Enhancing Public Contract Code analysis with Graph Retrieval-Augmented Generation

Eleonora Ghizzota[1,*,†], Lucia Siciliani[1,†], Pierpaolo Basile[1,†] and Giovanni Semeraro[1,†]

[1]Department of Computer Science, University of Bari Aldo Moro, via Edoardo Orabona 4, 70125, Bari, Italy

Abstract

The rapid progression of Generative Artificial Intelligence, together with researchers' increasing attention to the Public Administration (PA), opens up to novel cross-domain applications. Generative AI technologies like Large Language Models (LLMs) can expedite administrative purchasing processes and boost the transparency of the procurement life cycle. However, in a dynamic domain like the PA, updating LLMs training data can prove very prohibitive. Recently emerged Graph Retrieval-Augmented Generation (RAG) solves this data editing limitation, and tackles the lack of global information of traditional RAG techniques. Graph RAG leverages structural information across entities, enabling more comprehensive, context-aware responses.

This paper illustrates a preliminary application of Microsoft's GraphRAG in the PA domain, leveraging the latest Italian Public Contract Code corpus version. The experimental setting consists of an interface to let PA domain experts query the model about the Public Contract Code and evaluate the answers' correctness, completeness and fluency. Then, users filled out a satisfaction questionnaire to assess system usability and users' resistance to integrating this tool into their workflow.

Results reveal a general users' satisfaction with the system: it achieves a System Usability Score of 82.19 and a Net Promoter Score of 25. Questions for assessing the correctness, completeness and fluency of the answers to users' queries achieve a mean score above 3.70. Finally, results of the survey for assessing the users' resistance – measured in terms of Perceived Value, Switching Benefit, Switching Cost, and Self-efficacy For Change – make clear that users consider this tool beneficial to their way of working.

Keywords

Public Administration, E-Procurement, Graph Retrieval-Augmented Generation, Knowledge Graphs, Large Language Models

1. Background and Motivations

The rapid surge of Generative AI has pervaded numerous domains, thanks to its wide range of applications. Among those domains, the Public Administration (PA) may greatly benefit from integrating Generative AI into the workflow. Even before the surge of Generative AI, researchers have shown a growing interest in Public E-procurement[1] [1, 2, 3, 4, 5, 6]. The goal of Public E-procurement is to automate a public procurement procedure to purchase goods, works or services. Such technologies boost and expedite administrative purchasing processes, expanding market player participation and preserving transparency of the procurement life cycle, hence clarifying and guaranteeing the accuracy of the necessary controls.

With the rise of Generative AI technologies – e.g., Large Language Models (LLMs) – this goal is becoming more achievable in a domain that heavily relies on textual documents. The integration of an LLM might be convenient for assisting both PA professionals and citizens in the decision-making and information access processes. For instance, the assistance of LLMs in the administrative environment may ease the task of retrieving in a single answer a piece of information that requires analysing a

LLM-TEXT2KG 2025: 4th International Workshop on LLM-Integrated Knowledge Graph Generation from Text, June 2, Portoroz, Slovenia

*Corresponding author.

†These authors contributed equally.

e.ghizzota@phd.uniba.it (E. Ghizzota); lucia.siciliani@uniba.it (L. Siciliani); pierpaolo.basile@uniba.it (P. Basile); giovanni.semeraro@uniba.it (G. Semeraro)

0000-0002-0751-3891 (E. Ghizzota); 0000-0001-7116-9338 (L. Siciliani); 0000-0002-0545-1105 (P. Basile); 0000-0002-9421-8566 (G. Semeraro)

[1]www.agid.gov.it/en/intervention-areas/e-procurement

number sources [2, 3, 4]; because of the technical language many documents employ, this task gets even more complex. The assistance of an LLM would make this kind of task more accessible and less time-consuming by performing the hard work of collecting and understanding on behalf of the users.

Despite their significant contribution, information encoded in LLMs is limited to the data they have been trained on, and this information can become obsolete [7, 8]. In a constantly changing domain like PA, where regulations, directives and guidelines are subject to frequent updates and corrections, these data-wise limitations are much more evident, and maintaining data pertinence can be restrictive.

To address these limitations, the introduction of non-parametric memorisation techniques – e.g., Retrieval-Augmented Generation (RAG) [9], Adaptive Retrieval [10], Graph Retrieval-Augmented Generation [11, 12] – has been proposed. Non-parametric knowledge enhances the output of an LLM, that "consults" trustworthy knowledge sources for collecting fresh data that are not in its original training dataset. The LLM leverages this new knowledge and its training data to generate better responses, therefore specialising in an specific field in a cost-effective manner, without re-training.

However, traditional RAG techniques still face several limitations in real-world scenarios [13]. Their semantic similarity approach is not suitable for capturing the textual interconnections and relational knowledge. Excessively lengthy context in prompts can degrade the performance: [14] noticed that a better performance is achieved when relevant information occurs at the beginning or end of the input context, but worsens when relevant information is in the middle of long contexts. Lastly, vector RAG techniques cannot grasp global information, so they cannot adequately perform Query-Focused Summarisation (QFS) tasks, that are *sense-making* queries requiring a global comprehension of the data [11] – e.g., "*What are the key trends in how scientific discoveries are influenced by interdisciplinary research over the past decade?*" – rather than the retrieval of a specific piece of information. Sense-making tasks require reasoning over "*connections [...] to anticipate their trajectories and act effectively*" [15]. Numerous LLMs – e.g., GPT [16], Qwen2 [17], Llama [18], Gemini [19] – have shown great capabilities in sense-making tasks; nevertheless, when RAG is required, traditional vector RAG approaches cannot manage an entire corpus. Graph Retrieval-Augmented Generation tackles the issue integrating RAG with graph data like Knowledge Graphs (KGs) [20]. Information organised in graphs enables RAG to leverage the interconnections between multiple texts, and to take advantage of the abstraction and summarisation of textual data.

This paper illustrates a preliminary experiment with Graph Retrieval-Augmented Generation in the Public Administration domain, leveraging the corpus of the Italian Public Contract Code. Section 2 describes the data and methodology used, and Section 3 the experimental setting and results. Finally, Section 4 lays out the conclusions and presents some future works.

2. System

2.1. Corpus

The corpus leveraged for this experiment is the *Italian Public Contract Code*, last updated with the Legislative Decree 36/2023[2].

The Italian Public Contract Code, or Tender Code, is a law issued by the Italian Republic to regulate public tenders and administrative concessions. It was instituted in 1924, and has undergone several changes since 1994. This regulatory text describes the procedures through which the Public Administration acquires goods and services, awards contracts, and grants concessions. When the public sector needs to meet its procurement requirements, it must act under the rules of public procurement, a fundamental principle for selecting the contractor. Therefore, the phases leading to the selection of the contractor are determined by administrative law under the jurisdiction of the administrative judge, while the contract signed with the contractor is ruled by civil law under the jurisdiction of the civil judge.

[2]Available in Italian here https://www.bosettiegatti.eu/info/norme/statali/2023_0036.htm, in English here https://www.codiceappalti.it/Home/Legge/?legge=Italian_Procurement_Code_Decree_36/2023.

The Italian Public Contract Code consists of 229 articles, divided into 5 volumes, plus 37 attachments:

I. General principles and provisions regarding the digitalization of public contracts, their planning, and design;
II. Contracts for works, services, and supplies. It provides information on the relevant parties, particularly the contracting authorities on one side and economic operators on the other;
III. Procurement in special sectors;
IV. Public-private partnerships and concessions;
V. Dispute management, the National Anti-Corruption Authority[3], and includes final and transitional provisions.

With respect to former versions, the current Legislative Decree 36/2023 issued on March 31st 2023 systematises numerous reform requests and amending decrees, in order to speed up procedures and address the needs arising from the COVID-19 pandemic. Furthermore, driven by projects related to the National Recovery and Resilience Plan[4], it fosters transparency, digitalization of procedures, and their dematerialization and full traceability.

The Public Contract Code embodies the primary regulatory text a PA operator refers to when dealing with bid procedures, since it regulates their entire life-cycle, from the preparatory phase, to the participation requirements, until the definitive adjudication and the contract stipulation. Therefore, it is a fitting corpus for the preliminary experiment we intend to carry out, involving professionals from the PA domain.

2.2. Methodology

GraphRAG[5] [11] is a graph-based RAG strategy for enabling *sense-making* over an entire text corpus. The GraphRAG pipeline consists of three main phases: *(i)* extraction, *(ii)* clustering, and *(iii)* query.

Extraction. The input corpus – i.e., the Legislative Decree corpus (2.1) – is split into customisable text units – e.g., paragraphs or sentences – so that an LLM can extract entities, relations and claims. GraphRAG default models are **OpenAI GPT-4o-mini** as LLM and for **OpenAI text-embedding-3-small** to produce text embeddings. GraphRAG has several default prompts the LLM must be prompted with for the extraction, but they may not fit domain-specific corpora. Therefore, GraphRAG let users to automatically or manually tune the prompts. The *auto prompt tuning*[6] functionality provided by GraphRAG uses input data and LLM interactions to create domain adapted prompts for the generation of the knowledge graph. For a minimal prompt auto tuning, we specify only the *domain* and *language* parameters, "public administration e-procurement" and "Italian", respectively. These adaptations made the persona and task descriptions, and examples for few-shot prompting more domain-specific. For instance, below is the default prompt of the persona description for summarisation:

> *You are a helpful assistant responsible for generating a comprehensive summary of the data provided below.*

Conversely, below is the persona of the auto-tuned prompt:

> *You are an expert in public administration and e-procurement. You are skilled at analysing community structures and relationships, particularly in the context of digital procurement systems. You are adept at helping organizations understand the dynamics of their e-procurement communities, facilitating collaboration, and improving procurement processes. Using your expertise, you're asked to generate a comprehensive summary of the data provided below.*

[3] *Autorità Nazionale Anticorruzione* (ANAC)
[4] *Piano Nazionale di Ripresa e Resilienza* (PNRR)
[5] https://microsoft.github.io/graphrag/
[6] https://microsoft.github.io/graphrag/prompt_tuning/auto_prompt_tuning/

To further exemplify, while in the default prompts the entity types to extract are ORGANISATION, GEO, PERSON, the entity types in the auto-tuned prompts include PUBLIC ADMINISTRATION, CONTRACT, PROJECT, SERVICE, SUBCONTRACTOR, AUTHORITY, REGULATION.

An initial knowledge graph is created upon completion of the extraction step. The resulting graph of the Public Contract Code consists of 2,089 entities and 3,250 relations.

Clustering. Hierarchical clustering with Leiden technique [21] is performed on the knowledge graph, to detect community structures within the graph; entities in each cluster are distributed across different communities for a more detailed analysis. A *community* is a group made of densely intra-connected nodes, but sparsely inter-connected to other groups in the graph. For each community and its members, a summary is generated in a bottom-up, hierarchical manner. These summaries provide a general outlook on the data – i.e., principal entities, relations and claims in the community – and act as contextual information during the querying stage.

Query. At querying time, GRAPHRAG offers two strategies, fit to the information need: global search and local search. Considering the hierarchical nature of the community structure, queries can be answered leveraging the community summaries from different levels. Whether a particular hierarchy level in the community offers the best balance of summary detail and scope for general sense-making questions or not is still an open question. *Global* search is suitable for holistic, comprehensive queries that require reasoning over the entire data corpus and community summaries, e.g., "*What are the top five themes in the data?*". Global search implements a *map-reduce* strategy. For a given community level, the summaries are randomly shuffled and divided into chunks of fixed size. At *map* step, intermediate answers are generated in parallel, and the LLM scores in a $[0, 100]$ range how relevant to the query each of them is; answers scoring 0 are excluded. In the *reduce* step, intermediate answers are ranked according to their relevance and iteratively aggregated into a new context, until the token limit is reached. The final context is employed to generate the global answer. The quality of the global search answer is affected by the level of the community hierarchy chosen for getting community reports. Lower hierarchy levels, with their detailed reports, tend to yield more thorough responses, but also increase the time and resources for generating the response due to the quantity of reports.

Local search instead is appropriate for queries reasoning on precise entities occurring in the documents, e.g. "*What are the healing properties of chamomile?*". The local search technique locates a group of entities within the knowledge graph that are semantically linked to the user's input. These entities act as gateways into the knowledge graph, facilitating the retrieval of additional pertinent information, including associated entities, relationships, entity covariates, and community reports.

3. Evaluation

The experimental setting consists of an interface to let PA domain experts query the model about the Public Contract Code and evaluate the correctness, completeness and fluency of the answers (3.1). Then, users filled out a satisfaction questionnaire for assessing system usability and users' resistance to integrate this tool in their workflow (3.2).

This experiment involves 16 expert users with heterogeneous backgrounds. In order to have a demographic overview, users were asked questions about their age, educational qualification, profession, years of experience and IT proficiency (Q.0.1 - Q.0.5).

The *age* of users spans from 20 to 65 years old: 18.8% users are in their twenties, 31.3% in their thirties and only 6.3% in their forties; finally, 31.3% users are in their fifties, and 12.5% in their sixties.
As for their *educational qualification*, 12.5% users have a high-school diploma, 12.5% a Bachelor's degree, 50% a Master's degree, and 25% are Doctors.
As concerns users' *professional role*, 6.3% is a researcher, 12.5% is a freelancer, 12.5% is a university student, 31.3% is an employee, 37.5% is member of a professional order, e.g., lawyers, accountants, consultant, engineers.

Users' *years of experience* span from 1 to 37, with an average of 13 years. Users' average *IT proficiency*, on a 5-point Likert scale (1 - Very Low, 5 - Very High), is 3.44.

3.1. Answer evaluation

For each answer to their query, users were asked to evaluate its *correctness*, *completeness* and *fluency* on a 5-point Likert scale (1 - Strongly Disagree, 5 - Strongly Agree). Users were allowed to choose between local and global search (Sec. 2.2). A total of 73 queries was asked. Table 1 illustrates the questions and the mean, variance and standard deviation of their score.
Correctness achieves a mean score of 3.80, and the highest variance and standard deviation scores. On the other hand, completeness achieves the lowest mean score, 3.74. These results are backed up by the answers on how to improve the system (Sec. 3.2), that stress the importance of providing more specific answers and the respective article references.
Finally, as expected by an LLM, fluency obtains the highest mean score, 4.48, and the lowest variance and standard deviation scores.

Table 1
Questions for assessing the correctness, completeness and fluency of GraphRAG answers.

ID	Question	Mean	Variance	Std Dev
Q.1.1	The answer provided by the system is *correct*.	3.80	1.74	1.32
Q.1.2	The answer provided by the system is *complete*.	3.74	1.31	1.14
Q.1.3	The answer provided by the system is *fluent*.	4.48	0.50	0.71

3.2. System evaluation

According to the standard ISO 9241-11, system usability can be measured in terms of effectiveness, efficiency, and satisfaction. The **System Usability Score** (SUS) was proposed by John Brooke in 1986 [22], and it proved to be intuitive and solid over hundreds of studies. Today, the SUS is still widely used to measure the usability of websites and applications [23]. The survey consists of 10 standard questions with the 5-point Likert scale (1 - Strongly Disagree, 5 - Strongly Agree). The SUS score for each survey participant is computed as in Equation 1, and assumes values in the range $[0, 100]$. Q_{odd} and Q_{even} are the scores assigned to odd and even numbered questions in the questionnaire.
The proposed system achieves 82.19, notably above the margin of the admissible range, which guidelines [24] state to be 68.

$$SUS = \left(\left(\sum Q_{odd} - 5\right) + \left(25 - \sum Q_{even}\right)\right) * 2.5 \tag{1}$$

Table 2
Questions for the SUS questionnaire, on a 5-point Likert-scale.

ID	Question
Q.2.1	I think that I would like to use this system frequently.
Q.2.2	I found the system unnecessarily complex.
Q.2.3	I thought the system was easy to use.
Q.2.4	I think that I would need the support of a technical person to be able to use this system.
Q.2.5	I found the various functions in this system were well integrated.
Q.2.6	I thought there was too much inconsistency in this system.
Q.2.7	I would imagine that most people would learn to use this system very quickly.
Q.2.8	I found the system very cumbersome to use.
Q.2.9	I felt very confident using the system.
Q.2.10	I needed to learn a lot of things before I could get going with this system.

For further insights on the likelihood that users would recommend our system with, we compute the **Net Promoter Score** (NPS). The idea behind the NPS, proposed by Bain&Co.[7], is to divide the users into *promoters*, *passives* and *detractors* of the item, based on their answer: users providing ratings between 10 and 9 are considered promoters, between 8 and 7 are passives and finally, from 6 to 0 are detractors. NPS consists of a single question Q.3.1, "*How likely is it that you would recommend this system to a friend or colleague?*". The NPS is computed as in Equation 2 and assumes values in $[-100, +100]$. Any score above 20 is considered encouraging, whereas 50 is excellent and above 80 first-rate.
The proposed system has 50% promoters and 25% detractors, therefore the obtained NPS is 25.

$$NPS = promoters - detractors \tag{2}$$

To collect detailed feedback from the users about the system, the survey includes an open-ended question Q.4.1, "*How could we improve our tool?*". Four users answered and they all agree to *(i)* include the references to the provided information, *(ii)* better detail the answers with references to the exact articles, norms and legislative texts. To summarise, although SUS and NPS results look contrasting at first, keep in mind that the SUS questionnaire addresses the usability of the system, not how good the system is at its intended task. High variance and standard deviation of correctness and completeness in Table 1, and answers to Q.4.1 justify the obtained NPS. The NPS score is in line with the scores obtained in other experiments in automating PA tasks: [4] achieved 34.4, [2] 30.3. This highlights that users found the tool easy to understand and use, but fewer users would recommend it for tasks that require a trustworthy source of information.

Finally, we conducted a separate study to evaluate how much *resistance* users would put up when integrating the system in their workflow. To this purpose, we consider the following four constructs proposed in [25, 26]:

- *Perceived Value* (PVL). Users' overall evaluation of the costs and benefits of adopting the tool, as determined by their attitude towards the change;
- *Switching Benefit* (SWB). Users' perception of the benefit that will derive from the adoption of the new tool;
- *Switching Cost* (SWC). Users' perception of the costs and efforts required to switch or integrate the new tool;
- *Self-efficacy For Change* (SFC). Users' perception of their ability to easily adapt to the new tool.

The survey consists of 14 questions with the 7-point Likert scale (1 — Strongly Disagree, 7 — Strongly Agree). Table 3 illustrates the questions.

Table 4 illustrates the statistics of the survey related to the users' resistance, in terms of mean, variance and standard deviation. The three positive constructs – i.e., Perceived Value, Switching Benefit and Self-efficacy For Change – scored a mean value above 5, while the negative construct Switching Cost scored a mean value of 3. These values are indicators of a very positive users' feedback. This insight is supported by the low variance and standard deviation values, suggesting that most users share a positive viewpoint on the system.

For a comprehensive analysis, the Pearson correlation coefficient ρ of the mean scores obtained by each construct was computed. The results show a strong positive correlation between the Perceived Value and Switching Benefit, and a negative correlation between Switching Cost and every positive construct.

4. Conclusions and Future Works

This paper illustrates a preliminary application of Microsoft's GraphRAG on the corpus of the Italian Public Contract Code. We leverage the *auto prompt tuning* feature to obtain prompts tailored to

[7] www.bain.com/insights/introducing-the-net-promoter-system-loyalty-insights/

Table 3
Questions for the Perceived Value (PVL), Switching Benefit (SWB), Switching Cost (SWC), and Self-efficacy For Change (SFC) constructs.

ID	Question
PVL 1	Considering the time and effort I have to spend, integrating the tool into my way of working is worthwhile.
PVL 2	Considering the loss that I incur, integrating the tool into my way of working is of good value.
PVL 3	Considering the hassle that I have to experience, integrating the tool into my way of working is beneficial to me.
SWB 1	Integrating the tool into my current way of working would enhance my effectiveness on the job.
SWB 2	Integrating the tool into my current way of working would enable me to accomplish relevant tasks more quickly.
SWB 3	Integrating the tool into my current way of working would increase my productivity.
SWB 4	Integrating the tool into my current way of working would improve the quality of the work I do.
SWC 1	I have already put a lot of time and effort into mastering the current way of working.
SWC 2	It would take a lot of time and effort to integrate the tool into my current way of working.
SWC 3	Integrating the tool into my current way of working could result in unexpected hassles.
SWC 4	I would lose a lot in my work if I were to integrate the tool into my current way of working.
SFC 1	Based on my own knowledge, skills and abilities, using the tool in my everyday work activities would be easy for me.
SFC 2	I am able to integrate the tool in my current way of working without the help of others.
SFC 3	I am able to integrate the tool into my current way of working reasonably well on my own.

Table 4
Statistics related to the four constructs. ρ is the Pearson correlation coefficient, significant at .05 level.

	Mean	Var	Std Dev
PVL	5.584	.009	.095
SWB	5.281	.053	.231
SWC	3	.268	.518
SFC	5.625	.004	.062

ρ	PVL	SWB	SWC	SFC
PVL	1	.876	−.350	.083
SWB		1	−.453	.097
SWC			1	−.620
SFC				1

the corpus. These adaptations made the persona and task descriptions, and examples for few-shot prompting, more domain-specific. The experimental setting consists of an interface to let PA domain experts query the model about the Public Contract Code and evaluate the correctness, completeness, and fluency of the answers. The users then filled out a satisfaction questionnaire to assess system usability and users' resistance to integrate this tool into their workflow. Results reveal a general users' satisfaction with the system: it achieves a SUS of 82.19 and a NPS of 25. Questions for assessing the correctness, completeness and fluency of the answers all achieve a mean score above 3.70. Correctness and completeness mean scores are backed up by the answers on how to improve the system, that highlight the significance of giving more specific answers and the respective articles references. Results of the survey on users' resistance make clear that users consider this tool beneficial to their workflow: positive constructs obtain a mean value above 5, whereas the negative construct 3. Altogether, expert users consider the proposed system a valuable and helpful tool for their way of working.

Concerning future works, according to the answers to the open-ended question for tool improvements, it emerged that providing the textual reference to exact articles and regulations would make the system more reliable and trustworthy. Moreover, performing a comparative analysis of vector-based RAG and Graph RAG techniques may provide interesting insights. Finally, modelling legislation changes over time is a non-trivial task that would bring the proposed system a step further.

Acknowledgments

We acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), Spoke 6 - Symbiotic AI (CUP H97G22000210007) under the NRRP MUR program funded by the NextGenerationEU.

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly for sentence polishing, and ChatGPT for aiding the translation of legal terminology and expressions. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] P. Lops, M. Di Ciano, N. Lopane, L. Siciliani, V. Taccardi, E. Ghizzota, G. Semeraro, et al., Ai-based decision support systems for the management of e-rocurement procedures., in: IIR, 2022.

[2] L. Siciliani, V. Taccardi, P. Basile, M. Di Ciano, P. Lops, Ai-based decision support system for public procurement, Information systems 119 (2023) 102284.

[3] L. Siciliani, E. Tanzi, P. Basile, P. Lops, Automatic generation of common procurement vocabulary codes., in: CLiC-it, 2023.

[4] L. Siciliani, E. Ghizzota, P. Basile, P. Lops, Oie4pa: open information extraction for the public administration, Journal of Intelligent Information Systems 62 (2024) 273–294.

[5] E. Musumeci, M. Brienza, V. Suriani, D. Nardi, D. D. Bloisi, Llm based multi-agent generation of semi-structured documents from semantic templates in the public administration domain, in: International Conference on Human-Computer Interaction, Springer, 2024, pp. 98–117.

[6] A. Colombo, A. Bernasconi, S. Ceri, An llm-assisted etl pipeline to build a high-quality knowledge graph of the italian legislation, Information Processing & Management 62 (2025) 104082.

[7] B. AlKhamissi, M. Li, A. Celikyilmaz, M. Diab, M. Ghazvininejad, A review on language models as knowledge bases, 2022. URL: https://arxiv.org/abs/2204.06031. arXiv:2204.06031.

[8] A. Hogan, X. L. Dong, D. Vrandečić, G. Weikum, Large language models, knowledge graphs and search engines: A crossroads for answering users' questions, 2025. URL: https://arxiv.org/abs/2501.06699. arXiv:2501.06699.

[9] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[10] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, H. Hajishirzi, When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023. URL: https://arxiv.org/abs/2212.10511. arXiv:2212.10511.

[11] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, J. Larson, From local to global: A graph rag approach to query-focused summarization, arXiv preprint arXiv:2404.16130 (2024).

[12] Y. Hu, Z. Lei, Z. Zhang, B. Pan, C. Ling, L. Zhao, Grag: Graph retrieval-augmented generation, 2024. URL: https://arxiv.org/abs/2405.16506. arXiv:2405.16506.

[13] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, S. Tang, Graph retrieval-augmented generation: A survey, 2024. URL: https://arxiv.org/abs/2408.08921. arXiv:2408.08921.

[14] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, P. Liang, Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics 12 (2024) 157–173.

[15] G. Klein, B. Moon, R. R. Hoffman, Making sense of sensemaking 1: Alternative perspectives, IEEE intelligent systems 21 (2006) 70–73.

[16] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., Gpt-4 technical report, arXiv preprint arXiv:2303.08774 (2023).

[17] A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al., Qwen2. 5 technical report, arXiv preprint arXiv:2412.15115 (2024).

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al., Llama 2: Open foundation and fine-tuned chat models, arXiv preprint arXiv:2307.09288 (2023).

[19] G. Team, R. Anil, S. Borgeaud, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, K. Millican, et al., Gemini: a family of highly capable multimodal models, arXiv preprint arXiv:2312.11805 (2023).

[20] A. Hogan, E. Blomqvist, M. Cochez, C. d'Amato, G. de Melo, C. Gutiérrez, S. Kirrane, J. E. Labra Gayo, R. Navigli, S. Neumaier, A.-C. Ngonga Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. F. Sequeda, S. Staab, A. Zimmermann, Knowledge Graphs, number 22 in Synthesis Lectures on Data, Semantics, and Knowledge, Springer, 2021. URL: https://kgbook.org/. doi:10.2200/S01125ED1V01Y202109DSK022.

[21] V. A. Traag, L. Waltman, N. J. Van Eck, From louvain to leiden: guaranteeing well-connected communities, Scientific reports 9 (2019) 1–12.

[22] J. Brooke, et al., Sus-a quick and dirty usability scale, Usability evaluation in industry 189 (1996) 4–7.

[23] J. R. Lewis, The system usability scale: past, present, and future, International Journal of Human–Computer Interaction 34 (2018) 577–590.

[24] J. Sauro, J. R. Lewis, Quantifying the user experience: Practical statistics for user research, Morgan Kaufmann, 2016.

[25] I. Mahmud, T. Ramayah, S. Kurnia, To use or not to use: Modelling end user grumbling as user resistance in pre-implementation stage of enterprise resource planning system, Information Systems 69 (2017) 164–179.

[26] H.-W. Kim, A. Kankanhalli, Investigating user resistance to information systems implementation: A status quo bias perspective, MIS quarterly (2009) 567–582.