AutoFlash: Study Flashcard Generation Using Knowledge Graphs and Artificial Intelligence

Mariana Gouveia[1[53356]], David Guilherme[1[56333]], Guilherme Prazeres[1[64373]], and Duarte Gonçalves[1[64465]]

Faculdade de Ciências da Universidade de Lisboa, Lisboa, Portugal
https://ciencias.ulisboa.pt

Abstract. This paper introduces a novel approach to enhancing study tools by integrating Knowledge Graphs (KGs) and Artificial Intelligence (AI). Leveraging the structured representation of data in KGs, the proposed system automatically generates dynamic study flashcards to support efficient learning and review of complex topics. The process begins with the user selecting a theme or concept, after which relevant information is extracted from the Knowledge Graph. Using large language models (LLMs), this information is transformed into three types of flashcards: a summary flashcard that provides a concise overview of the topic, fact flashcards that present key pieces of information, and multiple-choice question flashcards designed to assess the user's understanding of the material. AI techniques are employed to enhance the clarity, relevance, and originality of the generated content. The system aims to offer an adaptive and user-friendly solution for producing personalized educational materials. Its scalability and flexibility allow for application across diverse subjects and domains.

Keywords: Knowledge Graphs · Flashcards · Artificial Intelligence · large language models (LLMs) · Educational Technology

1 Introduction

The exponential growth of information in academic and professional domains presents a significant challenge for learners attempting to acquire and retain knowledge efficiently. While traditional study methods such as note-taking, reading, and summarization are widely used, they often demand substantial manual effort and may not adapt well to diverse cognitive styles or learning needs. Flashcards, however, remain a well established and empirically supported tool, particularly effective for promoting active recall and spaced repetitions, two strategies known to enhance long-term memory retention [16,5].

Despite their pedagogical value, manually creating flashcards can be time-consuming and cognitively burdensome. This has led to increased interest in automated systems capable of generating personalized flashcards, thereby reducing learner workload and enhancing accessibility [9]. Such systems aim to transform raw data into structured, digestible formats that support focused and effective study.

In this work, we introduce a system for the automatic generation of educational flashcards using structured data extracted from open Knowledge Graphs (KGs), such as Wikidata [6]. KGs represent a rich source of semantic information, encoding relationships between entities in a machine-readable format [13]. Our system begins with a user-defined academic concept or theme and employs a Large Language Model (LLM) to expand it into related subtopics. Each subtopic is mapped to its corresponding entity in Wikidata using the Wikidata Search API. Relevant related informations are then retrieved through a SPARQL query based on these entity identifiers.

To ensure the educational relevance and clarity of the retrieved data, the system processes selected triples using an LLM that transforms them into natural language. The result is a structured JSON containing three types of flashcards: a single summary flashcard that offers a synthesized overview of the topic, several fact-based flashcards presenting concise factual information, and multiple-choice question flashcards designed to assess understanding through active engagement.

The integration of LLMs enhances the fluency, coherence, and contextual appropriateness of the generated content. Recent advances in natural language generation have shown that LLMs, when combined with structured data, can produce text that is both semantically rich and linguistically refined [10,17]. This synergy allows our system to bridge the gap between machine-readable knowledge and human-friendly educational materials.

This paper details the architecture and implementation of the proposed system, evaluates its capabilities, and discusses its potential to support scalable, adaptive, and intelligent learning environments. By combining semantic web technologies with state-of-the-art language models, the system contributes a novel approach to personalized and automated educational content generation.

2 Related Work

The integration of Knowledge Graphs (KGs) and Artificial Intelligence (AI), particularly Large Language Models (LLMs), in educational tools represents a growing area of interest in the research community. Existing studies have explored diverse aspects of this intersection, including automatic flashcard generation, the use of KGs for personalized learning, and AI-enhanced educational platforms. This section provides a comprehensive review of prior work across these domains, outlining both current advancements and unresolved challenges that inform the design of our proposed system.

2.1 Flashcard Generation

Flashcards are widely recognized for their utility in reinforcing learning through active recall and spaced repetition. Traditional tools such as Anki [1] rely on user-generated content, requiring learners to manually create question-answer pairs. While effective, this approach is labor-intensive and lacks scalability.

To address these limitations, recent research has explored automated flashcard generation. For instance, Cheng et al. introduced the WikiFlash system [2], which leverages natural language processing techniques to extract key concepts from Wikipedia articles and generate flashcards of varying complexity. WikiFlash integrates quality filtering mechanisms and supports multiple abstraction levels for users with different needs. User studies indicated that these automatically generated flashcards were comparably useful to those authored by humans. However, the system's reliance on Wikipedia raises concerns regarding content reliability, given the platform's open-edit nature and potential inconsistencies in information quality. While WikiFlash demonstrates scalability and personalization potential, it lacks integration with structured knowledge bases, limiting its ability to offer semantically coherent and curriculum-aligned study materials.

2.2 Knowledge Graphs in Education

Knowledge Graphs have emerged as a powerful representation for organizing and structuring domain knowledge in education. Their ability to model semantic relationships among concepts enables personalized and adaptive learning strategies.

Gu and Qianyi [8] proposed a system for recommending personalized learning paths using a KG constructed from dynamically aggregated data across various online educational environments. The system captures a learner's knowledge state and activities, enabling the delivery of contextually appropriate content tailored to individual learning goals. This semantic approach promotes intentional and effective learning by aligning recommended paths with users' evolving educational needs.

Similarly, Ramazanova et al. [14] provided a systematic classification of KG applications in education, identifying use cases for students, educators, and institutional stakeholders. The study highlighted how KGs facilitate the construction of individualized learning trajectories, content organization, and informed decision-making. Additionally, it surveyed techniques for KG construction, including machine learning and natural language processing, and emphasized the need for improved recommendation systems to better serve underrepresented educational populations. Despite their potential, current implementations often prioritize structural completeness over pedagogical usability, limiting their adoption in practice.

2.3 Artificial Intelligence in Educational Tools

The role of AI in modern educational environments is expanding, particularly in supporting Outcome-Based Education (OBE) through adaptive systems, real-time feedback, and automated assessment. Rani et al. [15] reviewed AI-driven platforms that facilitate performance monitoring, dynamic content delivery, and personalized learning. These systems allow educators to shift focus from administrative tasks toward individual learner support, improving alignment with OBE objectives.

In a qualitative study, Funda et al. [7] investigated student experiences with AI tools at a rural institution. Through focus group interviews, the authors identified themes including enhanced accessibility, personalized learning pathways, and increased motivation. Participants highlighted the benefits of AI in clarifying complex topics and enabling self-paced learning, particularly in under-resourced contexts. Nonetheless, concerns were raised regarding ethical implications, dependency on technology, and the erosion of traditional instructional practices.

Collectively, these studies demonstrate the transformative potential of AI in education, while also underscoring the necessity for context-aware, ethically grounded implementation strategies that balance innovation with established pedagogical principles.

2.4 Limitations of Existing Solutions

Despite notable progress in integrating AI and KGs into educational systems, several limitations persist across existing solutions.

In the domain of automated flashcard generation, systems such as WikiFlash [2] often produce content that lacks contextual nuance, resulting in superficial or ambiguous questions. The dependence on unstructured sources like Wikipedia further undermines the consistency and credibility of the generated materials.

KG-based educational applications, though promising, frequently suffer from underutilization and fragmented implementation. As noted by Ramazanova et al. [14], many KGs are designed with a focus on structural or technical completeness, rather than pedagogical integration or learner accessibility. Moreover, current systems seldom support the transformation of structured data into linguistically coherent, learner-friendly educational content.

Approaches such as Gu's learning path recommender [8] emphasize semantic alignment between learners and content but typically overlook the importance of adaptive language generation. This omission limits the effectiveness of these systems in engaging learners with diverse backgrounds and language proficiencies.

Similarly, while AI-enabled educational tools [15,7] have shown positive impacts on student engagement and personalization, they often prioritize analytics and delivery over content generation. Additionally, challenges related to transparency, bias, and over-reliance on AI remain unresolved, threatening the long-term sustainability of these solutions.

A pervasive limitation across these studies is the absence of a unified framework that integrates structured knowledge representations, personalized learning mechanisms, and high-quality natural language generation. This disconnection hampers the development of AI systems that are simultaneously scalable, adaptive, and pedagogically effective.

3 Methodology and System Overview

This section describes the architecture and workflow of the proposed system, which aims to generate personalized educational flashcards by leveraging structured data from knowledge graphs (KGs) and enhancing it through large language models (LLMs). The pipeline is composed of distinct but interdependent modules, ranging from user interaction and entity identification, to SPARQL querying, content generation, and final flashcard delivery.

Figure 1 illustrates the end-to-end workflow, from initial user input to final flashcard presentation. The system has a modular design, making it easy to incorporate future enhancements without requiring major structural changes.

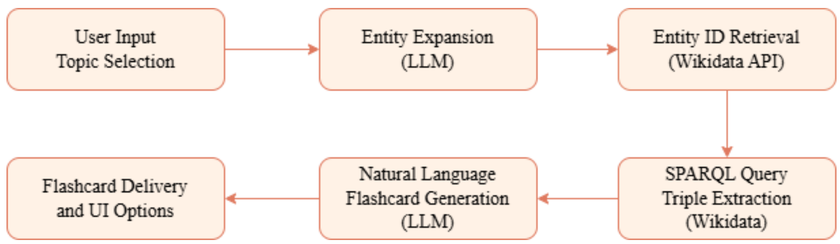


Fig. 1: System workflow from user input to flashcard generation [11].

To better understand the system's operation, the following subsections break down each stage represented in the diagram, from initial topic selection to the generation of refined educational content.

3.1 User Input and Topic Selection

The system begins with user interaction through a web-based search interface, where the user provides a topic of interest. This input is expected to be a general academic or conceptual theme, such as *"Knowledge Graphs"* or *"Deep Learning"*. To simplify testing and focus development efforts on output quality, no input validation mechanisms are implemented at this stage. Therefore, the user is responsible for ensuring the topic is semantically valid and coherent. This design choice ensures high-quality seeds for downstream knowledge extraction while maintaining development efficiency.

3.2 Entity Expansion via LLM

After receiving the topic, the system queries a large language model (LLM) through the OpenRouter API [12] to identify semantically associated concepts and subtopics. These returned entities represent more granular or related terms connected to the original topic, effectively broadening the scope of the query. For example, from an input like *"Deep Learning"*, the system might infer related entities such as *"Neural Networks"*, *"Backpropagation"*, or *"Gradient Descent"*. These serve as the knowledge anchors that guide subsequent information retrieval from the knowledge graph. Although the current version uses a freely available LLM, the architecture allows easy integration with more advanced or domain-specific models in future iterations.

3.3 Entity Identification via Wikidata API

To effectively query a knowledge graph like Wikidata [6], it is necessary to resolve each concept into its corresponding unique identifier (Q-ID). This step uses the Wikidata Search API [4] to retrieve the Q-IDs for each entity returned by the LLM, including the main topic. The result is a dictionary of entity–identifier pairs, which serves as the basis for constructing SPARQL queries. For example, the entity *"Knowledge Graph"* is resolved to the identifier `Q33002955` [3], enabling precise and unambiguous access to structured information.

3.4 SPARQL Triple Extraction

Using the retrieved Q-IDs, the system constructs a SPARQL query to extract relevant data from the Wikidata knowledge graph. For each entity, the system retrieves up to ten random **triples**, where each triple consists of a **subject**, a **predicate**, and an **object**. These triples encode semantically rich relationships, such as *(Deep Learning, subclass of, Machine Learning)* or *(Backpropagation, used in, Neural Networks)*.

To avoid repetitions of the same subject–predicate pairs with different objects, we apply the `DISTINCT` keyword to the SPARQL queries. This increases the diversity of the extracted knowledge by preventing redundant or overly similar results (e.g., multiple synonyms or closely related variants of the same concept). As a result, the system provides a broader and more informative foundation for generating educational content, encapsulating both factual knowledge and conceptual relationships.

3.5 Natural Language Flashcard Generation

Each triple obtained from the SPARQL endpoint is passed through the LLM, which reformulates the structured data into coherent natural language. For each triple, the model generates a single **summary** hat provides a brief and coherent explanation of the concept, a set of **fact statements** that highlight key pieces of information in a straightforward manner, and several **multiple-choice question** formed from the subject and predicate, accompanied by four possible answers, one of which is the accurate **response** based on the object. All generated content is stored in JSON format to be used by the user interface. This process transforms raw, machine-readable information into accessible educational material tailored for human consumption. The generation emphasizes fluency, contextual relevance, and pedagogical usefulness.

3.6 Flashcard Formatting and Output Delivery

Finally, the system organizes the generated content into three types of flashcards based on user preferences: summary cards, fact cards, and multiple-choice question cards. These flashcards are presented through a user-friendly interface that allows users to browse through different cards, request more examples, or export the summary and fact cards as images for external use and offline study. Users can also test their knowledge using the multiple-choice question cards, with the system providing feedback on correct answers and tracking their performance. If a user is unsatisfied with the flashcards content, the system can re-generate alternative versions by selecting and processing different triples related to the same topic.

4 Implementation

The proposed system is implemented using a modular client-server architecture designed to enhance scalability, maintainability, and flexibility. It consists of two main components: a backend knowledge engine that performs data processing and content generation, and a frontend interface that facilitates user interaction. This separation of concerns allows the system to efficiently manage computational loads and interface responsiveness while supporting parallel development and easier adaptation to future platforms.

The backend integrates external semantic knowledge sources and large language models (LLMs) via API calls to automatically generate enriched educational content. From a user's natural-language prompt, the system produces a curated set of structured facts, multiple-choice questions, and a contextual summary. This section details the system's architectural design, key technologies, and core implementation decisions.

4.1 System Architecture

The backend is implemented in Python and encapsulated within a class-based module named `KnowledgeEngine`, which orchestrates the complete workflow from prompt interpretation to educational content generation.

Upon receiving a prompt, the system invokes a large language model via the OpenRouter API to extract core thematic concepts. These concepts are then resolved to corresponding entities in Wikidata using the Wikidata Search API. For each resolved entity, dynamically constructed SPARQL queries retrieve relevant subject-predicate-object triples, which are subsequently transformed into readable facts. Finally, the system leverages the LLM to synthesize these facts into structured content, including multiple-choice questions and a narrative summary. Communication with external services is managed through RESTful API calls using the `requests` module, and a configuration file (`keys.json`) securely stores API credentials.

The frontend is developed using the Streamlit's framework and compiled as a web application. Users interact with a clean, responsive interface that includes an input field for custom prompts. Once the user submits a query, the prompt is transmitted to the backend via an HTTP POST request. Upon receiving the JSON-formatted response, the interface displays the generated content in three segments: a colorful carousel of facts with animated transitions, a set of interactive multiple-choice questions, and a stylized summary.

This architecture enables the backend to be deployed independently (e.g., as a cloud function or containerized microservice), while the frontend can be adapted across platforms with minimal effort. The decoupled design promotes parallel development, simplifies debugging, and ensures long-term scalability.

4.2 LLM Integration

Large language models (LLMs) play a central role in the system by enabling semantic interpretation and natural language generation across multiple stages of the backend pipeline. All interactions with the LLM are performed via the OpenRouter API using structured HTTP POST requests. Prompts are sent in natural language, containing the relevant information, and the LLM responds with a JSON-formatted message.

The first stage of LLM integration occurs during entity expansion. When a user submits a prompt and presses "Generate", the system sends the prompt to the LLM, requesting the identification and extraction of related thematic concepts. These are returned as a JSON list and subsequently used for entity ID retrieval via Wikidata.

Following the execution of SPARQL queries, the retrieved subject-predicate-object triples, while structurally informative, are often too abstract or fragmented for educational use. To address this, the LLM is again invoked during the Natural Language Flashcard Generation phase. At this stage, the LLM filters and transforms the triples into coherent, human-readable statements. It is instructed to ignore identifiers, metadata, references associated with copyright or templated classification systems, and to avoid tautologies or trivial assertions, that are typically not relevant to the users.

These refined triples are then used to generate three types of educational content, returned in a structured JSON format: factual statements, multiple-choice quiz questions, and a topic summary. For the quiz component, the system prompts the LLM to create up to ten multiple-choice questions. Each question is constructed using the subject and predicate of a triple, with the object serving as the correct answer. Additionally, the model generates three plausible distractors for each question. The factual statements are derived directly from the structured triples, while the summary is synthesized from the pool of facts and questions, offering a concise and pedagogically meaningful overview of the topic.

All LLM interactions are modularized within the `KnowledgeEngine` class, with each method encapsulating specific API calls and associated prompt engineering strategies.

4.3 Wikidata and SPARQL Integration

The system uses Wikidata as its primary structured knowledge base. Data extraction from Wikidata is performed using SPARQL queries submitted to the Wikidata Query Service (WDQS), and entity resolution is carried out through the Wikidata Search API.

Following theme extraction by the LLM, each identified concept, is submitted to the Wikidata Search API in order to retrieve the most relevant Q-ID (unique identifier). The request returns a list of candidate entities, from which the system selects the top-ranked result, typically the one with the highest relevance score or best match, and uses its Q-ID in subsequent SPARQL queries.

For each resolved Q-ID, a SPARQL query is dynamically generated to retrieve up to ten associated facts in the form of subject-predicate-object triples. The use of the `DISTINCT` keyword ensures that the query returns only subject–predicate pairs with different objects, avoiding redundant and very similar statements or values linked to the same property.

The retrieved data includes human-readable property labels (`propertyLabel`), their associated values (`valueLabel`), and brief descriptions (`valueDescription`), enabling the system to generate clear and informative educational statements. The triple results are parsed in JSON format and sent to the LLM for final refinement during the Natural Language Flashcard Generation phase.

Encapsulated within the `KnowledgeEngine` class, this querying mechanism supports modularity and reusability, forming the foundation for semantic data retrieval in the system.

4.4 Flashcard Export to Image

To support offline access and content sharing, the platform allows users to export flashcards as PNG images. This feature is integrated into the `Streamlit` interface, where flashcards are rendered using styled HTML.

When a user selects the export option, the HTML structure of the visible flashcard is captured and converted into an image using a browser-side rendering library. The image mirrors the on-screen appearance, preserving layout, colors, and typography, and is made available for download.

Export is enabled only for summary and fact flashcards, as these contain informative content suitable for reference. In contrast, question flashcards are excluded not only because their answer options are rendered outside the card layout, but also because their primary purpose is to assess whether users have retained knowledge from the informational flashcards. Exporting these would diminish their interactive and evaluative function.

5 Evaluation

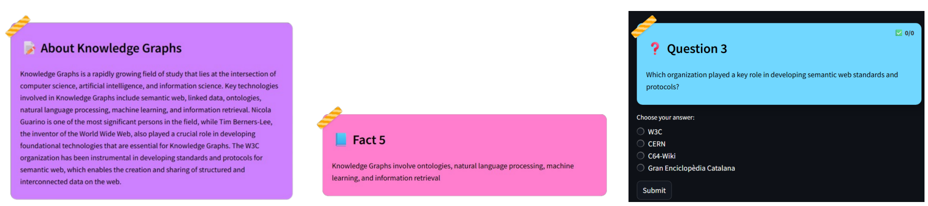
To assess the quality and usefulness of the flashcards generated by our system, we carried out an internal evaluation process. Although the ideal approach would involve end users interacting with the platform and rating each flashcard, practical limitations such as time constraints and the high computational cost of repeated model queries made a large-scale user study unfeasible.

Instead, we selected a representative sample of generated flashcards and had each group member independently evaluate them in isolation, without access to other member's scores, using a 1–5 Likert scale based on three criteria:

- **Clarity:** Assesses whether the flashcard is understandable, well-structured, and free from grammatical errors.
- **Relevance:** Measures how accurately the flashcard reflects the original knowledge base content, ensuring alignment with the subject matter.
- **Originality:** Evaluates the creativity of the flashcard in rephrasing or presenting information without merely copying the source material.

To better analyze the evaluations for each output, the average of the collected scores is calculated to produce a final score metric. This provided a preliminary yet structured assessment of the system's performance.

At the beginning of each round, a prompt is selected and all members evaluate the same set of generated flashcards, separately from each other. Each sample includes a Summary, two Fact flashcards, and two Multiple-Choice Questions, as illustrated in Figure 2.



(a) Summary (b) Fact (c) Multi-Choice Question

Fig. 2: Example of the Flashcard auto-generated

5.1 Results

We conducted three rounds of internal testing to assess the system's performance across diverse themes: *Knowledge Graphs*, *Deoxyribonucleic Acid (DNA)*, and *Portugal*. For each theme, the system generated a Summary, two Facts, and two Multiple-Choice Questions. Table 1 presents the average scores per element and theme.

Table 1: Evaluation of Summaries, Facts, and Questions

Theme	**Knowledge Graph**					**Deoxyribonucleic Acid**					**Portugal**				
Element	**Summary**	**Fact 1**	**Fact 2**	**Q1**	**Q2**	**Summary**	**Fact 1**	**Fact 2**	**Q1**	**Q2**	**Summary**	**Fact 1**	**Fact 2**	**Q1**	**Q2**
Clarity	4	5	5	5	3	5	4	3	5	2	3	2	5	5	3
Relevance	4	2	4	5	3	5	3	3	4	2	4	2	4	5	2
Originality	4	2	3	4	4	5	2	2	4	2	3	1	2	4	3

The results indicate that the system performs best on structured, technical topics; for instance, flashcards generated for *Deoxyribonucleic Acid* and *Knowledge Graphs* received consistently high scores, particularly in the Summary and Fact categories. These outputs were generally clear, accurate, and displayed thoughtful rewording of source material. However, performance declined for broader themes like *Portugal*, where flashcards were often generic or lacked sufficient context.

The underlying issue appears to be inconsistent knowledge extraction and shallow content generation when prompts are less constrained.

6 Discussion

This section outlines the system's limitations, possible future improvements, and summarizes the main contributions of the project.

Limitations: While the system successfully integrates knowledge graphs and large language models for flashcard generation, several limitations remain. First, the input mechanism relies on general topic descriptions, which may not support in-depth or highly specialized learning needs. The quality and diversity of generated subtopics heavily depend on the language model's output, which can lead to repetitive or thematically narrow results.

Additionally, since subdomain discovery and flashcard content are model-driven, the system lacks fine-grained control over conceptual coverage. This can result in uneven distribution of knowledge or redundancy across cards. The evaluation method is also subjective, relying on user feedback, which introduces variability and bias into quality assessment.

Future Work: Future improvements may include refining the user interface with advanced filtering options and interactive flashcard types, such as toggle interactions or spaced repetition formats. To reduce dependency on LLM-generated subtopics, hybrid approaches that incorporate domain-specific ontologies or structured taxonomies could guide entity expansion with greater consistency and coverage. Additionally, combining model outputs with curated knowledge bases could help ensure more diverse and pedagogically relevant content.

To strengthen evaluation, a more robust validation process involving unbiased participants and structured assessment rubrics would offer deeper insight into the system's educational effectiveness.

Summary: This project demonstrates a functional pipeline for generating educational flashcards by combining structured knowledge from Wikidata with LLM-driven natural language generation. The system transforms SPARQL-derived triples into accessible study content, illustrating how the integration of knowledge graphs and AI can support learners through an adaptive, data-enriched educational tool.

References

1. Anki: Anki. https://apps.ankiweb.net/ (nd)
2. Cheng, Y., Ding, Y., Foucher, S., Pascual, D., Richter, O., Volk, M., Wattenhofer, R.: Wikiflash: Generating flashcards from wikipedia articles. ETH Zurich (2020), https://tik-db.ee.ethz.ch/file/bec013d6c6ebd4f08580689ba5e3e5ca
3. Contributors, W.: Wikidata entity: Knowledge graph (2024), https://www.wikidata.org/wiki/Q33002955, accessed: 2024-05-14
4. Contributors, W.: Wikidata query service api (2024), https://www.wikidata.org/w/api.php, accessed: 2024-05-14
5. Dunlosky, J., Rawson, K.A., Marsh, E.J., Nathan, M.J., Willingham, D.T.: Improving students' learning with effective learning techniques: Promising directions from cognitive and educational psychology. Psychological Science in the Public Interest **14**(1), 4–58 (2013)
6. Foundation, W.: Wikidata: A free and open knowledge base (2025), https://www.wikidata.org, accessed: 2025-05-11
7. Funda, V., Bamigboye, O., Bamigboye, F.: Artificial intelligence: Experiences of students on ai tools usage for educational development. EDUCATIO Journal of Education **9**(2) (2024). https://doi.org/10.29138/educatio.v9i4.1585
8. Gu, Q.: Personalized learning path recommendation based on knowledge graph. Transactions on Social Science, Education and Humanities Research **8**, 2024 (2024), https://www.researchgate.net/publication/381451150_Personalized_Learning_Path_Recommendation_based_on_Knowledge_Graph
9. Kumar, R., Singh, J.P.: Automatic flashcard generation for effective learning – a survey. Education and Information Technologies **26**, 4703–4729 (2021)
10. Liu, W., Zhou, P., Zhao, Z., Wang, Z., Ju, Q., Deng, H., Wang, P.: Can knowledge graphs reduce hallucinations in llms? a survey. In: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) (2024), https://aclanthology.org/2024.naacl-long.219.pdf
11. Ltd, J.: draw.io: Diagrams for everyone (2025), https://app.diagrams.net/, accessed: 2025-05-14
12. OpenRouter: Openrouter api (2024), https://openrouter.ai, accessed: 2024-05-14
13. Paulheim, H.: Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web **8**(3), 489–508 (2017)
14. Ramazanova, V., Zagorulko, Y.A., Sambetbaeva, M.: Classification and analysis of the application of knowledge graphs in education. International Journal of Educational Technology **2**, 425–436 (2024), https://www.researchgate.net/publication/385868391_Classification_and_Analysis_of_the_Application_of_Knowledge_Graphs_in_Education
15. Rani, S., Kaur, G., Dutti, S.: Educational ai tools: A new revolution in outcome-based education. In: Explainable AI for Education: Recent Trends and Challenges, pp. 43–60. Springer (2024). https://doi.org/10.1007/978-3-031-72410-7_3
16. Roediger, H.L., Butler, A.C.: The critical role of retrieval practice in long-term retention. Trends in Cognitive Sciences **15**(1), 20–27 (2011)
17. Zhang, J., Zhang, B., Zhang, L.: Applications of knowledge graphs in natural language processing: A survey. IEEE Access **9**, 99999–100012 (2021)