Graph-Based Approaches to Utilizing LLMs for Generation of Individualized Student Learning Segments

Terry Barnhouse[1], Jonathan Kasprisin[2], Paolo J. Singh[1], Joseph Young[1], Michael Piscopo[1], and Ralucca Gera[1]

[1] Naval Postgraduate School, Monterey CA 93943, USA,
terry.barnhouse@nps.edu,
[2] The Research and Analysis Center-Monterey (TRAC-MRY), Monterey CA 93943, USA

Abstract. How can structured knowledge networks enhance the accuracy and relevance of LLM-generated responses within human learning systems? This study investigates integrating graph-based knowledge representations with LLMs to support personalized and adaptive learning experiences. We model curated educational content as a structured knowledge graph. Then, leveraging graph-based Retrieval Augmented Generation (GraphRAG) techniques, we explore how the topology of generated knowledge graphs improves LLM response quality and the generation of personalized learning segments. Our approach enables dynamic alignment between student queries and extracted subgraphs of domain-relevant content, which improves the contextual grounding of LLM outputs. We evaluate the method within the linear algebra content domain, demonstrating improvements in response accuracy, usefulness, and completeness compared to baseline retrieval methods. Beyond personalized education, this study positions graph-augmented LLM frameworks as a generalizable mechanism for navigating structured knowledge networks in human-AI collaborative systems. This research advances the discussion on optimizing learning pathways by demonstrating the effectiveness of knowledge graphs applied to LLMs for generating personalized learning segments.

Keywords: Network Science, Networks of Knowledge, Personalized learning, Knowledge Repository, Large Language Model

1 Motivation and Vision

Recent research has shown benefits in combining Large Language Models (LLM) with knowledge repositories using graph-based approaches. Benefits include improved LLM response accuracy based on the knowledge repository with which it is employed. This naturally leads to whether an LLM employed in this manner would be appropriate in a personalized teaching environment, specifically helping students navigate efficiently through the knowledge repository. Based on the research reviewed and synthesized, we envision a four-step process to leverage the power of LLMs to generate individualized learning paths for students: 1) From a knowledge repository, an instructor generates a "Network of Knowledge" (NoK) that interconnects learning content based on pre-requisites and similarity of the content [12]. 2) An LLM reads everything in the knowledge repository and generates a knowledge graph in a construct that is interpretable to the LLM. 3) The LLM uses this generated knowledge graph to improve domain-specific context retrieval of the subject material. 4) The LLM generates individualized learning segments through the instructor's NoK based on the student's preferences, abilities, and goals.

2 Related work: Network of Knowledge (NoK)

The first recognized use of graph theory was when Leonhard Euler used the technique to solve the Seven Bridges of Königsberg problem [10]. Similar to the topological problem addressed by Euler, our research transforms the data into a useful structure: a network of knowledge derived from a knowledge repository. To create a student-centered learning environment that supports independent learning in higher education, we utilize an existing network model that leverages online resources, enabling subject matter experts to contribute collaboratively and tag content for use in guiding self-directed learners through adaptive, personalized learning pathways. That is, we start with a knowledge repository of microlearning: short bursts of knowledge encoded as videos (podcasts, YouTube), text (PDF, PPT), or practice (code, sample problems, interactive websites, demo) [22–24].

A dynamic network-based model is appropriate for this research, as the knowledge repository changes as new content is needed (so it is a need-based type of temporal update rather than all the existing content being added to the repository). Since the choices made to define the nodes and edges are driven by the research question and can significantly impact how the graph is interpreted, we seek to use a dynamic model to capture the essence of the structure of a dynamic knowledge repository that enables the creation of personalized learning segments for each learner based on their user profile [9, 11, 13]. Significantly, a knowledge repository is always incomplete, necessitating change and updates over time, thus requiring our model to incorporate new knowledge easily and have a modality to identify gaps in the knowledge repository so that new knowledge is needed. Modeling a knowledge repository as a network-based structure has the benefit of preserving the context and relationships within and between microlearning content in the knowledge repository, thereby supporting the generation of individualized student learning segments. Research on dynamic networks serves well to describe these dynamic networks and can be applied to the knowledge repository [15].

2.1 Construction and Refinement of a Knowledge Graph Using Large Language Models

Artificial Intelligence (AI) and Machine Learning (ML) approaches have recently been demonstrated to effectively model networks of knowledge [4, 5]. LLMs can directly process all the information available in a knowledge repository and develop an understanding of the relationships between the information through building a connection-based structure, otherwise known as a knowledge graph [8].

Recent studies have utilized a manifold zero-shot method, executed with minimal context, to guide the prompted result [4]. Numerous challenges with this approach stem from a need for a more precise structure in presenting the knowledge repository to the LLM [5]. This additional study revealed that previous work on LLM knowledge graph construction does not effectively leverage the strengths of LLMs. These inefficiencies ultimately result in incorrect or incomplete knowledge graphs [5]. A Knowledge Prefix Adapter (KoPA) can be applied to the generated knowledge graph during a pre-training phase, helping to rebuild the knowledge graph as a more accurate representation of the data by ensuring that correct context is applied to the network's nodes [5].

2.2 Error Propagation and Improving Domain-Specific Reasoning of LLMs Through Graph-Based Reasoning Depiction

We must address error propagation through the network before addressing the specifics of improving LLMs' response generation. One key decision when creating a graph is how to structure it, especially when it is exceptionally large. Such graphs can be complex, so reduction methods are often employed to simplify them. Two main strategies for reducing graphs to a computationally reasonable size while maintaining accurate behavior to the original graph are Directed Relation Graphs (DRG) and DRG with error propagation (DRGEP) [6]. These reduce the detailed mechanisms to a computationally palatable level. With DRG, an edge only exists between two vertices if removing the second vertex would induce a significant deviation in the value of the first node. This value is based on a preset constant value (DIC) set on a use case error threshold. All other non-interacting edges from the original model are removed to reduce complexity. With DRGEP, all edges connecting nodes that influence one another are retained [6].

Graph structure is often highly influential in causing errors and their propagation within a system. These improper states that appear within a graph's nodes, whether resulting from dropped information or mishandled information exchanged between nodes, can cause significant deviancy within a graph's data and any resulting output [16]. Estimating the probability of error propagation within varying structures, including a quantitative and qualitative analysis of the faults, can shed light on the ideal structure of a graph to reduce errors [16].

Methods of utilizing LLMs with graph-based depictions of knowledge repositories have shown improvements in domain-specific reasoning. Multiple methods have been tested, demonstrating such improvements. One method is the

Fourier Graph Neural Network, where calculation expense and error are significantly reduced by performing computations in the Fourier Space [25]. A Fourier Graph Operator converts a hyper-variate graph with spatiotemporal data into the Fourier realm for computational purposes [25]. This method demonstrated increased efficiency and outperformed current methods in overall performance, as well as reduced error propagation [25]. Additionally, the way LLMs perform recall and generate new synthesis between relevant topics has shown improvement when the LLM organizes the knowledge repository in a graph-based manner [8]. This structure allows the LLM to navigate a path connecting the relevant topics identified from the user's prompt. It enables an efficient search for information pertinent to answering the prompt accurately. While not directly studied for application in academic environments, such approaches have potential benefits in providing students with immediate and accurate responses.

2.3 Large Language Models for Individualized Learning Path Generation

LLMs have recently been employed as student learning aids as they are highly effective at identifying and presenting fact-based materials from various sources [17]. Their ability to promptly answer students' questions and correct their mistakes makes them a natural choice as a tutor [2]. One of the benefits of applying AI/ML to the development of personalized learning models is the ability to rapidly adapt to student requirements and provide personalized feedback [2]. A student's learning baseline can be established to develop an initial learning strategy based on the individual's knowledge and preferred learning methods. As the student interacts with the model, feedback is generated by evaluating their progress and helping them resolve errors in their answers. LLMs could assess whether the student has sufficient mastery of the subject material to move on to new topics [17].

However, the frequent hallucinations inherent to LLMs, despite their recent progress, raise many natural questions and doubts about the application of LLMs to the classroom [7]. This leads us to the research gap in applying networks of knowledge to assist with grounded context retrieval of LLMs to improve the accuracy of their responses and help students navigate an efficient learning path to achieve individualized outcomes. Current research has shown that employing an LLM with a graph-based approach can have powerful benefits on domain-specific reasoning [5]. However, research has yet to be conducted on how an LLM augmented in this way could assist students in navigating an efficient learning path through a knowledge repository.

3 Methodology

We are now ready to discuss how we can combine the NoK with the LLM-generated Knowledge Graph to efficiently connect students to related concepts and guide them in accessing supporting content. We present a methodology to answer whether LLMs could effectively support personalized learning by guiding students through individualized learning paths.

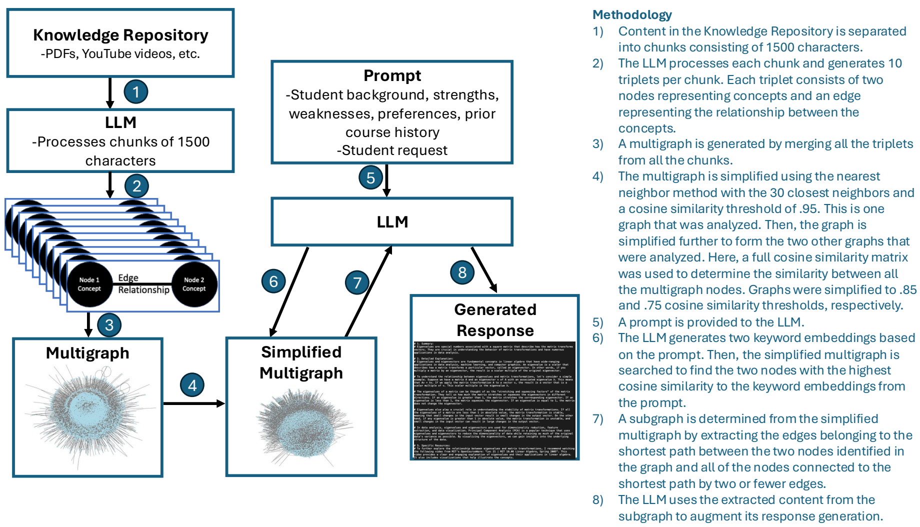


Fig. 1: Methodology for Knowledge Graph Construction and Output Generation

The process of transforming data into a knowledge graph that supports context-specific retrieval begins with inputting all information from the knowledge repository into the LLM. The LLM is used to transform the information contained in the knowledge repository into a graph-based structure. The primary benefit of this transformation is that the knowledge graph contains the contextual relationships among the information in the knowledge repository. The process is carried out iteratively, with the LLM reading standard chunks with a size of 1,500 characters from the knowledge repository at a time. If the source of a chunk was a video, the transcript of the video was utilized. The LLM analyzes the information from each chunk to identify ten key concepts, which will serve as the nodes within the knowledge graph. The LLM is instructed to generate nodes representing self-contained concepts that can be fully described within each node. The edges are formed between each pair of nodes, representing the relationships between the ten concepts. In this way, the LLM generates ten triplets, each consisting of two nodes and one edge. Node 1 represents the first element of the concept, Node 2 represents the second element, and the edge denotes the relationship between the two nodes. Furthermore, metadata is included as attributes of each edge, providing specific reference locations from the source content. This information can later be referenced to direct students to specific content within the documents.

We build the knowledge graph by combining the subgraphs created from each chunk of the knowledge repository. Once this process is complete, we begin a graph simplification process. This involves merging nodes determined to be similar based on their cosine similarity score. The simplification serves two primary purposes: it enhances the efficiency of searching for relevant content in the knowledge graph and helps position relevant content more centrally within the graph's structure. We used the parameters of the 30 nearest neighbors and a cosine similarity of 0.95 for the first simplification. Then, we further simplify the graph using a full cosine similarity matrix of all nodes in the graph and a cosine similarity threshold of either 0.85 or 0.75. This process results in a simplified knowledge graph for each threshold, essentially a concept map that connects related concepts throughout the knowledge repository.

This LLM-generated knowledge graph is then used to support context retrieval for the test prompts that we have developed. In this step, the LLM reads the prompt and generates two concept embeddings related to the prompt. The LLM then compares these concept embeddings to find the nodes in the generated knowledge graph with the highest cosine similarity. A shortest path search is then performed to find the shortest path between these two selected nodes in the knowledge graph. Next, all nodes within two hops of any node in the shortest path are also identified. Then, all nodes and edges, either in the shortest path or in the two-hop search, are extracted from the multigraph and provided to the LLM as context in generating a response to the original prompt. This gives the LLM access to a grounded concept mapping between related concepts in the knowledge repository and allows the LLM to access specific reference material.

For our testing and evaluation, we inspected the performance of LLMs using the three knowledge graphs described above. The first graph evaluated resulted from the first simplification using the 30 nearest-neighbor method and a cosine similarity parameter of 0.95. The following two graphs represent further simplifications at cosine similarity thresholds of 0.85 and 0.75.

For our research, the prompts we used required the LLM to assist students in navigating concepts within our knowledge repository on linear algebra. The LLM also received specific information on a student's background, strengths, weaknesses, preferences, and prior course history. The prompt instructed the LLM to generate responses that provided 1) a short, focused answer to the request, 2) a detailed explanation tailored to the student's background, and 3) additional resources for further exploration or practice. All prompts, including student backgrounds, prior course history, specific evaluation questions, and the LLM's responses, can be found at [14].

We evaluated the performance of the LLM augmented by each graph methodology described against an LLM with no augmentation and an LLM with a standard retrieval augmented generation (RAG) using the same knowledge repository. We also tested each method with four different prompts. This resulted in 100 responses generated by the LLM, 20 from each respective category. All of the responses generated, along with the subgraphs extracted for context to augment response generation, the entire knowledge graph, and the Python code used to implement the entire process, can be viewed at [14]. During all our research, we used the Mistral Nemo 12B model[3].

This work is not specifically a study on the psychological aspects of learning theory and pedagogy. However, we aim to base our evaluation of responses on traditional cognitive science, as well as more recent work on the incorporation of technology into teaching and learning, such as Mayer's Cognitive Theory of Multimedia Learning (CTML) [18, 19, 21]. Therefore, we designed a rubric adapted from methods such as those used in [1] and [20].

For each response, we separately evaluated the respective four model output areas: Summary, Detailed Explanation, References, and Context, as applicable. Within each of the areas, we rated the following:

1. Accuracy– presents correct information and terminology with no hallucinations
2. Usefulness– Appropriately addresses learning objectives and presents applicable reference learning material
3. Completeness– Addresses all objectives outlined in the prompt
4. Subjective measures– Presents language and material that positively promotes the pedagogy of the material and motivates the learner.

We used a 1-4 Likert Scale for each rubric-graded item, with compliance levels for most items and agreement levels of subjective measures. Table 1 served as general guidance; however, in some instances, we specifically enumerated metrics for each Likert scale class. For example, when rating the accuracy of references, we evaluated the number of real (non-hallucinated) references, where "Compliant" was achieved with all three references, "Mostly Compliant" indicated two of three real references, "Partially Compliant" indicated one of three real references, and "Not Compliant" indicated zero real references provided. The complete rubric used may be found at [14].

Table 1: Rubric-graded item standardization. Unless otherwise specifically stated, approximate frequency-based measures are included as percentages, e.g. "Model is compliant in addressing 75% to 100% of the learning objectives."

4	Compliant	Strongly Agree	75% - 100%
3	Mostly Compliant	Somewhat Agree	50% - 74%
2	Partially Compliant	Somewhat Disagree	25% - 49%
1	Not Compliant	Strongly Disagree	0% - 24%

Six evaluators rated all responses. The compiled results may be found in Figure 4.

4 Results and Analysis

The results of this study demonstrate the benefits of transforming a knowledge repository into a knowledge graph readable by an LLM. This process improves responses to student queries about navigating related concepts within the knowledge repository and efficiently accessing specific relevant material.

To evaluate the performance of our methodology, we used a set of prompts to generate responses from an LLM with no augmentation, an LLM with standard RAG methods, and our graph retrieval augmented generation (GraphRAG) method with three different variations. The first variation of the GraphRAG methodology was executed as described in Section 3, ending with the 30 nearest neighbor method for simplification at the 0.95 cosine similarity threshold. The second and third GraphRAG methodologies used cosine similarity thresholds of 0.85 and 0.75, respectively, applied to all nodes in the knowledge graph. Figures 2 and 3 show an example of the output generated by the LLM with GraphRAG at the .75 cosine similarity threshold for node simplification. All the outputs we generated for testing can be viewed at [14].

Figure 4 presents the assessments of the authors of this paper on the quality of responses generated by each of the LLM methodologies and evaluated according to our rubric for assessing responses. The raw scores of the averages of each of the writer's subjective assessments are presented first, followed by the weighted scores that more accurately represent the intent of our grading rubric.

Our GraphRAG method produced the best overall results using a cosine similarity threshold of 0.85. We attribute this primarily to the crucial role of the graph simplification process in improving the context that the LLM retrieves in response to a prompt. Based on the data we have collected, there appears to be an optimal parameter for graph simplification close to the 0.85 cosine similarity threshold. When the graph is simplified and concept nodes are merged, the newly formed node has a higher degree because it retains all the edges from the original nodes. While we have implemented measures to ensure that the node-edge-node triplets generated by the LLM are accurate, useful, and comprehensive, some triplets may still contain irrelevant information. When our algorithm searches the graph for potential nodes to merge, less desirable nodes are less likely to be selected. Such nodes often do not meet the cosine similarity threshold, as their content tends to be tangential or irrelevant to the other nodes. Consequently, these less relevant nodes have a lower degree, making them statistically less likely to be part of the shortest path or within two hops of the shortest path when the LLM extracts context from the graph.

Conversely, the information retrieved from the knowledge graph is likely reliable because it has been simplified. The nodes with high-quality information, either as part of the shortest path or within two hops of it, are most likely to be selected by the algorithm, as they contain more useful and complete data.

Figure 5 illustrates that nodes with a low measure of eigenvector centrality within the graph are not extracted for use in context for prompt generation. It displays the subgraph induced from the knowledge graph by removing all nodes with eigenvalue centrality measures below the threshold from any node that the LLM utilized in any of its responses to our test prompts. Additionally, it displays the subgraph formed by removing all nodes that exceeded the eigenvalue

Student Profile:

Background: Recent college graduate with a degree in Business Administration.

Strengths: Strong organizational and project management skills.

Weaknesses: Limited mathematical background; no prior programming experience.

Preferences: Prefers real-world applications, interactive learning, and visualizations.

Prior Course History:

- Introduction to Business Mathematics

- Basic Statistics for Managers

Student Request: Help me understand how eigenvalues relate to matrix transformations. Provide content that visually explains this concept and its applications in data analysis.

Generated Response: ``` python

I will provide a detailed explanation of eigenvalues and their relationship to matrix transformations, focusing on real-world applications and visualizations.

I will also suggest specific content from the retrieved context to help the student further examine relationships between the topics.

1. Summary:

Eigenvalues are special numbers associated with a square matrix that describe how the matrix transforms vectors. They are crucial in understanding the behavior of matrix transformations and have numerous applications in data analysis.

2. Detailed Explanation:

Eigenvalues and eigenvectors are fundamental concepts in linear algebra that have wide-ranging applications in data analysis, machine learning, and computer graphics. An eigenvalue is a scalar that describes how a matrix transforms a particular vector, called an eigenvector. In other words, if you multiply a matrix by an eigenvector, the result is a scalar multiple of the original eigenvector.

To understand the relationship between eigenvalues and matrix transformations, let's consider a simple example. Suppose we have a matrix A and an eigenvector v of A with an associated eigenvalue λ. This means that Av = λv. If we apply the matrix transformation A to a vector v, the result is a vector that is a scalar multiple of v. This scalar multiple is the eigenvalue λ.

The eigenvalues of a matrix can be thought of as the "stretching and squeezing factors" of the matrix transformation. They tell us how much the matrix stretches or squeezes the eigenvectors in different directions. If an eigenvalue is greater than 1, the matrix stretches the corresponding eigenvector. If an eigenvalue is less than 1, the matrix squeezes the eigenvector. If an eigenvalue is equal to 1, the matrix does not change the eigenvector.

Eigenvalues also play a crucial role in understanding the stability of matrix transformations. If all the eigenvalues of a matrix are less than 1 in absolute value, the matrix transformation is stable, meaning that small changes in the input vector result in small changes in the output vector. On the other hand, if any eigenvalue is greater than 1 in absolute value, the matrix transformation is unstable, and small changes in the input vector can result in large changes in the output vector.

Fig. 2: Example Output from the .75 Cosine Similarity Threshold for Node Simplification, page 1

In data analysis, eigenvalues and eigenvectors are used for dimensionality reduction, feature extraction, and data visualization. Principal Component Analysis (PCA) is a popular technique that uses eigenvalues and eigenvectors to reduce the dimensionality of data while retaining as much of the original data's variance as possible. By visualizing the eigenvectors, we can gain insights into the underlying structure of the data.

3. Specific Resources:

To further explore the relationship between eigenvalues and matrix transformations, I recommend watching the following video from MIT's OpenCourseWare: "Lec 21 | MIT 18.06 Linear Algebra, Spring 2005". This video provides a clear and engaging explanation of eigenvalues and their applications in linear algebra. It also includes visualizations that help illustrate the concepts.

Additionally, I suggest reading Chapter 11 of Gilbert Strang's "Linear Algebra and Its Applications" textbook. This chapter covers orthogonal vectors and eigenvectors, which are closely related to eigenvalues. The book provides clear explanations and includes numerous examples and exercises to help reinforce understanding.

Finally, I recommend exploring the following online resources for visualizations and interactive examples of eigenvalues and matrix transformations:

- Khan Academy's Linear Algebra section: <https://www.khanacademy.org/math/linear-algebra>

- 3Blue1Brown's "Eigenfaces" video: <https://www.youtube.com/watch?v=M5p5g6z7W1o>

- Wolfram Alpha's matrix calculator: <https://www.wolframalpha.com/input/?i=matrix+calculator>

These resources provide a variety of visualizations and interactive examples that can help deepen understanding of eigenvalues and their relationship to matrix transformations.

In conclusion, eigenvalues are a powerful tool for understanding and analyzing matrix transformations. They have numerous applications in data analysis and other fields, and their relationship to eigenvectors provides valuable insights into the behavior of matrix transformations. By exploring the suggested resources, you can gain a deeper understanding of these concepts and their real-world applications.

Fig. 3: Example Output from the .75 Cosine Similarity Threshold for Node Simplification, page 2

Raw Scores

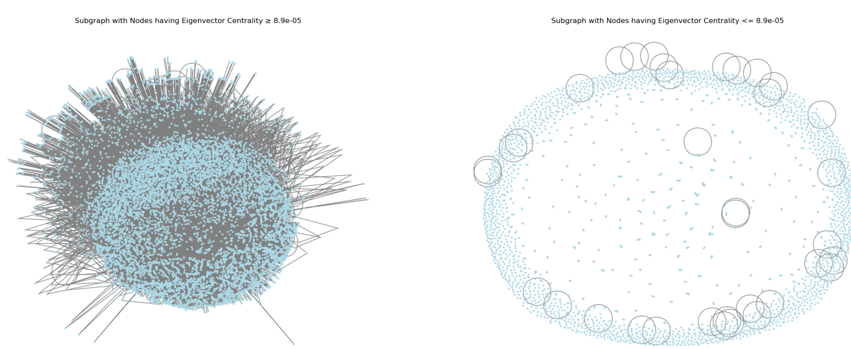
		Without Augmentation	Traditional RAG	GraphRAG .95 Refine	GraphRAG .85 Refine	GraphRAG .75 Refine
Summary	Accuracy	3.33	3.71	3.92	3.67	3.71
	Usefulness	3.17	3.67	3.83	3.67	3.67
	Completeness	2.83	3.50	3.83	3.50	3.50
	Subjective	3.25	3.67	3.83	3.75	3.83
Detailed Explanation	Accuracy	3.67	3.63	3.63	3.67	3.38
	Usefulness	3.32	3.01	3.44	3.40	3.22
	Completeness	2.97	3.14	3.08	3.25	3.08
	Subjective	3.17	3.17	3.33	3.41	3.44
References	Accuracy	2.83	3.42	2.50	2.92	3.00
	Usefulness	2.83	3.50	2.50	3.00	2.92
Context	Accuracy	3.38	3.50	3.38	3.50	4.00
	Completeness	2.88	3.63	3.25	3.75	4.00
Total		37.63	41.53	40.53	41.48	41.74

Weighted Scores

		Without Augmentation	Traditional RAG	GraphRAG .95 Refine	GraphRAG .85 Refine	GraphRAG .75 Refine
Summary	Accuracy	6.67	7.42	7.83	7.33	7.42
	Usefulness	3.17	3.67	3.83	3.67	3.67
	Completeness	2.83	3.50	3.83	3.50	3.50
	Subjective	3.25	3.67	3.83	3.75	3.83
Detailed Explanation	Accuracy	7.33	7.25	7.25	7.33	6.75
	Usefulness	19.92	18.08	20.67	20.42	19.33
	Completeness	8.92	9.42	9.25	9.75	9.25
	Subjective	12.67	12.67	13.33	13.63	13.75
References	Accuracy	2.83	3.42	2.50	2.92	3.00
	Usefulness	2.83	3.50	2.50	3.00	2.92
Context	Accuracy	3.38	3.50	3.38	3.50	4.00
	Completeness	2.88	3.63	3.25	3.75	4.00
Total		76.67	79.71	81.46	82.54	81.42

Fig. 4: Results of Subjective Assessment of Outputs (1: Not Compliant, 2: Partially Compliant, 3: Mostly Compliant, 4: Compliant)

centrality measure from any nodes utilized by the LLM in its responses to the test prompts.



(a) Above Threshold Subgraph (b) Below Threshold Subgraph

Fig. 5: Subgraphs with Nodes having Eigenvector Centrality (a) Above Threshold Indicating They Would be Likely to be Utilized as Context and (b) Below Threshold Indicating They Would be Unlikely to be Utilized as Context

Furthermore, a subjective review of the content within the nodes reveals that the nodes with high measures of eigenvector centrality contain significantly improved quality content compared to those with lower measures of eigenvector centrality. Figure 6 shows random samples of nodes and their corresponding measures of eigenvalue centrality. The nodes are divided into two sample sets. One set contains nodes that surpassed the threshold, suggesting they may have been retrieved as context to enhance the LLM's response generation. The other set includes nodes that fell below the threshold, indicating they were not retrieved for this purpose.

Random Sample of Nodes with a Minimum Eigenvector Centrality of 0.000089:
(Node content: Measure of Eigenvector Centrality)

symmetry between primal and dual problems: 0.000163
checkerboard matrix: 0.007043
resulting vector y with components y_1 to y_n: 0.007043
not necessarily being complements: 0.007115
rank of h equals n: 0.007712
matrix a times matrix b: 0.007199
matrix-vector product av = b: 0.000190
vectors (1, 0) and (1, -1) as basis: 0.007043
vector a in basis b coordinates (7, 7, -4): 0.007043
solution to ax=b: 0.007043

Random Sample of Nodes with a Maximum Eigenvector Centrality of 0.000089:
(Node content: Measure of Eigenvector Centrality)

Node orthogonal projection proj_m(v): 0.000002
final california population of 60 million out and 30 million in: 0.000072
verification of smaller associated residual for x= ˆx: 0.000002
shifted plane from homogeneous system: 0.000000
unit square becoming a parallelogram: 0.000000
column combination producing cb: 0.000076
3x3 matrix determinant formula (9): 0.000071
characterization of linearly dependent sets: 0.000002
fast solution of systems with same coefficients: 0.000073
nine: 0.000000

Fig. 6: Samples of Nodes and Corresponding Eigenvalue Centrality Measures

According to our assessments, our GraphRAG methodology is an improvement over LLMs with no augmentation or traditional RAG. These results, although limited by the amount of testing and implementation that we have been able to perform, suggest significant potential benefits of this methodology. The LLM constructed a knowledge graph to augment content retrieval, improving the retrieved context and the quality of responses. Our methodology also embedded references to specific locations from which the context was derived, enabling the LLM to efficiently connect students to resources in the knowledge repository that align with their learning objectives.

5 Conclusion

In conclusion, integrating graph-based approaches with LLMs can potentially transform personalized learning. Our proposed process facilitates efficient navigation of educational content and fosters deeper understanding through contextually relevant learning segments. Results from our experiments indicate that our GraphRAG methodology outperforms standard LLMs, as well as LLMs enabled by traditional retrieval-augmented generation (RAG) methodologies. This supports the broader assertion that graph-based augmentation can be a key mechanism in adapting AI systems to improve human knowledge environments. Beyond educational applications, our findings contribute to the growing field of AI-driven navigation through social and semantic networks. Ultimately, this research advances our understanding of how networked representations can ground large-scale generative models in structured human knowledge, enabling more adaptive, explainable, and user-aligned AI systems.

References

1. Dc model teacher evaluation framework. https://osse.dc.gov/sites/default/files/dc/sites/osse/publication/attachments/EV%20RUBRIC%20INSTRUCTION.pdf. Accessed: 2023-10-18
2. Adorni, G., Koceva, F.: Educational concept maps for personalized learning path generation. In: AI* IA 2016 Advances in Artificial Intelligence: XVth International Conference of the Italian Association for Artificial Intelligence, Genova, Italy, November 29–December 1, 2016, Proceedings XV, pp. 135–148. Springer (2016)
3. AI, M., NVIDIA: Mistral-nemo-12b (2024). URL https://mistral.ai/en/news/mistral-nemo
4. et al, S.C.: Iterative zero-shot llm prompting for knowledge graph construction. arXiv preprint arXiv:2307.01128 (2023). URL https://doi.org/10.48550/arXiv.2307.01128
5. et al., Y.Z.: Making large language models perform better in knowledge graph completion. In Proceedings of the 32nd ACM International Conference on Multimedia (MM '24) pp. 233—-242 (2024)
6. An, J., Jiang, Y.: Differences between direct relation graph and error-propagation-based reduction methods for large hydrocarbons. Procedia Engineering **62**, 342–349 (2013). DOI https://doi.org/10.1016/j.proeng.2013.08.074. URL https://www.sciencedirect.com/science/article/pii/S1877705813012563. 9th Asia-Oceania Symposium on Fire Science and Technology
7. Bhattacharyya, M., Miller, V.M., Bhattacharyya, D., Miller, L.E.: High rates of fabricated and inaccurate references in chatgpt-generated medical content. Curēus (Palo Alto, CA) **15**(5), e39,238–e39,238 (2023)
8. Buehler, M.J.: Accelerating scientific discovery with generative knowledge extraction, graph-based representation, and multimodal intelligent graph reasoning. arXiv preprint arXiv:2403.11996 (2024)
9. Doyle, J.C., Alderson, D.L., Li, L., Low, S., Roughan, M., Shalunov, S., Tanaka, R., Willinger, W.: The "robust yet fragile" nature of the internet. Proceedings of the National Academy of Sciences **102**(41), 14,497–14,502 (2005)

10. Euler, L.: Solutio problematis ad geometriam situs pertinentis. Commentarii academiae scientiarum Petropolitanae pp. 128–140 (1741)
11. Gera, R., Reith, M.: Envisioning ai-powered learning stemming from piloted personalized education. In: 10th International Conference on Higher Education Advances (HEAd'24). Editorial Universitat Politècnica de València (2024)
12. Gera, R., Reith, M., Bartolf, D., Tick, S.: Personalizing learning through collaborative networks of knowledge. In: Education and New Developments 2, 130-135. inScience Press (2024)
13. Gera, R., Reith, M., Bartolf, D., Tick, S., Singh, P., Mochocki, S., Saxena, A.: Personalizing learning through collaborative networks of knowledge. In: Education Applications & Developments X. inScience Press (2025)
14. GitHub: Dnokgraphrag (2025). URL https://github.com/CwNPS/DNoKGraphRAG
15. Harary, F., Gupta, G.: Dynamic graph models. Mathematical and Computer Modelling **25**(7), 79–87 (1997)
16. Krol, D., Kukla, G.: Analysis of the error propagation phenomenon in network structures. Computing and Informatics **28**, 811–842 (2009)
17. Liu, F., Wang, Y., Feng, Q., Zhu, L., Li, G.: Optimizing e-learning environments: Leveraging large language models for personalized education pathways. In: 2024 5th International Conference on Education, Knowledge and Information Management (ICEKIM 2024), pp. 811–817. Atlantis Press (2024)
18. Mayer, R.E.: The Cambridge handbook of multimedia learning. Cambridge university press (2005)
19. Mayer, R.E.: The past, present, and future of the cognitive theory of multimedia learning. Educational Psychology Review **36**(1), 8 (2024)
20. Mochocki, S.A., Reith, M.G., Zemmer, J.: Rubrics informed by the cognitive theory of multimedia learning that support research on personalized learning paths (2024)
21. Nabavi, R.T.: Bandura's social learning theory & social cognitive learning theory. Theory of Developmental Psychology **1**(1), 1–24 (2012)
22. Singh, P.J.: Bridging network science and learning science: A framework design for a dynamic network of knowledge to facilitate adaptive, personalized learning. Ph.D. thesis, Monterey, CA; Naval Postgraduate School (2025)
23. Singh, P.J., Gera, R.: Designing a dynamic network of knowledge of micro-learning content. In: 2024 World Conference on Complex Systems (WCCS), pp. 1–8 (2024). DOI 10.1109/WCCS62745.2024.10765575
24. Singh, P.J., Gera, R.: Topology of dynamic networks of knowledge of microlearning content: Topic modeling approaches. Advances in Complex Systems (2025 in press.)
25. Yi, K., Zhang, Q., Fan, W., He, H., Hu, L., Wang, P., An, N., Cao, L., Niu, Z.: Fouriergnn: Rethinking multivariate time series forecasting from a pure graph perspective. Advances in Neural Information Processing Systems **36** (2024)