Automated Learning Path Evaluation with AI-Generated Judges

Parvathy Menon[a] Nuha Aburamadan[b] Ahmed Ali Seyam[c] Arash Kermani Kolankeh[d]

Canadian University Dubai, Dubai City Walk, Dubai, UAE
[a]20210001663@students.cud.ac.ae; [b]20230004302@students.cud.ac.ae;
[c]ahmed.seyam@cud.ac.ae; [d]arash.kolankeh@cud.ac.ae

Abstract. This paper introduces an approach to evaluating learning path recommendations in e-learning systems. It employs the capabilities of AI-generated personas by large language models (LLMs) to assess learning paths that align with user preferences and behavior. The proposed system uses Wikipedia clickstream data to recommend learning paths through Wikipedia articles. An LLM, such as GPT-4, simulates expert assessments to validate the learning paths, addressing scalability and efficiency challenges in statistical data collection from human experts for validation. This integration reduces reliance on human expertise and enables real-time evaluation. The system architecture uses MongoDB for tree-like data storage and management, and Flask for streamlined automation of the validation process.

Keywords: Learning Path, Recommender Systems, Large Language Models, Artificial Judges, Artificial Intelligence.

1 Introduction

When using knowledge base repositories or online encyclopedias like Wikipedia, Google Scholar, Internet Archive, etc., users click on hyperlinks or embedded text within individual pages that redirect to other relevant pages by topic or some other factors. We hypothesize that based on the statistics of click behavior, we can generate optimized learning paths though the interlinked documents. The hypothesis is based on the intuition that if on a Wikipedia page the user click on a link to a second page, the second page is the prerequisite for understanding the original page. The goal of this paper is not proving this hypothesis but introducing a method for testing such theories. Naturally, to prove the efficiency and accuracy of the recommended learning paths, domain experts would need to be queried which is adequate for individual cases but for analyzing hundreds of different domains and huge number of connected pages, we need a better solution. That's why we propose a learning path recommender system that sends a repository of generated learning paths to a persona generation module that can use LLMs like OpenAI's GPT-4 for assessment. The LLM generates a determinate number of artificial expert personas which can judge the quality and reliability of big numbers of learning paths. This allows for automating the judging process. **Fig.1** illustrates the proposed system. The click data is fed into a Learning Path Generator module

(LPG) that creates recommended learning sequences through the content. Generated paths are stored in a Path Repository, which connects to a Persona Generation module (PG). This module uses an LLM to create AI expert personas based on the common categories among the learning components within a path. The AI personas provide automated feedback about the relevance of the learning paths, replacing the need for human domain experts in the evaluation process.

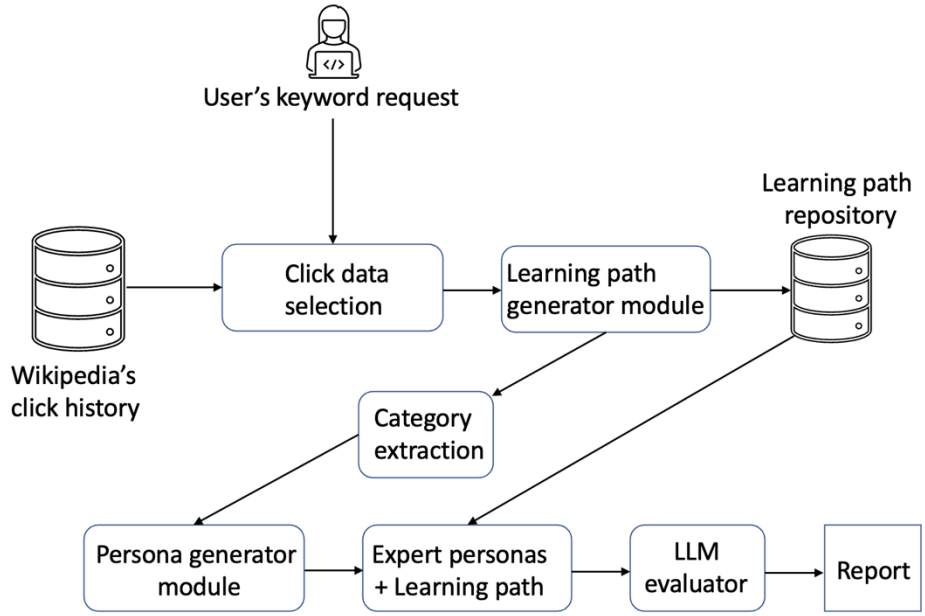

Fig.1 Flow diagram depicting an AI-powered learning path recommender system

2 Related Research

Creating personalized learning paths in e-learning systems is becoming increasingly important, especially in adaptive learning environments. In this line, one can refer to Rahayu's study, which focuses on ontology-based recommender systems that use AI techniques, such as Bayesian networks, to recommend learning paths that adjust based on the user's behavior. This approach is like the current study, which uses hierarchical traversal techniques to guide users through Wikipedia pages, offering personalized recommendations that adapt to how users interact with the content [1].

Nabizadeh et al. explored the personalization of learning paths, emphasizing its importance in addressing the diverse limitations, backgrounds, and goals of users. They

described a learning path as the practical implementation of curriculum design, consisting of a sequence of learning activities aimed at helping users achieve specific learning objectives. Recognizing the growing need for personalization, they reviewed various methods proposed over the last decade, analyzing their techniques, advantages, and disadvantages.The study identified key parameters essential for personalizing learning paths and provided an overview of the approaches used to evaluate these personalization methods. Additionally, they discussed significant challenges associated with learning path personalization, such as scalability, adaptability, and quality enhancement, which remain critical to improving the effectiveness of personalized education systems [2].

In the T. Saito and Y. Watanobe study, the researchers explore the use of recurrent neural networks to develop learning path recommender systems. This approach emphasizes real-time adaptation of learning paths based on user interactions, comparable to the way the current study relies on data-driven traversal and categorization methods. Both studies highlight the importance of utilizing user data to continuously refine recommendations, aiming to make the learning experience more personalized and effective [3].

A different perspective is provided in Fuster-López's study that integrates multiple types of recommenders into a single framework for structuring learning paths. This approach aligns with the current study's method of using hierarchical traversal, demonstrating the benefits of a modular approach in adaptive learning. Both studies emphasize the need for diverse tools to create personalized learning paths, recognizing that flexibility and adaptability can significantly improve user engagement and learning outcomes [4].

Automating persona generation based on data analysis is important during content development, such as feed curating. One paper proposes an approach that makes use of statistical analysis on data collected from large social media sites like YouTube to generate personas based on trends detected. These personas, derived from market segments, offer valuable insights into user preferences and behavior, grounded in real-time data collection and analysis [5].

Qiu et al. proposed a framework leveraging LLM for simulating counselor-client interactions in mental health counseling. Their approach addresses challenges in traditional methods, such as the time-intensive, cost-prohibitive, and privacy-sensitive nature of human annotation, which limits scalability. The framework employs two instances of GPT-4 via zero-shot prompting, where one LLM assumes the role of a client with a realistic user profile, and the other acts as an experienced counselor, delivering professional responses using integrative therapy techniques. They began with automatic assessments of the client-side performance and conducted comparative analyses between LLM-generated and human-generated dialogues. To benchmark the effectiveness of

their LLM-based counselor, extensive experiments were performed, positioning it against state-of-the-art models in mental health. The results demonstrated the potential of LLMs to generate meaningful and professional interactions while revealing areas where machine-generated conversations differ from those of human counselors [6]. A possible concern for the use of personas instead of real experts can be seen when using personas as judges. One study shows findings that personas generated for the purpose of judging often are given simplistic and unrefined prompts and guidelines that can lead to a lack of reliability in generated responses and assessments and even sometimes inconsistency in opinion with a human expert. The authors of this study propose a system that allows the LLM itself to express its level of uncertainty on whatever judgment it makes, with this new system increasing agreements on judgments to a higher degree than previously observed, hence enhancing the reliability and efficiency of using personas as judges [7], [8].

When testing performance, often, responses generated from personas can lack coherence. The quality of the persona generation system can be improved by giving clear and concise prompts and being data-driven. One paper proposes a multi-module model that assigns different roles, such as determining when responses should be given, decoding responses, and detecting the starting point for decoding. The results of this model showed how with a simple chatbot, using some collected data, coherent and diversified responses can be generated [9].

3 Methodology

The project involved several key steps:

a. Initial Python Implementation:

Data Parsing: The project started with using Python for loading and analyzing a large Wikipedia clickstream TSV file dataset, publicly available from Wikipedia, which tracks the number of user transitions between pages. Later this was optimized by using a (key:value) based database.

Hierarchy Traversal: Initially, a recursive function was developed in Python to traverse the hierarchy of pages by recursively selecting the most-clicked child of pages, starting from a parent page. This was done based on the click data provided by Wikipedia.

Wikipedia API Integration: To enrich the recommended paths, the ‘wikipediaapi’ library was used to get summaries and categories for each page.

b. Optimization with MongoDB:

In the improved and optimized implementation, instead of using Python for loading the click data into the memory and traversing it with a recursive function, the data was imported to MongoDB, a NoSQL database, and the functionality of MongoDB, was used for simulating tree traversals and for information retrieval.

MongoDB is organized by using (key:value) to stimulate the graph of interconnected pages. The keys in this case where the page pairs and the values, the number of clicks from the first page to the second one in the pair. The clickstream data was imported into a MongoDB collection and the high-level interaction with the database was then done using Python.

Enhanced Querying: With MongoDB's indexing and query optimization features, it became possible to quickly identify the most frequently clicked child pages for each parent page.

Efficient Traversal: The recursive traversal was adjusted to query the MongoDB collection instead of iterating over data in memory, significantly improving performance. Proper indexing increased the speed considerably.

c. Flask Integration:

Flask Application: To make the results accessible and interactive, a Flask web application was developed. Flask allowed users to input keywords and view the reading paths remotely and continuously.

Dynamic Output: When a user inputs a keyword, the Flask server triggers the traversal process, generating a learning path that includes related Wikipedia page titles, their summaries, and top categories.

d. Data for Persona Generation and Assessment:

Using the learning path recommender system, a repository of 1000 records was generated.
Each record contained an initial keyword, a recommended learning path for that specific keyword, and common categories of the pages located on the recommended learning path. The common categories were used in the next phase of the system for creating AI-driven judges to assess the learning paths. The process is depicted in **Fig. 2.**

The second phase of the recommender system ensures that the generated learning paths are logical and relevant to the goal article. To address this, we integrated a judge persona generation module to assess the quality of the recommended learning paths which for now uses GPT-4io but can be scaled up to support more resource-intensive models.

e. Assessing learning paths with judge personas:

The system was built using Python, with actual persona generation handled by OpenAI's API, with the system structure being based on the framework provided by Chan, X., et al in their paper titled 'Scaling Synthetics Data Creation with 1,000,000,000 Personas' [10]. The system contains modules that handle different aspects of the system including persona generation and learning path assessment.

The persona generation module (PG) was adapted from the code of the above-mentioned paper. This modified code handles extracting the categories from the repository and feeding them as a user prompt to the API to generate personas who are experts in the categories to ensure validity and relevance of assessment.
Learning path assessment: ***The learning path assessment module (LPA)*** combines specified learning path for each keyword to the related generated persona to create a prompt. It then sends the prompt to assess whether the sequence and the content of the recommended learning path are logical for understanding the keyword. The system then iterates through all paths, i.e., sequences of topics, previously saved in a file, and calls the PG and LPA modules to provide an Excel file with the description of the generated personas and their assessments of the assigned learning paths.

Process Flow: The excel file containing the output of the recommender system is analyzed in the next stage. As mentioned, each row of this file contains a keyword, related recommended learning path and a list of categories that are common topics among the recommended pages. For each entry, the categories list is fed to the PG, the persona generating module with a prompt which generates the description of a persona who would be an expert in the topics mentioned in the list of categories. In other words, the persona's background is relevant to the topics. The persona and extracted learning path are then sent to another module where it is asked to assess, as an expert with the related background, whether the learning path is a logical sequence of topics to understand the keyword for that specific entry.

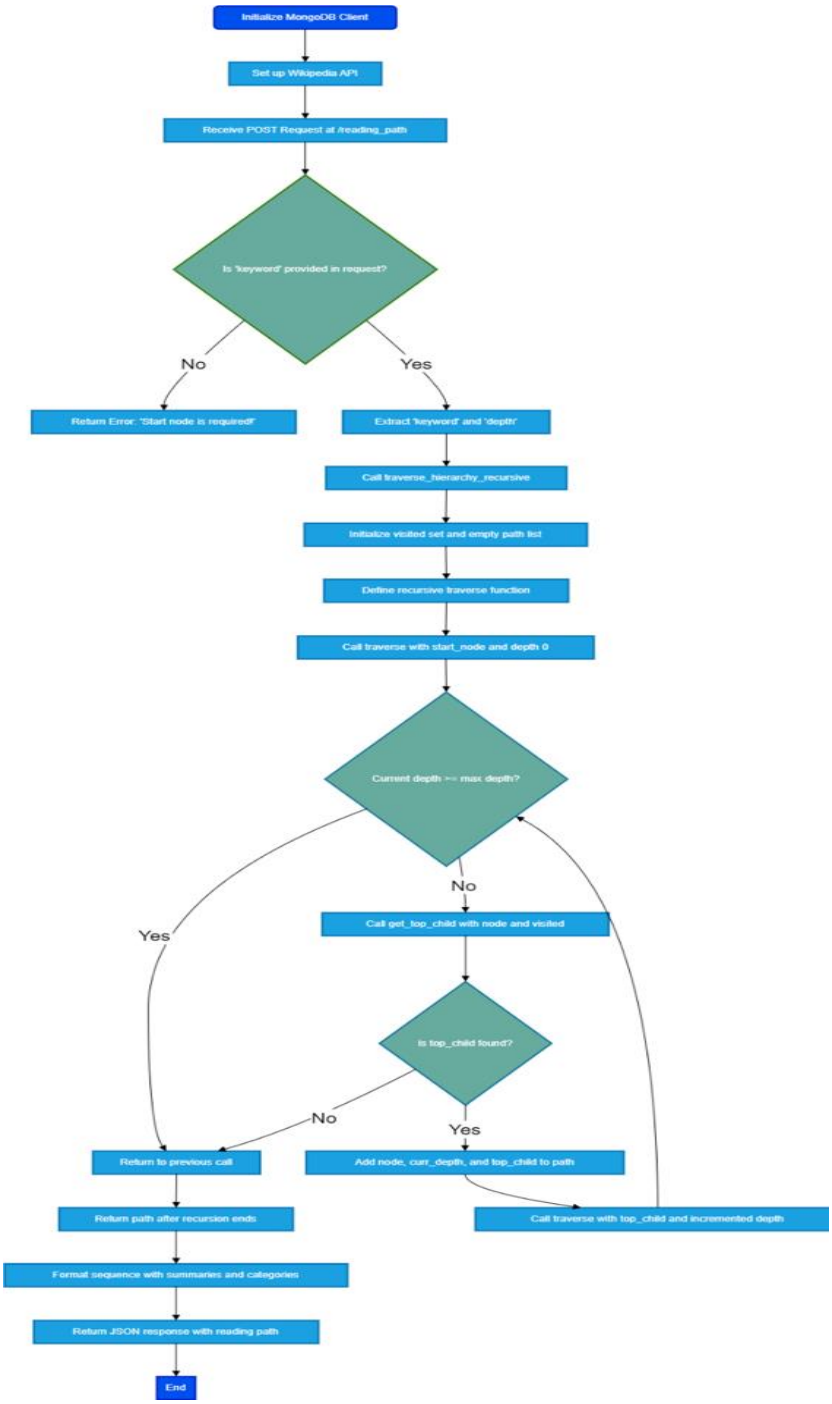

Fig. 2. Proposed approach for Learning path recommender system.

4 Results

The optimized process, implemented using MongoDB and Flask, provided an effective, interactive way to explore Wikipedia clickstream data. Users could input any Wikipedia keyword and instantly receive a logical reading path. The output included up to five related Wikipedia pages, each suggesting a potential next step in the user's learning journey and the top 5 categories associated with each page (**Fig. 3)**. It also gives concise summaries of each page as a part of the output.
As for the results of the judge persona module, the results are saved as a JSON file containing only persona background and assessment of learning path, which could be a 'yes', 'no' or 'unclear' for any number of specified entries **Fig. 4**. With a progress bar being implemented to show status of persona generation and process completion.

Keyword	recom_1	recom_2	recom_3
Large_language_model	Large_language_model	Word_n-gram_language_model	Recurrent_neural_network

Fig. 3. An example of a parent node keyword and recommended learning path


```
[
  {
    "entry": 1,
    "persona_background": "Dr. Alex Turner is a computational linguist and data scientist
     with extensive expertise in language modeling and Markov models. With a keen eye for
     detail, they specialize in identifying and correcting unsourced statements in
     academic articles, ensuring the integrity of scholarly communication.
     Dr. Turner is also well-versed in citation management, particularly in addressing
      issues related to CS1 formatting, including long volume values and numeric author
      names. Their work bridges the gap between theoretical modeling and practical
      applications in natural language processing.",
    "assessment": "Yes"
  }
]
```


Fig. 4. Generated judge persona and learning path assessment

To show effectiveness of generated personas and to ensure assessments are not random, we took a sample of 1000 personas and their assessments and sent them through a program that reads each entry and extracts the final assessment, whether it's a 'yes', 'no' or 'unclear' and calculates the percentage of each in **Table 1**.
We also wanted to check the effects of depth of recommendation on assessments, so we modified the program to accept only three recommendations for each entry and ran everything again. From the analysis received we can see a significant increase in approval rate, implying recommended paths of lesser depth are assessed as more logical or effective to understand initial keyword. As a side effect, the percentage of unclear pages was also increased.

Table 1. *Analysis of generated assessments.*

Learning Path Depth	Approval Rate	Disapproval Rate	Unclear Rate
5	21%	72%	7%
3	32%	52%	16%

5 Conclusion

This work is an example of testing a hypothesis based on statistical data from the opinion of the artificial personas. We had the hypothesis that a learning path could be generated based on the average human behavior. Testing this hypothesis with real human experts would be close to impossible, although we could test it using personas created by an LLM. In our proposed learning path recommender system users could efficiently receive learning paths derived from the Wikipedia click-stream data. The system integrated a user-friendly interface that allows users to view suggested learning paths to understand specific topics along with summaries.

For future developments we plan to investigate refining our persona generation module by incorporating other metrics for assessing our recommendations along with expanding to other online knowledge bases, applying what worked here in other domains as well to improve and streamline content curation potentially. We also want to test the validity of the persona generation module with other LLMs like Falcon or Llama to see which ones perform better. With the acquisition of more compute resources in the future, we will increase the number of generated and tested data points.

References

1. Rahayu, N. W., Ferdiana, R., & Kusumawardani, S. S. (2023). A systematic review of learning path recommender systems. Education and Information Technologies, 28(6), 7437-7460. doi:https://doi.org/10.1007/s10639-022-11460-3.

2. A. H. Nabizadeh, J. P. Leal, H. N. Rafsanjani, and R. R. Shah, "Learning path personalization and recommendation methods: A survey of the state-of-the-art," Expert Syst. Appl., vol. 159, p. 113596, Nov. 2020, doi: 10.1016/j.eswa.2020.113596.

3. T. Saito and Y. Watanobe. Learning Path Recommender System based on Recurrent Neural Network. 2018 9th International Conference on Awareness Science and Technology (iCAST), Fukuoka, Japan, 2018, pp. 324-329, doi: 10.1109/ICAwST.2018.8517231.

4. Fuster-López, A., Cruz, J. M., Guerrero-García, P., Hendrix, E. M. T., Košir, A., Nowak, I., Pereira, A. I. (2024). On conceptualisation and an overview of learning path recommender systems in e-learning. Ithaca: Retrieved from https://ezp.cud.ac.ae/login?url=https://www.proquest.com/working-papers/on-conceptualisation-overview-learning-path/docview/3069342476/se-2.

5. An, J., Kwak, H., & Jansen, B. J. (2016). Validating social media data for automatic persona generation [Review of Validating social media data for automatic persona generation]. In 2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA. IEEE Xplore. https://ieeexplore.ieee.org/abstract/document/7945816 .

6. H. Qiu and Z. Lan, "Interactive Agents: Simulating Counselor-Client Psychological Counseling via Role-Playing LLM-to-LLM Interactions," Aug. 28, 2024, arXiv: arXiv:2408.15787. doi: 10.48550/arXiv.2408.15787.

7. Dong, Y. R., Hu, T., & Collier, N. (2024). Can LLM be a Personalized Judge? ArXiv.org. https://arxiv.org/abs/2406.11657.

8. Mingchen Zhuge et. al. Agent-as-a-Judge: Evaluate Agents with Agents. arXiv preprint arXiv:2410.10934v2, Oct. 2024. [Online]. Available: https://arxiv.org/abs/2410.10934v2

9. Qian, Q., Huang, M., Zhao, H., Xu, J., & Zhu, X. (2017). Assigning personality/identity to a chatting machine for coherent conversation generation. ArXiv.org. https://arxiv.org/abs/1706.02861

10. Chan, X., Wang, X., Yu, D., Mi, H., & Yu, D. (2024). Scaling Synthetic Data Creation with 1,000,000,000 Personas. ArXiv.org. https://arxiv.org/abs/2406.20094