Generación Automatizada de Resultados de Aprendizaje mediante LLM: Diseño y Validación

Learning Outcome Generation using LLM: Design and Validation

Nelson Garrido
CAETI – Universidad Abierta Interamericana – Facultad de Tecnología Informática
Av. Montes de Oca 745, Ciudad de Buenos Aires, Argentina.

Carlos Neil
CAETI – Universidad Abierta Interamericana – Facultad de Tecnología Informática
Av. Montes de Oca 745, Ciudad de Buenos Aires, Argentina.

Claudia Pons
CAETI – Universidad Abierta Interamericana – Facultad de Tecnología Informática
Av. Montes de Oca 745, Ciudad de Buenos Aires, Argentina.

DOI *https://doi.org/10.59471/raia2025217*
Enviado: junio 2025. **Aceptado**: octubre 2025. **Publicado:** diciembre 2025
Como citar: Garrido, N., Neil, C., & Pons, C. (2025). Generación Automatizada de Resultados de Aprendizaje mediante LLM: Diseño y Validación . Revista Abierta De Informática Aplicada, 9(1). *https://doi.org/10.59471/raia2025217*

Resumen

Este artículo explora el uso de la inteligencia artificial para automatizar la generación de Resultados de Aprendizaje (RA) en contextos de educación superior. La propuesta combina un Modelo de Lenguaje Extenso (LLM) con una arquitectura de Generación Aumentada por Recuperación (RAG), con el objetivo de mejorar la precisión, coherencia y relevancia pedagógica de los textos generados. Para lograrlo, se integraron un corpus de documentos disciplinares y una base de datos de RA previamente validados por la comunidad educativa, los cuales fueron utilizados como fuentes contextuales durante el proceso de generación automática. La arquitectura propuesta fue implementada y se analizaron diversos escenarios experimentales utilizando un único curso, modificando configuraciones de entrada como la estructura del prompt y la temperatura del modelo. Los resultados muestran que el sistema es capaz de generar RA estructuralmente correctos y alineados con los parámetros curriculares. Como trabajo futuro, se propone la incorporación de mecanismos automáticos para evaluar la calidad pedagógica, junto con la extensión del modelo para apoyar la generación de otros artefactos educativos relevantes.

PALABRAS CLAVES: automatización educativa, generación de resultados de aprendizaje, modelos de lenguaje extensos, generación aumentada por recuperación.

Abstract

This article explores the use of artificial intelligence to automate the generation of Learning Outcomes (LO) in higher education contexts. The proposal combines a Large Language Model (LLM) with a Retrieval-Augmented Generation (RAG) architecture, aiming to improve the accuracy, coherence, and pedagogical relevance of the generated texts. To achieve this, disciplinary document corpus and a database of LO previously validated by the educational community were integrated and used as contextual sources during the automatic generation process. The proposed architecture was implemented, and various experimental scenarios were analyzed using a single course, modifying input configurations such as prompt structure and model temperature. The results show that the system is capable of generating structurally correct LO, aligned with curricular parameters. As future work, the incorporation of automated mechanisms to assess pedagogical quality is proposed, along with extending the model to support the generation of other relevant educational artifacts.

KEYWORDS: educational automation, learning outcome generation, large language models, retrieval-augmented generation.

Introduction

Learning Outcomes (LO) are clear statements of what students should know, understand, and be able to do at the end of a learning experience (Gaete Quezada, 2021). They are essential in competency-based curriculum design, facilitating alignment between teaching, learning, and assessment (Kennedy, 2006). However, designing effective LO requires pedagogical expertise and time, making their consistent development a challenge for educators.

Manual LO formulation often lacks linguistic and structural uniformity, leading to ambiguities and inconsistent assessment (Biggs, 2003). Moreover, ensuring that LO aligns with course goals and assessment strategies demands continuous refinement. Automation thus emerges as a strategy to improve both efficiency and accuracy in this process.

Text generation approaches typically fall into rule-based or machine learning-based systems (Chu et al., 2025). Rule-based methods often produce rigid or unnatural texts and fail to handle complex contexts (Benites et al., 2023). In contrast, machine learning models—particularly those based on neural networks—have demonstrated greater adaptability and contextual understanding (Benites et al., 2023).

Large Language Models (LLMs) offer significant advantages for generating coherent and context-aware text (Min et al., 2024). When combined with Retrieval-Augmented Generation (RAG), they can incorporate external, domain-specific knowledge during generation, improving relevance and factual accuracy (Posedaru et al., 2024). In the context of LO, this allows the integration of validated curricular materials to strengthen academic alignment (Neil, 2024; Neil et al., 2023). Rather than replacing educators, these technologies aim to support them by simplifying the LO drafting process. This enables teachers to dedicate more attention to designing meaningful learning experiences and refining pedagogical strategies (Yeung et al., 2025).

This article presents a model that integrates LLM with RAG to automate LO generation. It leverages contextual data—such as syllabi and validated LO examples—to enrich generation without sacrificing accuracy or curricular integrity. The remainder of the paper is organized as follows: Section 2 presents the theoretical foundations of LLM and RAG in education. Section 3 details the proposed architecture. Section 4 describes the evaluation methodology and experimental setup. Section 5 discusses the results and limitations. Section 6 concludes and suggests directions for future work.

Learning Outcome Formulation

In practice, manually drafting a LO entails multiple challenges. The lack of standardized guidelines, variability in linguistic structure, and the workload it represents for instructors hinder its consistent application, especially in contexts requiring scalability or the involvement of multiple stakeholders.

The competency matrix, in turn, complements LO writing, as it is a key tool for curriculum design. It allows for the definition of the levels of mastery expected of students in each competency of the graduate profile across the various curricular components of the study plan (Neil et al., 2023). In this framework, the first level of mastery focuses on acquiring basic knowledge with high teacher guidance; the second level develops skills through the application of knowledge with relative autonomy; and the third level integrates the full competency, solving complex problems with complete autonomy. These levels guide the selection of verbs according to established taxonomies (Prieto J., 2012).

To support this process, several authors have proposed formal structures. One of the most well-known is that of Prieto J. (2012), who outlines an LO structure composed of four essential elements:

- **Verb:** expresses the action the student is expected to perform.
- **Knowledge object:** the content or subject matter being addressed.
- **Purpose:** the intended outcome or application of the learning.
- **Condition:** the context or criteria under which the learning will be developed or assessed.

LO can be formulated using the following structure:

[Verb] + [Knowledge object] + [Purpose 1 / Purpose 2 / …] + [Condition 1 / Condition 2 / …]

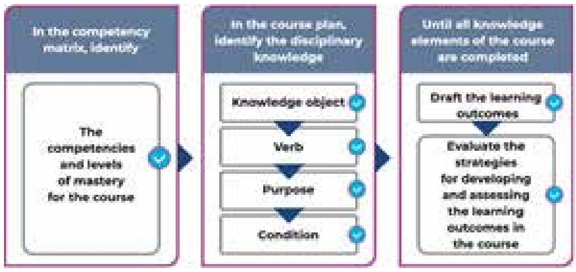

Fig. 1. *Learning Outcome Writing Process (Neil et al., 2023).*

Considering the previously established structure, it is pertinent to examine in greater detail the methodological process underlying the formulation of Learning Outcomes (LO). As illustrated in Figure 1, adapted from Neil et al. (2023), this process is iterative rather than linear and is typically organized into five key steps: identification of competencies and proficiency levels based on the course's competency matrix; definition of knowledge objects by selecting and grouping the subject's key contents; selection of an appropriate verb according to the cognitive level, following Bloom's taxonomy (1956); establishment of the purpose that the student must achieve; and definition of the conditions under which the knowledge is to be applied. This structured approach ensures that the resulting LO are aligned with graduation competencies, are measurable, and effectively guide both teaching and learning.

Automation Model Based on LLM and RAG

Given the challenges involved in manually writing LO, there is a clear need to automate this process. A viable alternative is the use of LLM, able to generate coherent text from prompts. This approach leverages the ability of LLMs to address complex tasks without requiring additional training or fine-tuning, thus simplifying their adoption (Yeung et al., 2025).

LLM belong to the field of generative artificial intelligence (Corchado et al., 2023), which specializes in producing content based on learned patterns. These models are trained on large volumes of data, enabling them to generate consistent responses by capturing semantic and contextual relationships in language (Ray, 2023).

To integrate relevant information into the model, the RAG technique can be employed. This architecture allows combining a LLM with an external knowledge base, thereby enhancing the quality of the generated responses (Posedaru et al., 2024). Contextual documents are converted into embeddings and stored in a vector database, from which relevant fragments can be retrieved to support the LLM's generation process.

Figure 2 illustrates the general architecture used, based on semantic storage and the retrieval of relevant data. This architecture has been widely adopted in various types of applications (Jeong, 2023; Li et al., 2023; Pavlyshenko, 2023). Once the system receives a prompt, it retrieves the semantically closest data, processes it with the LLM, and returns a proposal aligned with educational objectives.

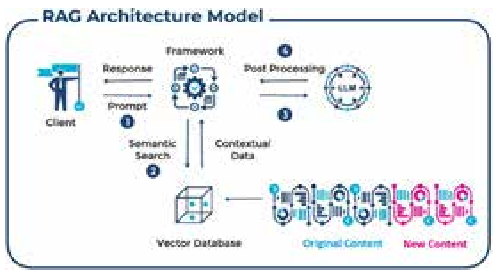

Fig. 2. *Proposed model for LO generation using the RAG architecture.*

To implement this workflow, open-source tools were used: LangChain (Langchain, 2025) to structure the RAG architecture; ChromaDB (Chroma, 2023) for embedding storage; and the Llama model (Meta, 2024) as the main LLM, all executed in a local environment. This configuration allows the incorporation of user-specific data not included in the original training of the model, thereby enhancing the system's effectiveness by extending its ability to process personalized information (Posedaru et al., 2024). The process, summarized in Figure 3, began with the selection of contextual documents, which were split into overlapping chunks and converted into embeddings stored in ChromaDB. Once prepared, the system was activated to process new prompts and generate learning outcomes based on the retrieved information.

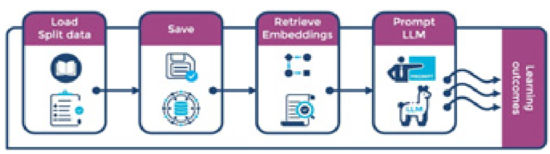

Fig. 3. *Automated generation process of learning outcomes using semantic search and structured data. Adapted from Posedaru et al. (2024).*

Model Selection and Technical considerations

This study opted for the Llama 3.1 model, a variant of LLM that can be executed in local environments. This choice is based on its design, which is optimized to function without relying on external servers, thus preserving data privacy and facilitating its implementation in academic or institutional contexts with infrastructure constraints (Corchado et al., 2023).

Although more powerful alternatives exist, such as Gemini or ChatGPT in their GPT-3 and GPT-4 versions, which incorporate a higher number of parameters (Ray, 2023), Llama highlights for its balance between performance and efficiency. It is designed to run on low-resource devices and has demonstrated adequate performance in various domains, such as the medical (Li et al., 2023) and financial fields (Pavlyshenko, 2023). These experiences show its adaptability to different contexts through adjustments in training and parameter tuning.

The selection of this model also responds to the philosophy of this work: to offer a scalable, controlled, and reproducible solution, capable of adapting to real-world educational environments without compromising the quality of the generated outcomes.

Integration of Documentary Inputs to Enrich the Model

To enhance the accuracy of the generated LO, the model's informational context was expanded through the incorporation of specific documents and previously validated examples. This strategy, aligned with the approach proposed by (Posedaru et al., 2024), seeks to optimize the quality of the generated outputs through the retrieval of contextual information.

The documents selected to be converted into embedding included the course syllabus, learning guides, guidelines for drafting LO, and a set of validated LO collected from both undergraduate and graduate programs. The latter was structured in a table with four columns—Course, Competencies, Skills, and LO—following the approach of Neil et al. (2023).

In total, 45 LO were collected from 18 graduate-level courses and 194 from 56 undergraduate courses. As suggested by (Benites et al., 2023), this integration of standardized inputs helps reinforce the coherence of the generated texts. Additionally, the impact of reducing the contextual input was evaluated, which, according to Yeung et al. (2025), is key to avoiding interference and ensuring more accurate generation aligned with curricular goals.

Operational Parameters of the Generative Model

Although this work does not explore in depth the implementation or coding aspects, it is relevant to outline the key configurations adopted during the generation process. This process begins with a dynamic prompt constructed using the following parameters:

- **Course:** corresponding disciplinary area.
- **Competency:** general competency associated with the curricular area.
- **Capability:** specific and observable subcomponent of competency.
- **Governing verb:** main cognitive action, defined based on educational taxonomies (e.g., Design, Analyze, Evaluate).

The model was configured to generate between two and four learning outcomes per execution. In its current version, each execution produces a single proposal per input, although the system allows multiple iterations when variations or alternatives are required.

Functional Evaluation of the Proposed Model

The evaluation of the proposed model required an experimental design that would allow for the analysis of its behavior across different generation scenarios. To this end, the methodological guidelines of Neil et al. (2023) were adopted as a reference, providing a framework for the elaboration of LO. Based on this foundation, different configurations were used concerning both the documentary inputs and the prompt formulation, with the goal of examining their impact on the generated LO. All tests were conducted within the same technological environment, consisting of the Llama 3.1 model, the ChromaDB vector database, and the LangChain framework. Throughout these configurations, the following curricular parameters were kept constant:

- **Course:** *Systems Analysis I*
- **Competency:** *Specify, design, and develop information systems.*
- **Capability:** *Identify and formulate information system problems.*
- **Proficiency level:** *The competence is addressed at proficiency level 1*.
- **Governing verb:** *Understand.*

A base prompt was created using the defined parameters and adapted into variants for each test scenario. This allowed analysis of the model's performance under different levels of contextual support while preserving core curricular elements. Although this article is in English, all source documents and prompts used were in Spanish, reflecting the native language of pedagogical materials in the educational institutions where the model is applied.

Scenario 1. Configuration with Full References

This test evaluated the model in a scenario with maximum contextual assistance, using a database generated from embeddings constructed on documents that included RA construction guidelines, the official course syllabus, learning guides, and a repository of learning outcomes previously validated by experts. The generation was executed with a temperature setting of 0 to ensure deterministic responses that were structurally coherent and aligned with the institutional pedagogical framework.

It is important to note that the temperature acts as a hyperparameter that regulates the degree of randomness in the selection of tokens during text generation. Low temperature values—such as 0 or 0.2—favor more controlled, stable, and formally consistent outputs, which are especially suitable for educational tasks that demand precision and uniformity (Radford et al., 2019). Conversely, higher values (such as 0.8 or 1.0) allow a greater linguistic variability and expressiveness, enhancing the creativity of the model (Zhao W. et al., 2023).

TABLE 1. PROMPT USED TO INTERACT WITH THE MODEL

You are an assistant specialized in higher education. You work with technology-related degree programs, specifically in Systems Engineering. *Your task is to generate between two and four Learning Outcomes for the following course, strictly adhering to the required structure* *### Parameters:* *- Course: Systems Analysis I* *- Competency: Specify, design, and develop information systems* *- Capability: Identify and formulate information systems problems* *- Governing verb: Understand* *### Instructions:* *Each Learning Outcome must follow exactly this format:* *{course}: [Verb] + [object of knowledge] + [purpose] + [conditions]* *- The **verb must be exactly** the one provided.* *- The LO must be clear, specific, and aligned with the given competency and capability.* *- The **format must be strictly observed**, without adding explanations or headings.* *### Example: Computer Networks:* *[Analyze] + [network protocols] + [to detect vulnerabilities] + [in simulated environments].* *### Task: Generate between two (2) and four (4) Learning Outcomes in the exact format specified above.*

Table 1 presents the prompt used to generate the LO, while Table 2 displays the outputs generated by the system under this configuration.

TABLE 2. SCENARIO 1. LO GENERATED FOR THE COURSE SYSTEMS ANALYSIS I

[Understand] + [information systems concepts] + [to identify design problems] + [in software development projects]
[Understand] + [systems analysis models] + [to evaluate the complexity of a system] + [in collaborative work environments]
[Understand] + [systems modeling tools] + [to represent relationships between components] + [in systems integration scenarios]

Regarding the generated LO, it was observed that the model correctly respected the established structure and that the produced texts exhibit a high degree of coherence..

Scenario 2. Model with Structural Reference, Excluding Previous LO Content

In this variant, the full corpus was retained except for the thematic content of the previous LO, which was used solely as a structural reference. To ensure this behavior, only the Instructions block from the original prompt (Table 1) was modified, explicitly stating that the linguistic structure of the stored LO (format, order, style) should be used as a reference, without reusing their thematic content. Additionally, it was established that information must be drawn exclusively from the official course syllabus and the included practical assignments. The final section of the prompt was also adjusted to reinforce this directive. This test allowed the isolation of the value of institutional thematic guidance compared to the use of existing LO. The results are presented in Table 3.

TABLE 3. SCENARIO 2. LO GENERATED FOR THE COURSE SYSTEMS ANALYSIS I.

[Understand] + [software architectures] + [to identify design patterns] + [in distributed systems]
[Understand] + [business models] + [to analyze system complexity] + [in contexts involving critical information management]
[Understand] + [functional and non-functional requirements] + [to formulate information system problems] + [in real-world scenarios]

Scenario 3. Execution Without Prior LO References

In the previous test, the model was instructed not to use the thematic content of the LO, but only their structure. However, this restriction may not have been sufficient to fully prevent their influence. Therefore, in this test, all previously generated LO were completely removed so that the model would work solely with theoretical guidelines, the course syllabus, and its learning guides. This setup allowed for the evaluation of the model's semantic autonomy in a scenario with reduced contextual support and helped avoid potential biases derived from previously generated outputs (Table 4).

TABLE 4. SCENARIO 3. LO GENERATED FOR THE COURSE SYSTEMS ANALYSIS

[Understand] + [information systems problems] + [to identify system needs and requirements] + [in real or simulated contexts]
[Understand] + [information system architectures] + [to design efficient and scalable solutions] + [considering factors such as security, performance, and maintainability]
[Understand] + [conceptual and logical models of information systems] + [to design and develop systems that meet user needs] + [using appropriate tools and techniques for modeling and analysis]

Scenario 4. Effects of Temperature Adjustment on the Generated Response

This test explored the effect of the model's temperature parameter on the linguistic variation and specificity of the generated LO. Configurations with temperatures of 0.8 (a balance between precision and diversity, Table 5) and 1.0 (greater creativity, Table 6) were compared. The results showed an increase in expressiveness without compromising structural integrity, yielding more flexible and richer responses that maintained academic coherence. This behavior is especially useful in exploration or creative contexts, where diversity in content formulation is valued (Zhao W. et al., 2023).

Table 5. Scenario 4-1. LO generated for the course Systems Analysis I. Temperature 1.0

[Understand] + [data structures] + [to identify complexity] + [in distributed systems]
[Understand] + [communication mechanisms] + [to evaluate efficiency] + [in computer networks]
[Understand] + [types of databases] + [to implement solutions] + [in web applications]

Table 6. Scenario 4-0. LO generated for the course Systems Analysis I. Temperature 0.8

[Understand] + [functionalities and requirements of the information system] + [to identify user problems and needs] + [in complex and dynamic systems]
[Understand] + [patterns and archetypes of information system problems] + [to apply existing software solutions] + [considering the system's scale and scope]
[Understand] + [functional and non-functional requirements of the information system] + [to define the boundaries and responsibilities of the development team] + [in collaborative projects with tight deadlines]

Conclusion and Future Work

The assisted generation of LO using a LLM within a RAG architecture proves to be an efficient and scalable strategy for automating the drafting of educational arti-facts. In multiple scenarios, the model produced coherent and well-structured LO, even amid contextual variability. A key finding was the absence of hallucinations, likely due to the contextual enrichment provided by RAG, which supports the reliability of the output. However, issues such as semantic overfitting and biases from non-diverse sources remain. Although the outputs align with curricular parameters, a systematic evaluation method is needed. Future work could focus on integrating validation rubrics, either as rule-based systems or embedded within the LLM, enabling autonomous decisions to regenerate, adjust, or approve the LO. This would foster iterative refinement and improve pedagogical robustness. The model also shows promises for generating other artifacts, such as rubric or assignments, though teacher involvement remains vital to ensure disciplinary relevance and educational integrity.

References

1. Benites, F., Benites, A. D., & Anson, C. M. (2023). Automated text generation and summarization for academic writing. *In Digital Writing Technologies in Higher Education: Theory, Research, and Practice* (pp. 279–301). Springer International Publishing. https://doi.org/10.1007/978-3-031-36033-6_18

2. Biggs, J. (2003). *Teaching for Quality Learning at University*.

3. Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Handbook I: cognitive domain*. (New York: David McKay, Ed.).

4. Chroma. (2023). *Chroma is the open-source search and retrieval database for AI applications*. *Https://Www.Trychroma.Com/*.

5. Chu, Z., Wang, S., Xie, J., Zhu, T., Yan, Y., Ye, J., Zhong, A., Hu, X., Liang, J., Yu, P. S., & Wen, Q. (2025). *LLM Agents for Education: Advances and Applications*.

6. Corchado, J. M., Sebastian López, F., Núñez, J. M. V., Raul Garcia, S., & Chamoso, P. (2023). Generative Artificial Intelligence: Fundamentals. *Advances in Distributed Computing and Artificial Intelligence Journal*, 12(1). *https://doi.org/10.14201/adcaij.31704*

7. Gaete Quezada, R. (2021). Evaluación de resultados de aprendizaje mediante organizadores gráficos y narrativas transmedia. *Revista de Estudios y Experiencias En Educación*, 20(44), 384–407. *https://doi.org/10.21703/0718-5162.v20.n43.2021.022*

8. Jeong, C. (2023). A Study on the Implementation of Generative AI Services Using an Enterprise Data-Based LLM Application Architecture. *Advances in Artificial Intelligence and Machine Learning*, 03(04), 1588–1618.

9. Li, Y., Li, Z., Zhang, K., Dan, R., Jiang, S., & Zhang, Y. (2023). ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge. Cureus. *https://doi.org/10.7759/cureus.40895*

10. Meta. (2024). *Introducing Llama 3.1: Our most capable models to date*. *Https://Ai.Meta.Com/Blog/Meta-Llama-3-1/*.

11. Min, B., Ross, H., Sulem, E., Veyseh, A. P. Ben, Nguyen, T. H., Sainz, O., Agirre, E., Heintz, I., & Roth, D. (2024). Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey. *ACM Computing Surveys*, 56(2), 1–40. *https://doi.org/10.1145/3605943*

12. Neil, C. (2024). *Marco conceptual para la definición, desarrollo y evaluación de competencias* (Fundación Iberoamericana de Estudios Superiores (ed.); 1a ed.). Fundación Iberoamericana de Estudios Superiores.

13. Neil, C., Battaglia, N., & De Vincenzi, M. (2023). La matriz de competencias como herramienta para orientar la escritura de resultados de aprendizaje. *XVIII Congreso Nacional de Tecnología En Educación y Educación En Tecnología-TE&ET 2023*.

14. Pavlyshenko, B. M. (2023). *Financial News Analytics Using Fine-Tuned Llama 2 GPT Model*.

15. Posedaru, B.-S., Pantelimon, F.-V., Dulgheru, M.-N., & Georgescu, T.-M. (2024). Artificial Intelligence Text Processing Using Retrieval-Augmented Generation: Applications in Business and Education Fields. *Proceedings of the International Conference on Business Excellence*, 18(1), 209–222. *https://doi.org/10.2478/picbe-2024-0018*

16. Prieto J. (2012). Las competencias en la docencia universitaria. *Pearson Educación*.

17. Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2019). *Improving Language Understanding by Generative Pre-Training*.

18. Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *In Internet of Things and Cyber-Physical Systems* (Vol. 3, pp. 121–154). KeAi Communications Co. *https://doi.org/10.1016/j.iotcps.2023.04.003*

19. Yeung, C., Yu, J., Cheung, K. C., Wong, T. W., Chan, C. M., Wong, K. C., & Fujii, K. (2025). *A Zero-Shot LLM Framework for Automatic Assignment Grading in Higher Education*.

20. Zhao W., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Wen, J.-R. (2023). *A Survey of Large Language Models*.