Building Specialized Software-Assistant ChatBot with Graph-Based Retrieval-Augmented Generation

Mohammed Hilel[1*] **Yannis Karmim**[1*] **Jean De Bodinat**[1] **Réda Sarehane**[2]
Antoine Gillon[2]
[1]RAKAM AI, Paris, France [2]Lemon Learning, Paris, France
[*]Equal contribution

Abstract:

Digital Adoption Platforms (DAPs) support employees in the use of complex enterprise software, but creating and maintaining in-app guidance remains largely manual. We propose an industrial Graph-based Retrieval-Augmented Generation (Graph-RAG) pipeline that automatically converts web applications into *state-action* knowledge graphs and retrieves a compact, connected subgraph relevant to a user query. The retrieved subgraph is textualized and injected into the LLM prompt to generate grounded step-by-step instructions without retraining, making it compatible with black-box APIs. We also introduce `SoftwareQA`, a benchmark of ≈1200 multiple-choice questions across 4 enterprise applications. Experiments show consistent accuracy gains across model sizes, reaching up to +16 points with graph augmentation. Code and dataset will be release.

Keywords:

LLMs ; Knowledge Graphs ; Retrieval-Augmented Generation

1. Introduction

Digital Adoption Platforms (DAPs) help employees navigate complex business software such as CRM [1], ERP [2], and HRMS [3] systems, but authoring their step-by-step guides remains largely manual [4]. LLMs [5, 6] could automate this, yet they hallucinate without structured knowledge of the target application [7], and fine-tuning is impractical due to catastrophic forgetting [8] and incompatibility with black-box APIs [9]. Retrieval-Augmented Generation (RAG) [10] currently outperforms fine-tuning for knowledge injection [11], and graph-structured retrieval further improves precision [12]. **However, standard RAG assumes the availability of well-structured and maintained textual documentation**. In enterprise settings, such documentation is sometime incomplete, outdated, or entirely unavailable. Moreover, raw document retrieval does not capture the procedural and state-dependent nature of UI interactions.

We present a Graph-RAG framework that addresses this gap. Our contributions are: (i) a **Software-to-Graph pipeline** that automatically extracts state-action graphs directly from web interfaces, removing the dependency on pre-existing documentation; (ii) a **graph retrieval** method adapted to large state-action graphs and compatible with black-box LLMs; (iii) **`SoftwareQA`**, a benchmark across four enterprise applications.

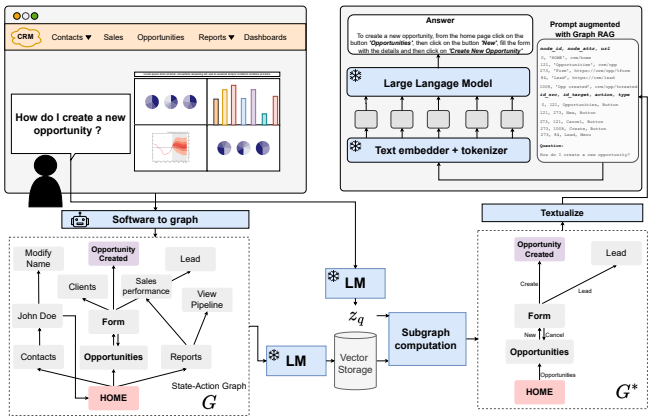

FIGURE 1. Graph-RAG overview: (1) the software is converted into a state–action graph G; (2) the user query is embedded to score graph elements; (3) a compact subgraph G^* is extracted via PCST; (4) G^* is textualized and provided as context to the LLM.

2 Method

Software-to-Graph. We define a knowledge graph $G = (V, E)$ where nodes $u \in V$ are interface states and edges $e \in E$ are user actions with textual attributes x_u, x_e. A Playwright-based bot crawls the web application via breadth-first traversal, detecting clickable elements through DOM inspection. The pipeline requires only login credentials and a home page URL, is domain-agnostic, and produces graphs updated incrementally as the software evolves. Table 1 reports graph statistics for our four applications.

TABLE 1. Extracted knowledge graphs statistics.

SOFTWARE	TYPE	#NODES	#EDGES	TIME(S)
SALESFORCE	CRM	7640	7655	5217
DOLIBARR	ERP	1714	2196	9652
PIPEDRIVE	CRM	120	694	670
IVALUA	ERP	450	464	449

Graph-based retrieval. Since full graphs are too large for an LLM prompt, we apply the Prize-Collecting Steiner Tree (PCST) algorithm [13] to extract a minimal connected subgraph G^*. Node and edge descriptions are encoded with a multilingual Sentence-BERT [14] (`MiniLM-L12-v2`); cosine similarity with the query selects top-k candidates. PCST then maximizes cumulative relevance prizes while minimizing traversal cost, enforcing a single connected component (`pcst_fast`, `num_cluster=1`). The user's current UI state is pinned with similarity score 1, anchoring the subgraph to the user's context.

Prompt construction. The textualized subgraph is concatenated with the user query and fed to a frozen LLM. Listing edges as `(src, tgt, action)` enables multi-hop reasoning [15] and reduces hallucinations [12]. Unlike G-Retriever, we omit GNN and MLP projection layers, making our approach directly compatible with any black-box LLM API without retraining.

3 Experiments

`SoftwareQA`. Questions and distractors were generated from official documentation using `gpt-oss-120b` in a three-pass pipeline (question generation → distractor generation → distractor filtering). This design also validates that our KG covers the official documentation, a useful property when docs are sparse or outdated. The benchmark covers three skills: *navigation*, *configuration*, and *concept* understanding, with ≈300 questions per application (Table 3).

TABLE 2. MCQ example from SOFTWAREQA.

Field	Content
ID / Type	285 / Configuration
Question	Navigation path to configure VAT accounts in Dolibarr?
A (✓)	Accounting → Configuration → VAT → Add rate
B	Home → Settings → Modules → Enable VAT
C	Third Parties → Configuration → Taxes → Create rate
D	Bank → Accounts → VAT tab → Add rate

TABLE 3. `SoftwareQA` dataset statistics.

Software	Nav.	Config.	Concept	Total
Salesforce	191	103	6	300
Dolibarr	134	159	7	300
Pipedrive	244	27	28	299
Ivalua	77	46	174	297

Results. Accuracy is measured as the fraction of correctly selected choices, making results fully reproducible. As shown in Table 4, Graph-RAG improves every model on every dataset (+0.2 to +16.4 pts). Smaller models benefit the most: Ministral-Small +GRAG reaches the unaugmented Qwen3-480B level, showing graph retrieval can substitute for model scale. Gains are highest on Ivalua, whose domain-specific procurement vocabulary is harder for LLMs to internalize.

TABLE 4. Accuracy (%). **SF**: Salesforce, **DB**: Dolibarr, **PD**: Pipedrive, **IV**: Ivalua.

Model	SF	DB	PD	IV
Small				
Ministral-S	75.6	79.0	94.9	77.9
Ministral-S +GRAG	**83.0**	**83.3**	**95.1**	**94.3**
Medium				
Mistral-M	79.7	78.0	91.6	76.3
Mistral-M +GRAG	**87.0**	**84.3**	**94.6**	**95.3**
Large				
Qwen3-480B	79.3	83.7	96.0	81.9
Qwen3-480B +GRAG	**87.0**	**86.3**	**96.3**	**94.3**

4 Conclusion and industrial impact

We presented a Graph-RAG framework that converts enterprise web applications into state–action knowledge graphs to ground LLM-generated software assistance. Compatible with black-box APIs and requiring no fine-tuning, it improves accuracy on `SoftwareQA`. Future work includes more robust graph extraction and user-level personalization. Our approach reduces manual authoring and maintenance effort in Digital Adoption Platforms, shortens adaptation cycles when enterprise software evolves, and lowers operational costs by scaling guidance generation across applications. Beyond assistance, grounding LLMs in executable UI knowledge also opens the path toward LLM-driven UI agents for partial enterprise workflow automation.

References

[1] Nutshell, "Crm stats: 8 stats that will influence your crm strategy in 2023," 2023, accessed: 2025-10-26. [Online]. Available: https://www.nutshell.com/blog/crm-stats#:~:text=8,%28source%3A%20Salesforce

[2] NetSuite, "Erp benefits: The key benefits of erp systems for your business," 2023, accessed: 2025-10-26. [Online]. Available: https://www.netsuite.com/portal/resource/articles/erp/erp-benefits.shtml#:~:text=There%20are%20countless%20ways%20an,on%20how%20certain%20processes%20work

[3] HRMorning, "What is an ats? everything you need to know about applicant tracking systems," 2023, accessed: 2025-10-26. [Online]. Available: https://www.hrmorning.com/articles/ats/#:~:text=Most%20ATS%20let%20HR%20pros,information%20all%20in%20one%20space

[4] G. Latka, "Lemon learning: Company profile and revenue data," 2024, accessed: 2025-10-26. [Online]. Available: https://getlatka.com/companies/lemon-learning#:~:text=In%202024%2C%20Lemon%20Learning%E2%80%99s%20revenue,increasing%20adoption%20across%20various%20industries

[5] B. et. al, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[6] Meta AI, "Introducing meta llama 3: The most capable openly available llm to date," 2024. [Online]. Available: https://ai.meta.com/blog/meta-llama-3

[7] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin *et al.*, "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions," *ACM Transactions on Information Systems*, vol. 43, no. 2, pp. 1–55, 2025.

[8] Z. Gekhman, G. Yona, R. Aharoni, M. Eyal, A. Feder, R. Reichart, and J. Herzig, "Does Fine-Tuning LLMs on New Knowledge Encourage Hallucinations?" in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 7765–7784. [Online]. Available: https://aclanthology.org/2024.emnlp-main.444/

[9] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[10] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. [Online]. Available: https://arxiv.org/abs/2005.11401

[11] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 237–250. [Online]. Available: https://aclanthology.org/2024.emnlp-main.15/

[12] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, and Z. Li, "Retrieval-augmented generation with knowledge graphs for customer service question answering," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR 2024. ACM, Jul. 2024, p. 2905–2909. [Online]. Available: http://dx.doi.org/10.1145/3626772.3661370

[13] D. Bienstock, M. X. Goemans, D. Simchi-Levi, and D. Williamson, "A note on the prize collecting traveling salesman problem," *Mathematical programming*, vol. 59, no. 1, pp. 413–420, 1993.

[14] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Conference on Empirical Methods in Natural Language Processing*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:201646309

[15] Y. Zhang, H. Wang, S. Feng, Z. Tan, X. Han, T. He, and Y. Tsvetkov, "Can llm graph reasoning generalize beyond pattern memorization?" *arXiv preprint arXiv:2406.15992*, 2024.