear benefit of scale points to a crucial direction for future research: next-generation tabular foundation models should be built upon *large* language models to leverage their powerful entity linking and knowledge encoding capabilities. Tapping into massive knowledge bases like the full Wikidata could unlock significant performance gains and lead to more powerful and versatile tabular learning systems.

References

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Alexis Cvetkov-Iliev, Alexandre Allauzen, and Gaël Varoquaux. Relational data embeddings for feature enrichment with background information. *Machine Learning*, 112(2):687–720, 2023.

Léo Grinsztajn, Edouard Oyallon, Myung Jun Kim, and Gaël Varoquaux. Vectorizing string entries for data processing on tables: when are larger language models better? *arXiv preprint arXiv:2312.09634*, 2023.

Félix Lefebvre and Gaël Varoquaux. Scalable feature learning on huge knowledge graphs for downstream machine learning. *Advances in Neural Information Processing Systems*, 38, 2025.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

Fabian M Suchanek, Mehwish Alam, Thomas Bonald, Lihu Chen, Pierre-Henri Paris, and Jules Soria. Yago 4.5: A large and clean knowledge base with a rich taxonomy. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 131–140, 2024.

Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8, 2011.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM computing surveys*, 55(12):1–38, 2023.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Hamidreza Kamkari, Alex Labach, Jesse C Cresswell, Keyvan Golestan, Guangwei Yu, Maksims Volkovs, and Anthony L Caterini. Tabdpt: Scaling tabular foundation models. *arXiv preprint arXiv:2410.18164*, 2024.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data. *arXiv preprint arXiv:2506.16791*, 2025.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. Tabllm: Few-shot classification of tabular data with large language models. In *International conference on artificial intelligence and statistics*, pages 5549–5581. PMLR, 2023.

Josh Gardner, Juan C Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling. *Advances in Neural Information Processing Systems*, 37:45155–45205, 2024.

Myung Jun Kim, Leo Grinsztajn, and Gael Varoquaux. Carte: Pretraining and transfer for tabular learning. In *ICML*, 2024.

Myung Jun Kim, Félix Lefebvre, Gaëtan Brison, Alexandre Perez-Lebel, and Gaël Varoquaux. Table foundation models: on knowledge pre-training for tabular learning. *arXiv preprint arXiv:2505.14415*, 2025.

Gjergji Kasneci and Enkelejda Kasneci. Enriching tabular data with contextual llm embeddings: A comprehensive ablation study for ensemble classifiers. *arXiv preprint arXiv:2411.01645*, 2024.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*, 2014.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*, 2019.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.

Apoorv Saxena, Adrian Kochsiek, and Rainer Gemulla. Sequence-to-sequence knowledge graph completion and question answering. *arXiv preprint arXiv:2203.10321*, 2022.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. Ernie 2.0: A continual pre-training framework for language understanding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8968–8975, 2020.

Shangbin Feng, Weijia Shi, Yuyang Bai, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Knowledge card: Filling llms' knowledge gaps with plug-in specialized language models. *arXiv preprint arXiv:2305.09955*, 2023.

Martin Mráz, Breenda Das, Anshul Gupta, Lennart Purucker, and Frank Hutter. Towards benchmarking foundation models for tabular data with text. *arXiv preprint arXiv:2507.07829*, 2025.

Liane Vogel, Jan-Micha Bodensohn, and Carsten Binnig. Wikidbs: A large-scale corpus of relational databases from wikidata. *Advances in Neural Information Processing Systems*, 37:41186–41201, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: a python library for training and evaluating knowledge graph embeddings. *Journal of Machine Learning Research*, 22(82):1–6, 2021.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

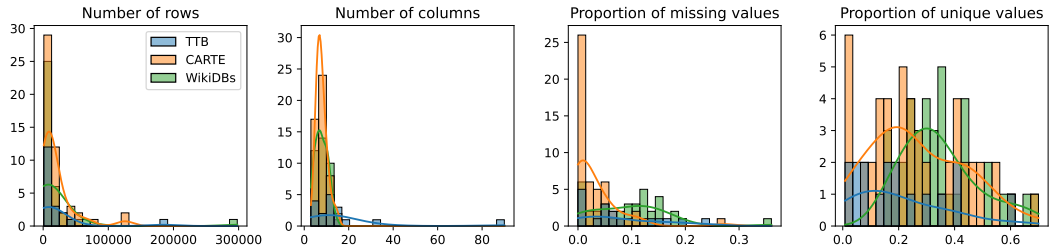

Figure 3: Statistics distribution across sources.

A More details on the experiments

Dataset selection and properties TextTabBench and CARTE are established benchmarks for tabular learning, providing real-world tables with varied text features, from short entity names to longer descriptions. Each table is associated with a predefined prediction task (regression, binary, or multi-class classification). WikiDBs is a large corpus of over 1.6 million semi-synthetic tables generated from Wikidata. To create meaningful tasks from this source, we first filtered for tables with at least 1,200 rows, then manually curated a subset of 37 tables for which we could define a relevant prediction problem. Table 1 summarizes the final distribution of tasks across the three sources, and Table 6 gives general properties of the tables. Further details on each individual dataset are provided in Table 7, Table 8, and Table 9.

Figure 3 gives statistics about table sizes, proportion of missing values, and mean column cardinality.

Table 3: Task distribution across sources, for linked tables.

Source	b-clf	m-clf	reg	Total
CARTE	0	0	3	**3**
WikiDBs	1	3	8	**12**
Total	**1**	**3**	**11**	**15**

Table 4: Embedding dimensions for the different baseline models.

Model	Dimension
TF-IDF + SVD	30 per column
FastText	300
TARTE	768
Llama-3.2-1B	2048
Llama-3.2-3B	3072
Llama-3.1-8B	4096
TabuLa-8B	4096
Qwen3-0.6B	1024
Qwen3-4B	2560
Qwen3-8B	4096
RoBERTa (base, large)	768, 1024
ERNIE 2.0 (base, large)	768, 1024
e5-v2 (small, base)	384, 768
T5 (small, base)	512, 768
KGT5 (small, base)	512, 768
OPT-1.3B	2048
Knowledge card	2048

Data preprocessing We adopt the original preprocessing from TextTabBench and CARTE. For WikiDBs, we apply a procedure similar to TextTabBench. We also ensure that multi-class classification tasks have at most 10 classes, each with at least 105 samples. For all 105 datasets, we then apply the following preprocessing pipeline: (1) we remove all numerical columns to focus our study on text-based knowledge; (2) we log-transform regression targets with wide-ranging distributions; (3) we downsample majority classes in multi-class problems to create balanced datasets; and (4) we discard any table with fewer than 1,050 rows post-processing to ensure sufficient data for evaluation. We also exclude one dataset from TextTabBench with excessively long text entries that exceed the context limits of some of our baselines.

Experiments on linked tables We have 15 linked tables, 4 for classification and 11 for regression. Details on these tables are provided in Table 3.

For the KG embedding models (DistMult, TransE, ComplEx and RotatE), we use $d = 300$ for the embedding dimension, and train them for 100 epochs with a batch size of 8192 and a learning rate of 10^{-3}, and use the default parameters of their PyKEEN implementation [Ali et al., 2021].

For KGs smaller than Wikidata5M (see Table 2), some rows of the linked tables are not matched to the KG. In that case, after embedding the rows corresponding to matched entities, we impute missing values using the mean along each column. If no row at all is matched in a table, we simply replace the missing values with zeros.

The number of parameters reported in Figure 2 for KG models is the number of entities in the KG multiplied by the embedding dimension d.

Extracting embeddings from LLMs We generated sentence-level embeddings from the serialized rows using the `SentenceTransformer` framework [Reimers and Gurevych, 2019], which provides a unified interface for a wide range of transformer-based models. We used it to extract representations from the following models: Llama 3 [Dubey et al., 2024], Qwen3 [Zhang et al., 2025], RoBERTa [Liu et al., 2019], T5 [Raffel et al., 2020], e5-v2 [Wang et al., 2022], OPT [Zhang et al., 2022], TabuLa [Gardner et al., 2024], ERNIE 2.0 [Sun et al., 2020], Knowledge Card [Feng et al., 2023], and KGT5 [Saxena et al., 2022] families, using pretrained checkpoints available on the Hugging Face Hub [Wolf et al., 2020]. For TARTE [Kim et al., 2025] and FastText [Bojanowski et al., 2017], we directly use the output embeddings of the model.

Table serialization To generate embeddings from LLMs, we serialize each table row into a natural language prompt. Following Gardner et al. [2024], we use the format: "The <col_a> is <val_a>. The <col_b> is <val_b>. What is the value of <target>?". For KGT5, we adapt the prompt to better match its pretraining format: "<col_a> | <val_a>. <col_b> | <val_b>. Predict: <target>". Constructing the embeddings across multiple columns (as opposed to the study of Grinsztajn et al. [2023]) is important because it enables the context (column name, other entries on the same row) to inform the representation, e.g. leading to disambiguate "Cambridge; UK" from "Cambridge; Massachusetts" in a table with columns "city; country".

Embedding dimensions Table 4 reports the embedding dimensions for the different baseline models used.

Table 5: Search space for XGBoost hyperparameters.

Hyperparameter	Distribution	Range
n_estimators	Integer	[50, 1000]
max_depth	Integer	[2, 6]
min_child_weight	Log-uniform	[1, 100]
subsample	Uniform	[0.5, 1.0]
learning_rate	Log-uniform	$[10^{-5}, 1]$
colsample_bylevel	Uniform	[0.5, 1.0]
colsample_bytree	Uniform	[0.5, 1.0]
gamma	Log-uniform	$[10^{-8}, 7]$
reg_lambda	Log-uniform	[1, 4]
alpha	Log-uniform	$[10^{-8}, 100]$

Table 6: Aggregated features of tabular datasets across sources. The cardinality is computed on 1,024 rows.

	TextTabBench	CARTE	WikiDBs
# columns	15.65	6.76	6.73
cardinality	286.36	371.44	463.70
string length	975.29	298.80	203.62
string similarity[1]	0.16	0.10	0.08

[1] cosine similarity of TF-IDF across rows

Metrics and score normalization We evaluate performance using the R2 score for regression and the ROC-AUC score for classification. To aggregate results across datasets of varying difficulty, we normalize scores for each dataset and random seed. Following Grinsztajn et al. [2022], we establish a normalized scale where the best-performing model scores 1 and the model at the 10th performance percentile scores 0. Other models' scores are mapped to this [0, 1] range via an affine transformation. For regression, we clip scores at 0 to mitigate the impact of poor-performing outliers.

Uncertainty estimation To account for statistical variability, we repeat each experiment 10 times with different random seeds. The error bars in our result figures represent the standard error of the mean across these runs.

XGBoost hyperparameter tuning For the XGBoost estimator, we perform hyperparameter optimization via a randomized search with 100 iterations. We use 5-fold cross-validation, repeated 5 times on the training set, to evaluate each hyperparameter configuration. The detailed search space is provided in Table 5.

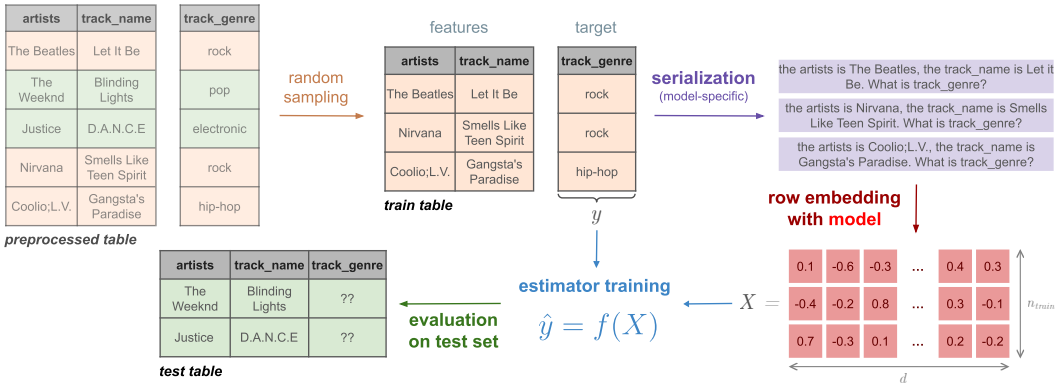

Figure 4: An overview of our evaluation pipeline. For each dataset, we sample training and test sets. We then serialize the rows and use the embedding **model** to generate a vector representation for each row. Finally, we train a tabular learning estimator to evaluate these embeddings.

Overview of the evaluation pipeline Figure 4 summarizes our evaluation pipeline.

B Additional results

B.1 Runtime analysis

The benefits of leveraging external knowledge come at a computational cost. Table 10 details the average runtimes for embedding generation and estimator fitting (ridge) across different embedding models and training sizes. As expected, larger models introduce a significant computational overhead. For instance, generating embeddings with an 8-billion-parameter LLM is, on average, over 100 times slower than using the non-pretrained baseline. This highlights the trade-off between predictive performance and the computational resources required for knowledge integration.

Table 7: Overview of TextTabBench datasets used in our benchmark. Table statistics after preprocessing.

Dataset	Task	# rows	# columns	# classes	# linked rows
Diabetes	b-clf	17,000	5	2	-
Job Frauds	b-clf	1,732	12	2	-
Kickstarter	b-clf	18,720	10	2	-
Lending Club	b-clf	11,254	13	2	-
Osha Accidents	b-clf	3,598	16	2	-
Customer Complaints	m-clf	1,384	9	4	-
Spotify	m-clf	10,000	4	10	-
Airbnb	reg	3,818	33	-	-
Beer	reg	2,914	6	-	-
California Houses	reg	11,349	14	-	-
Covid Trials	reg	1,165	14	-	-
Insurance Complaints	reg	37,484	9	-	-
IT Salary	reg	1,253	17	-	-
Mercari	reg	12,000	5	-	-
San Francisco Permits	reg	183,794	13	-	-
Stack Overflow	reg	19,427	90	-	-
Wine	reg	1,281	13	-	-

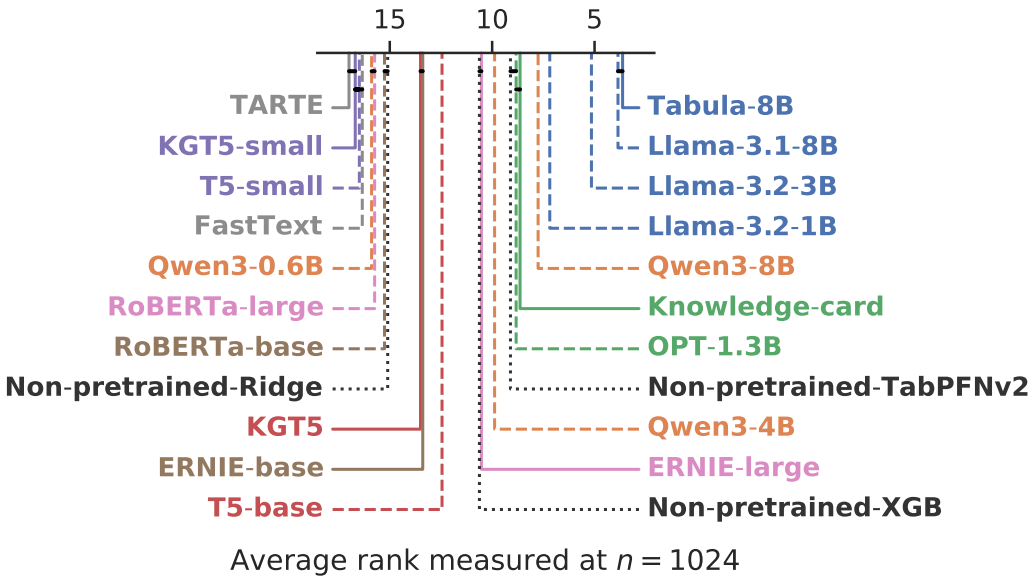

Figure 5: Critical difference diagram across all data sources and methods.

B.2 Overall model ranking

Figure 5 presents a critical difference diagram comparing the mean ranks of all embedding methods when paired with a Ridge predictor. It also includes the performance of more advanced estimators on non-pretrained representations for context.

B.3 Performance analysis by data source

Figure 6 illustrates the relative improvements of knowledge-rich representations over non-pretrained ones, broken down by data source. The benefits of external knowledge vary with dataset characteristics; tables from WikiDBs and CARTE, which are more knowledge-intensive, gain more from these representations than those from TextTabBench.

Table 6 also shows that TextTabBench table entries are less varied than the ones of CARTE and WikiDBs. This probably adds regularities that non-pretrained features combined with TabPFNv2 can leverage, explaining their better performance on these datasets.

Table 8: Overview of CARTE datasets used in our benchmark. Table statistics after preprocessing.

Dataset	Task	# rows	# columns	# classes	# linked rows
Chocolate Bar Ratings	b-clf	2,218	7	2	-
Coffee Ratings	b-clf	1,670	9	2	-
Michelin	b-clf	6,774	6	2	-
NBA Draft	b-clf	1,550	5	2	-
Ramen Ratings	b-clf	3,726	5	2	-
Roger Ebert	b-clf	2,668	6	2	-
Spotify	b-clf	41,096	8	2	-
US Accidents Severity	b-clf	20,930	10	2	-
Whisky	b-clf	1,788	7	2	-
Yelp	b-clf	60,088	9	2	-
Zomato	b-clf	60,302	8	2	-
Movies	reg	7,224	8	-	7,095
US Accidents Counts	reg	22,623	7	-	14,697
US Presidential	reg	19,857	7	-	13,221
Anime Planet	reg	14,391	7	-	-
Babies R Us	reg	5,085	5	-	-
Beer Ratings	reg	3,197	6	-	-
Bikedekho	reg	4,786	6	-	-
Bikewale	reg	8,992	6	-	-
Buy Buy Baby	reg	10,718	5	-	-
Cardekho	reg	37,813	14	-	-
Clear Corpus	reg	4,724	11	-	-
Company Employees	reg	10,941	8	-	-
Employee Remuneration	reg	35,396	3	-	-
Employee Salaries	reg	9,211	7	-	-
Fifa22 Players	reg	18,085	10	-	-
Filmtv Movies	reg	41,205	7	-	-
Journal JCR	reg	9,615	5	-	-
Journal SJR	reg	27,931	10	-	-
Japanese anime	reg	15,535	12	-	-
K-Drama	reg	1,239	9	-	-
ML/DS salaries	reg	10,456	8	-	-
Museums	reg	11,467	15	-	-
Mydramalist	reg	3,400	11	-	-
Prescription Drugs	reg	1,714	6	-	-
Rotten Tomatoes	reg	7,158	11	-	-
Used Cars 24	reg	5,918	7	-	-
Used Cars Benz Italy	reg	16,391	6	-	-
UsedCars.com	reg	4,009	9	-	-
Used Cars Pakistan	reg	72,655	5	-	-
Used Cars Saudi Arabia	reg	5,507	8	-	-
Videogame Sales	reg	16,410	5	-	-
Wikiliq Beer	reg	13,461	8	-	-
Wikiliq Spirit	reg	12,275	6	-	-
Wina Poland	reg	2,247	13	-	-
Wine.com Prices	reg	15,254	7	-	-
Wine.com Ratings	reg	4,095	7	-	-
WineEnthusiasts Prices	reg	120,975	9	-	-
WineEnthusiasts Ratings	reg	129,971	9	-	-
WineVivino Price	reg	13,834	6	-	-
WineVivino Rating	reg	13,834	7	-	-

Table 9: Overview of WikiDBs datasets used in our benchmark. Table statistics after preprocessing.

Dataset	**Task**	**# rows**	**# columns**	**# classes**	**# linked rows**
CC Authors	b-clf	16,224	8	2	1,302
Defenders	m-clf	18,610	11	10	8,700
Philosophers	m-clf	4,230	9	10	1,656
US Music Albums	m-clf	3,270	11	10	2,180
Artist Copyrights	m-clf	2,000	10	10	-
Artworks Catalog	m-clf	1,210	9	10	-
Forward Players	m-clf	1,400	11	10	-
Geographers	m-clf	1,130	10	10	-
Historic Buildings	m-clf	27,980	7	10	-
Islands	m-clf	19,650	4	10	-
Kindergarten Locations	m-clf	2,790	7	3	-
Magic Narratives	m-clf	1,062	5	9	-
Museums	m-clf	9,550	5	10	-
Noble Individuals	m-clf	1,400	10	10	-
Notable Trees	m-clf	1,408	5	8	-
Parish Churches	m-clf	1,350	5	10	-
Sculptures	m-clf	3,720	7	10	-
Spring Locations	m-clf	5,930	3	10	-
State Schools	m-clf	2,800	4	10	-
Scientific Articles	m-clf	2,760	14	10	-
Sub Post Offices	m-clf	1,530	4	10	-
Transport Stations	m-clf	4,640	9	10	-
Business Locations	reg	16,821	5	-	16,438
Dissolved Municipalities	reg	13,462	7	-	1,656
Geopolitical Regions	reg	1,114	7	-	1,066
Historical Figures	reg	11,260	12	-	2,134
Municipal District Capitals	reg	1,658	6	-	1,267
Poets	reg	60,240	11	-	21,564
Territorial Entities	reg	36,717	8	-	34,189
WWI Personnel	reg	30,675	12	-	16,227
Artworks Inventory	reg	10,635	6	-	-
Drawings Catalog	reg	63,130	9	-	-
Eclipsing Binary Stars	reg	297,934	7	-	-
Registered Ships	reg	4,644	7	-	-
Research Articles	reg	6,962	7	-	-
Research Article Citations	reg	4,115	10	-	-
Ukrainian Villages	reg	21,355	4	-	-

Table 10: Average runtimes (in seconds) for embedding extraction and ridge fitting, for varying train set sizes.

	Train-size		
Models	64	256	1 024
TabuLa-8B	124 ± 141	145 ± 166	216 ± 258
Llama-3.1-8B	119 ± 133	140 ± 157	209 ± 247
Llama-3.2-3B	43 ± 51	51 ± 60	76 ± 93
Llama-3.2-1B	18 ± 20	21 ± 24	32 ± 37
Qwen3-8B	120 ± 144	140 ± 169	210 ± 262
Qwen3-4B	65 ± 82	76 ± 96	114 ± 150
Qwen3-0.6B	12 ± 13	14 ± 15	21 ± 22
Knowledge-card	25 ± 29	30 ± 34	45 ± 53
OPT-1.3B	23 ± 29	27 ± 34	40 ± 53
ERNIE-large	8 ± 6	10 ± 7	15 ± 9
RoBERTa-large	8 ± 6	10 ± 7	14 ± 9
ERNIE-base	5 ± 4	6 ± 4	8 ± 6
RoBERTa-base	4 ± 3	5 ± 3	7 ± 4
KGT5	5 ± 3	6 ± 4	8 ± 5
T5-base	5 ± 6	7 ± 7	9 ± 11
KGT5-small	3 ± 3	4 ± 3	6 ± 4
T5-small	4 ± 3	4 ± 3	6 ± 4
TARTE	4 ± 4	5 ± 5	8 ± 6
FastText	2 ± 4	3 ± 4	4 ± 6
Non-pretrained	0.5 ± 0.7	1 ± 1	2 ± 2

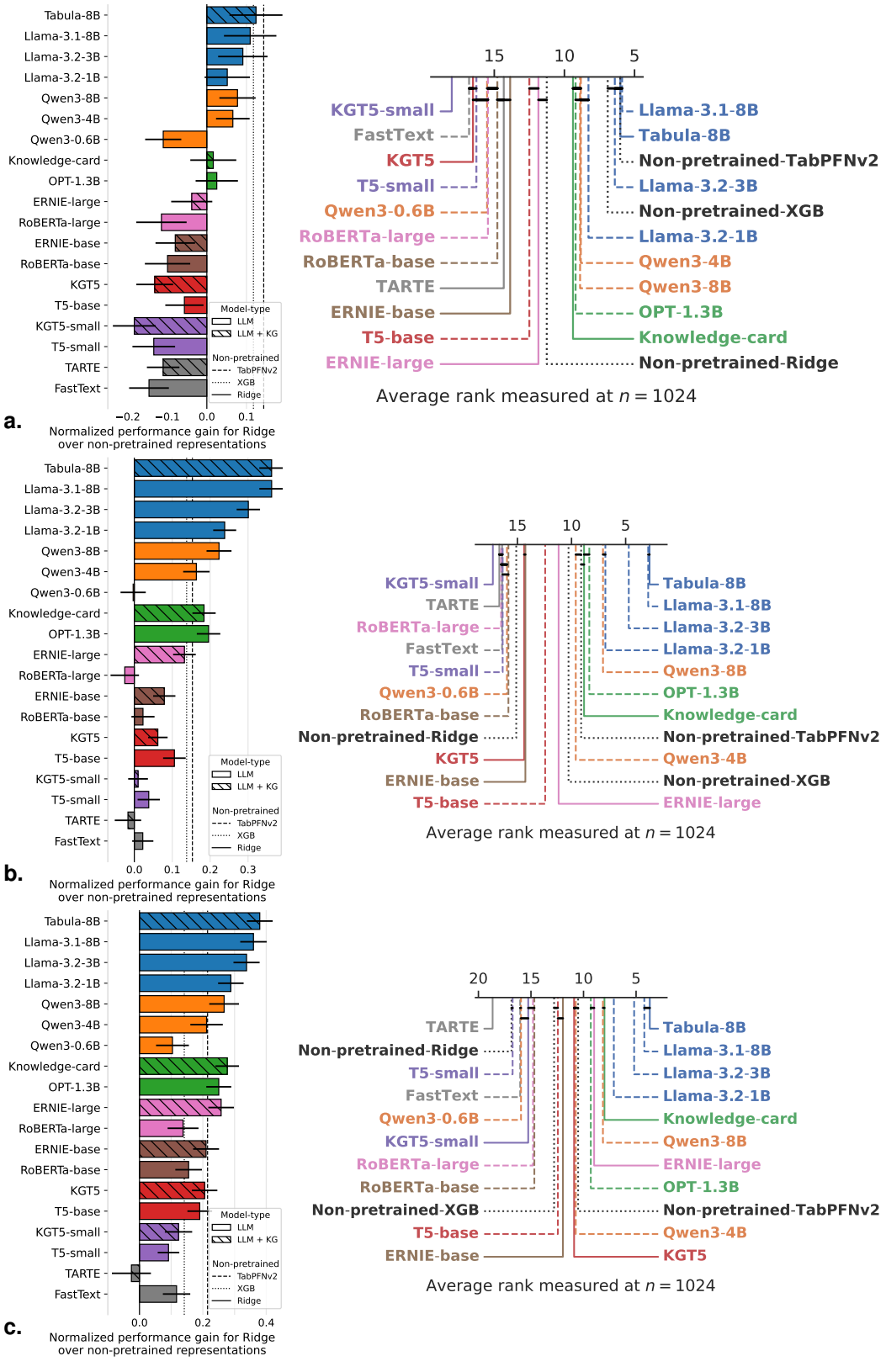

Figure 6: Relative improvements to non-pretrained string representations, when using a ridge model as a tabular learner. For each source, larger models consistently yield better performances: **a.** TextTabBench **b.** CARTE **c.** WikiDBs.