SWInG: A Semantic Web Integrated with Generative AI Architecture for Dynamic Data Generation

Wagner Luis Cardozo Gomes de Freitas[1], **Jose Ricardo da Silva Junior**[1]

[1]Department of Computer, Federal Institute of Rio de Janeiro,
Eng. Paulo de Frontin, RJ, 26.600-000, BR

wgfreitas@gmail.com, jose.junior@ifrj.edu.br

***Abstract. Introduction**: This paper proposes an architecture that combines Semantic Web technologies and Generative Artificial Intelligence to automate the creation and maintenance of content data in digital games and other applications across configurable contexts. The architecture uses semantic databases as the primary data source and leverages Generative AI to fill in missing information and suggest corrections. **Objective:** The goal is a semi-automated architecture that reduces content curation workload while keeping data current and relevant. The approach also seeks to optimize Generative AI use, thereby reducing the financial costs of content generation. **Methodology**: The architecture utilizes semantic databases and Generative AI, integrating an AI confidence score mechanism to support content curation across various domains. An online quiz game was developed using Wikidata and OpenAI APIs as a proof of concept. **Results**: By extracting 1,241 values out of an expected 1,576 from the Semantic Web, the study identified 335 missing values and demonstrated how AI-generated content, accompanied by confidence scores, can effectively supplement these gaps. The results show that most AI-generated values have high confidence (above 86%), with certain properties reaching nearly 100%.*
***Keywords** Content Generation, Semantic Web, Generative Artificial Intelligence, Serious games.*

1. Introduction

Keeping information up to date – such as population statistics – poses a significant challenge. A few solutions involve integration with external databases, the use of APIs (Application Programming Interfaces) to access information in real-time, and web scraping techniques to extract data from websites [Arnab et al. 2015]. Other methodologies use artificial intelligence to automate data updates, including machine learning algorithms for identifying and correcting outdated information [Bellotti et al. 2013]. However, these solutions are often imperfect. Integrating external databases can be complex and demand high computational resources. Both the availability and amount of information can limit the use of APIs, as well as the context in which that information is used. Web scraping techniques are often fragile and susceptible to changes in website structure. Artificial Intelligence (AI) usage can be costly, requiring high-quality, pre-trained datasets. AI may also be paywalled by a user fee in plenty of cases. It is a premise resulting in significant financial, computational, and time costs.

This paper proposes an architecture combining the Semantic Web and Generative Artificial Intelligence to automate the acquisition and maintenance of application content data. The Semantic Web extends the World Wide Web to make data machine-understandable [Shadbolt et al. 2006]. It structures and interconnects information, facilitating data recovery and updates. Generative Artificial Intelligence, in turn, allows for the generation of content from machine learning models [Hat 2023]. Those can complement or enhance the data in question.

To evaluate the efficiency of the proposed architecture, we developed a quiz game. We used Wikidata [Vrandečić e Krötzsch 2014] for semantic data and OpenAI for unavailable or incorrect content [Ouyang et al. 2022]. The dynamic nature of the real world may require the precision and relevance of information, such as in serious games. Only by being accurate can they ascertain a significant, positive, and valuable learning experience for the user. Like book editions, educational games may lose relevance due to new findings and require updates. However, manual updates are expensive, error-prone, and compromise game quality and credibility. Results from our experiment show that combining the Semantic Web and Generative AI enables the creation of dynamic, up-to-date quizzes, where AI complements up to 21.18% of missing information from the Semantic Web.

The main contributions of this paper are: (i) a novel architecture, named SWInG (Semantic Web Integrated with Generative AI), capable of utilizing semantic web data while complementing and validating its information using Large Language Models (LLMs) in a customized manner. This approach addresses the challenge of maintaining up-to-date and accurate information in applications by combining the structured data from the Semantic Web with the generative capabilities of AI; (ii) a semi-automated approach that leverages the strengths of both semantic web technologies and generative AI, allowing for a configurable level of human intervention in the data through an information confidence analysis; and (iii) an evaluation of the proposed architecture through a quiz game. This practical implementation demonstrates the effectiveness of SWInG in generating and maintaining data content, showcasing its potential for educational and informational applications.

2. Background

2.1. Semantic Web

The Semantic Web, proposed by Tim Berners-Lee [Berners-Lee et al. 2001a], is an extension to the World Wide Web. It aims to provide a common structure for sharing and reusing data across different applications, companies, and communities. The Semantic Web's core idea is to enrich the web with metadata, making data comprehensible to both humans and machines. It is organized as a stack of layers [Horrocks et al. 2005], where each layer adds semantics to a piece of data. The Resource Description Framework (RDF) layer, according to Lassila et al. [Lassila et al. 1998], gains extra importance as it complements XML. It enables the specification of semantics to XML-based data in a standardized way. Then, RDF performs the data interchange as a structure of triples $P(x, y)$, where P, x, and y are the binary predicate, subject, and object, respectively. These structures are used to access other sites that contain relevant definitions or address other topics [van Ossenbruggen et al. 2001]. RDF organizes and describes information as subject–predicate–object triples that collectively form an RDF graph. Access to these graphs is provided through the SPARQL Protocol and RDF Query Language (SPARQL), whose specification defines both the query syntax and the formal semantics for RDF datasets. Such graphs may include Uniform Resource Identifiers (URIs) that reference other resources, as well as literal values. Another layer in the stack of the Semantic Web is the W3C Web Ontology Language (OWL)[1]. OWL is a language designed to represent rich and complex knowledge about data, data group, and their relationships. It is designed to enable computer programs to leverage their knowledge and verify consistency.

The Semantic Web offers benefits compared to the traditional web. When machines can interpret the meaning of data, professionals can create smarter, more efficient systems. A few examples include machine-performed tasks such as semantic search, personalized recommendations, heterogeneous data integration, and knowledge discovery. Moreover, interoperability between systems and applications improves the change and reuse of data in different contexts [Shadbolt et al. 2006]. Adding semantics to data scattered across the internet enhances its usability by applications. It creates useful dynamic relationships. However, the semantic template is not always compatible with a specific application. Additionally, these relationships are difficult to evaluate when considering only the response of the SPARQL query. Creating ontologies and annotating data in RDF demands effort, time, and resources. The technologies involved are also complex and not standardized, which prevents the Semantic Web from being used on a large scale [Berners-Lee et al. 2001b]. Applications of the Semantic Web span across various domains. In the field of education, for example, Semantic Web-based systems aim to provide personalized and adaptive learning experiences by using ontologies to model knowledge and student profiles [Elfotouh et al. 2017].

2.2. Generative Artificial Intelligence

Generative Artificial Intelligence is an artificial intelligence subfield that uses machine learning models to create content like text, images, audio, and video. Generative Artificial Intelligence utilizes deep learning models [Goodfellow et al. 2016], such as generative adversarial neural networks [Goodfellow et al. 2014] and generative pre-trained transformers [Clark et al. 2020], to generate data from existing large datasets. These models are trained to recognize patterns and connections within the initial data and then generate new content based on those patterns. Most existing models require high computational power and a high volume of data. Recently, Generative Artificial Intelligence has emerged in various fields, including the creation of chatbots, the generation of code, and the production of creative content. However, using Generative Artificial Intelligence presents challenges. It requires a substantial amount of training data, can generate false or misleading information, and raises ethical concerns related to authorship and the originality of the content it produces [Hat 2023]. In the context of dynamic content generation, Generative Artificial Intelligence offers significant potential. In digital games and other applications, the ability to procedurally generate content such as terrains, items, and narratives reduces reliance on preexisting data and enables the creation of richer and more varied user experiences [Summerville et al. 2018]. Due to the high computational costs associated with maintaining generative AI infrastructure, a portion of these expenses may be reflected in the application's operational costs, including potential indirect operational costs that may affect end users of the application. Despite notable advancements, the use of Generative Artificial Intelligence for dynamic

[1] https://www.w3.org/TR/sparql11-query/

content generation faces several challenges. Building high-performance generative models typically requires large volumes of training data, which may be scarce for highly personalized content [Mao et al. 2024]. Generative Artificial Intelligence represents a rapidly evolving field with transformative potential for dynamic content creation across various applications. It can learn complex patterns in data and generate new instances, paving the way for more flexible, adaptive, and content-rich systems. Integrating Generative Artificial Intelligence with other technologies like the Semantic Web may overcome current limitations and drive innovation in automated content generation.

3. Proposed Architecture

SWInG[2] uses a methodological approach to: (i) collect data from a semantic database; (ii) process and store the data in a local database; and (iii) improve and expand the collected data using artificial intelligence. The proposed approach aims to fill missing Semantic Web data and dynamically update it through a semi-automated process. In SWInG architecture, the *Semantic Open Knowledge Base* provides structured and interconnected data on a wide range of topics. It is the primary data source used by our approach to query and extract information, utilizing technologies such as RDF and OWL to present knowledge in a semantically structured manner. Collected data from the Semantic Web is stored in the *Semantic Database*. The *Semantic Database* is normalized for consistency and quality of information [Rashid et al. 2019]. The normalization process removes special characters, such as diacritics, converts data to a uniform format, and resolves ambiguities. The normalized data is saved to a local database (*Application DB*) for performance access and better information handling as the application runs.

After collecting and normalizing the data, the system highlights properties of interest for items that the query may not report, thereby addressing missing information. Missing values are stored in the *DB of Empty Value Properties*, which the *Generative Artificial Intelligence (GAI) Platform* uses to suggest answers based on contextual data. The *GAI DB* stores the suggestions generated by the *GAI Platform*. Figure 1 shows an overview of the SWInG architecture.

To streamline the validation process, the architecture incorporates a *confidence threshold mechanism*. If the *GAI Plataform*'s response has a confidence score above a predefined limit, the system automatically accepts the answer and adds it to the application's database. Otherwise, the information produced is sent to a *Technical Manual Validation* for analysis and, in case of approval, saved to the application database. This feature reduces the workload for technical validators while balancing automation and data quality. To be generic and accommodate different data domains, the *Query Setting* is a configurable script responsible for determining the context and properties of the data to be extracted from the Semantic Web. In this case, querying a specific domain requires modification of this script. Similarly, the *Prompts Configuration* is also a configurable script responsible for defining AI prompts used to fill the gaps left by the Semantic Web. Both must be configured while specifying a context for data generation, using a template as a base to ease this process.

In the current version of the architecture, we chose to use Wikidata as the Semantic Open Knowledge Base. Using qualifiers and a mechanism for classifying declarations

[2]SWInG source code: https://github.com/GSimCog/swing

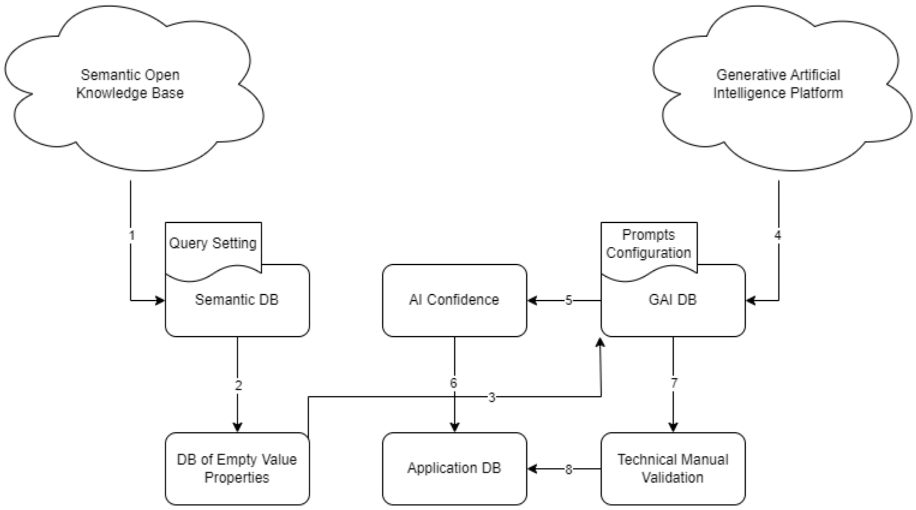

Figure 1. SWInG architecture for content generation using Semantic Web and Generative AI.

justifies that choice, as more than one value can exist for the same property. It allows for distinguishing the relevance and/or topicality of these declarations. For the *GAI Platform*, we chose OpenAI due to its easy integration, active ecosystem and community, and its flexible and adaptable characteristics. However, it is essential to note that alternative Semantic Open Knowledge Bases or GAI platforms can also be utilized, such as DBpedia[3] or Llama[4].

4. Evaluation

To evaluate the SWInG architecture, a multiple-choice online quiz game was created. We selected the definition of *country* as a concept, considering eight properties of a country, as outlined in Table 1. The data types include text, image, and audio, which were collected on May 21st, 2024. Figure 2 shows an example of the gameplay screen and final score. The user interface includes features to register and perform user login, as well as provide performance feedback.

The evaluated service-oriented architecture of the quiz is composed of two main parts. The frontend, developed in Flask, is responsible for the user interface, including the registration account page, user login page, quiz, results, and admin dashboard. Only technical users (human specialists) can access the admin pages. The backend manages game logic, data persistence, and communication with the SWInG architecture. The setting for the requests to the OpenAI API utilized the GPT-4o model, and the parameter *temperature* was set to zero. Lowering the temperature parameter toward zero makes a language model increasingly deterministic by pushing it to select the highest-probability tokens, which yields more predictable and coherent output. This setting is therefore preferred for applications that demand factual accuracy and minimal stylistic variation, such as structured question-answering or customer-service chatbots. Table 1 presents the prompts[5] for each property in this quiz game. Additionally, for each property, we

[3]https://www.dbpedia.org/
[4]https://www.llama.com/
[5]The full set of prompts can be checked at https://github.com/GSimCog/swing/tree/main/extra

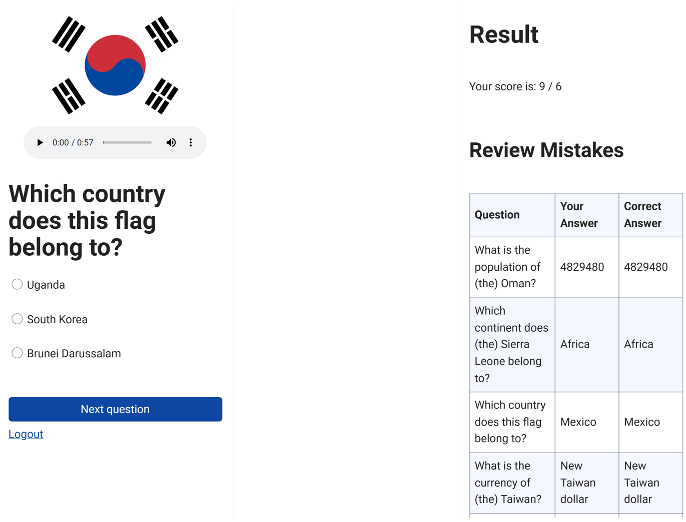

Figure 2. Example of gameplay screens. On the left is a quiz question with multimedia data, where on the right is the final score.

requested the confidence percentage, which ranged from 0 to 100. Besides that, except for *population, flag*, and *anthem*, we asked the prompt to add a separator in case of multiple answers.

At the start, the game verifies if the local database exists. That database stores the previously loaded and refined information that came from a query to the Semantic Web. If a local database is available, it is immediately loaded for faster access to the data. In case of negative conditions, the game consults the SWInG, and the SWInG will access the defined Semantic Open Knowledge Base through a SPARQL query. Then, it obtains the necessary dataset. The query returns the data, which is then normalized to populate the local database. During this normalization stage, the system identifies and registers the empty value properties. Based on the local database content, the system generates questions for the quiz. The user is presented with a random selection of questions covering various topics relevant to the respective context.

The quiz game frequently asks for new information in the SWInG. We update the local database if we identify altered property values, new countries, or countries that do not fit our new query. These updates are independent of previously performed value alterations based on AI suggestions. There are two justifications for this precedence: (i) despite being a public, collaborative project, Wikidata has mechanisms in place to ensure data quality, reliability, and accuracy. As for the OpenAI license agreement, it states the OpenAI system can make mistakes, and it is up to the user to fact-check important information; and (ii) the amount of edits per unit of time on Wikidata is currently expressive[6]. OpenAI models browse the web only within ChatGPT, not via the API we used. The model we used for this research, GPT-4o, received training up to October 2023. That said, the model is unable to respond with fresh data on population.

[6]Wikidata Dashboard: https://grafana.wikimedia.org/

Table 1. Properties and prompts.

Property	Prompt
capital	What is the capital of (the) <*country_name*>, without abbreviations, commas, or periods? Give me the answer as currently as possible with only the capital name.
currency	What is the currency of (the) <*country_name*>, without abbreviations, commas, or periods? Give me the answer as currently as possible with only the currency name.
population	What is the exact numeric population of (the) <*country_name*> in digits, without abbreviations, commas, or periods? Give me the answer as currently as possible in digits; only one number.
official language	What is the official language of (the) <*country_name*>, without abbreviations, commas, or periods? Give me the answer as currently as possible with only the language name.
continent	Which continent does (the) <*country_name*> belong to? Without abbreviations, commas, or periods. Give me the answer as currently as possible with only the continent name.
highest point	What is the highest point of (the) <*country_name*>, without abbreviations, commas, or periods? Give me the answer as currently as possible with only the highest point name.
flag	Provide one URL of the official flag as currently as possible of (the) <*country_name*>. This URL must refer to a web image file such as SVG, JPG, or PNG. Provide only the URL.
anthem	Provide one URL of the official anthem as currently as possible of (the) <*country_name*>. This URL must refer to a web sound file such as OGG. Provide only the URL.

5. Results

The quizzes were up-to-date with Semantic Web content, and querying Wikidata resulted in extracting 1,241 properties using the *country* as context. The initial data evaluation revealed that some values for properties relevant to the study were missing from Wikidata.

Disregarding conceptual specificities in the definitions of terms like *country* and *sovereign state*, that query returned data on 197 countries. In our context, a *country* refers to a modern, present-day state with recognized borders and a government. They are typically United Nations members or affiliated with other international organizations. The term *sovereign state* represents states that possess complete autonomy and independence, such as modern nation-states. Although all sovereign states are countries, not all countries are sovereign states, as several countries are dependent territories. That way, the total amount of property values of interest corresponds to 1,576 (197 countries $\times$ 8 properties). If we subtract the number of expected values from Wikidata from our actual result, it represents 335 missing values, which OpenAI suggests. That semi-automatic feature can complement missing information in the semantic database, speeding up the update and maintenance of the local database. Considering the gaps in information unavailable on the Semantic Web (Wikidata, in this case), using OpenAI aims to reduce the workload during the data curation stage.

Confidence accompanies all information produced by OpenAI. Figure 3 shows statistical information regarding all these missing 335 values, grouped by each property. According to this figure, it is possible to observe no missing value for *capital*. For all the missing values, confidence is over 86%. Besides that, the *population* property is the most sensible, as this information can change more frequently. Additionally, it can be observed that OpenAI's confidence is less than 96% for a country *flag* image. The *official language* confidence mean (green triangle mark) is about 98% with a median (orange line in the box) of 100%. On the other hand, the *highest point* of a country has a median of 90% and a mean of approximately 93%. According to the image, the most confident property provided by OpenAI is the *currency* and *continent*. With a confidence threshold of about 90%, some AI-provided information requires manual administrator approval, but the majority is added directly to the database, filling gaps the Semantic Web could not provide. This approach significantly reduced the workload of technical users while maintaining data quality.

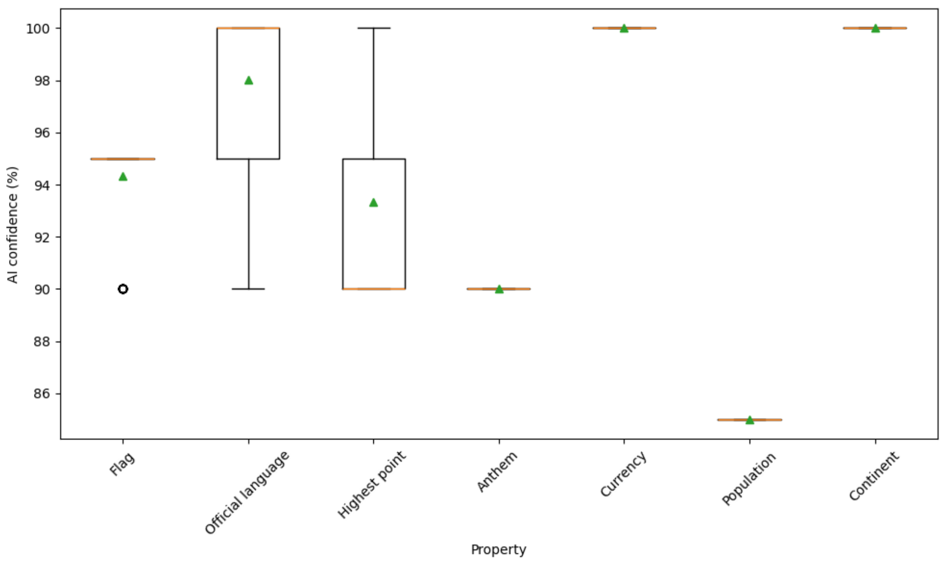

Figure 3. Statistical information for OpenAI confidence.

Additionally, we examined the performance of its queries to Wikidata. This examination linked the number of returned lines against time and the length of the answer. We added the clause $LIMIT(\alpha)$ (number of lines returned) to the query, where $\alpha \in \{1, 5, 10, \ldots, 260, \infty\}$. This evaluation took place between March 12nd and March 18th, 2024, through 1,512 requests to Wikidata. For $\alpha = \infty$, a total of 258 lines were obtained, containing the entire dataset of interest on Wikidata at the moment of the evaluation. The discrete and empirical part of this test is on Figure 4. Despite the lack of a straightforward relation between *Response Time* and *LIMIT*, we observed an average of 2.59 seconds per answer. Besides that, it suggests that limited queries offer little gain when comparing the time using $LIMIT(1)$ (2.80 seconds) and $LIMIT(\infty)$ (3.08 seconds). We assume these time values differ due to the network infrastructure at the time of the query. As time does not vary significantly with the α value, it may be more efficient to perform one single long query, instead of many queries with a limited number of lines.

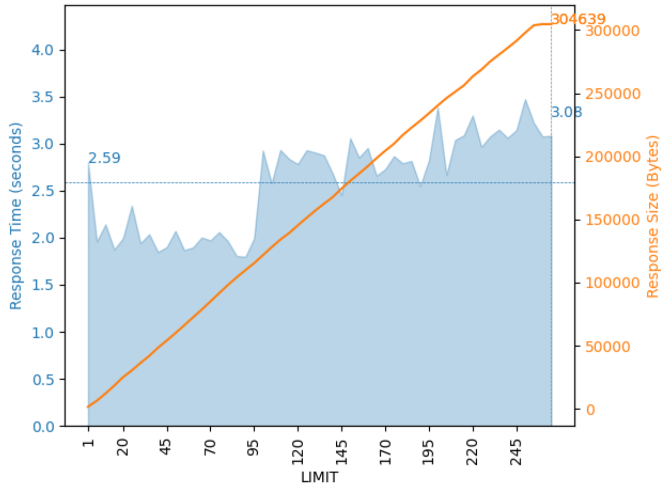

Figure 4. Response Time and Size values for specific numbers of returned rows.

Another piece of evidence to highlight in Figure 4 is the *Response Size* for queries where $\alpha > 258$, which returns the entire dataset for this evaluation. In those cases, the amount of data returned is 304,639 bytes (0.30 M bytes). Specifically for the queries in this study, we were able to meet the non-functional bandwidth requirement to create the local database, which caused no performance or scalability difficulties. However, the flag image relies on the user's internet connection, as it is downloaded and read from the Wikidata repository, which can slow down the application.

6. Threats to Validity

Absent our efforts to mitigate the threats to validity, we identified internal ones that could impact the results. First, data from Wikidata and OpenAI did not undergo human verification for accuracy, as no qualitative analysis was conducted. In this case, even if the *GAI Platform* fills in the empty values, the data quality and accuracy may be compromised. We strive to ensure data quality assurance by implementing a threshold mechanism for confidence. This approach enables a balance between automation and human oversight, tailoring the data quality requirements for each type of application. For example, general quizzes can use lower thresholds than specialized scientific ones. However, it is essential to note that this level of confidence is a reflection of OpenAI's confidence in its results.

As presented in Section 4, the OpenAI model used in this project was trained up to October 2023, whereas data on Wikidata is continually edited. To minimize that threat, we performed manual validation of the elements suggested by OpenAI in the local database. That way, we eliminate inaccurate information before it reaches the user. The results of this evaluation may not be generalized to other contexts or the Semantic Open Knowledge Base. This research focused solely on Wikidata and OpenAI, disregarding other data sources. We considered only the context *country* and eight properties, and as such, the OpenAI performance may vary depending on model updates or changes in the training data. The efficiency of those suggestions may be inconsistent in the long term. To mitigate this, a future approach may utilize multiple sources of semantic data, covering specific domains of interest. Our approach is designed to be extensible to new knowledge domains: practitioners need to redefine the SPARQL query and adjust the AI prompts that fill in missing data. These changes, however, still demand a moderate level of technical expertise in both semantic-web querying and prompt engineering. Error-mitigation hinges on the level of trust placed in an LLM's output. While hallucinations remain a potential threat to validity, we curtailed this risk by iteratively designing and vetting prompts against ground-truth answers. The onus ultimately lies with the practitioner to craft clear, context-rich instructions that maximise coherence and accuracy, thereby further reducing hallucination rates. Dependence on OpenAI's proprietary models poses sustainability risks: future work could be disrupted by pricing or quota changes, and the October 2023 knowledge cut-off may distort dynamic data such as current population counts. To reduce vendor lock-in and ensure reproducibility, we consider benchmarking openly licensed LLMs such as Meta Llama 3. Cut-off limitations can be mitigated through retrieval-augmented generation pipelines, which fetch up-to-date information at inference time.

7. Conclusion

We presented an approach for producing and keeping up-to-date data based on the Semantic Web. When data is missing or multiple answers are found, artificial intelligence is applied to fill such gaps. The results show that integrating the Semantic Web and Generative Artificial Intelligence technologies may enhance the completeness and accuracy of content data in applications. Specifically, the use of Generative Artificial Intelligence enabled the fulfillment of all empty properties (21.18%) of data from the Semantic Web. That highlights the potential of Generative Artificial Intelligence as a complement to the data available on the Semantic Web.

Despite promising results, it is worth noting that our evaluation was exclusively quantitative. Our analysis focused on measuring the number of properties filled by Generative Artificial Intelligence, without a qualitative check of the accuracy and relevance of those suggestions. Therefore, the results provide an initial overview of the positive impact this technology may have, but they cannot assure the truthfulness or quality of that information. We mitigate that issue through a semi-automated mechanism based on data confidence. Data confidence higher than a threshold is updated directly to the application database, while data below this threshold undergoes technical user approval. For future work, performing qualitative information analysis is desired, both through the Semantic Web and Generative Artificial Intelligence. That analysis ought to include an accuracy check for the suggestions. Another proposal is to submit post-approval technical users' data to Wikidata, ensuring more reliable information in the future. Finally, future studies could explore the implementation of automated data validation mechanisms by employing multiple AI models to verify their convergence in the results. SWInG lays the groundwork for hybrid systems that continually refine content by merging semantic data with Generative AI.

Acknowledgments

The authors would like to thank CAPES, CNPq, and FAPERJ for the financial support.

References

Arnab, S., Lim, T., Carvalho, M. B., Bellotti, F., de Freitas, S., Louchart, S., Suttie, N., Berta, R., e De Gloria, A. (2015). Mapping learning and game mechanics for serious games analysis. *British Journal of Educational Technology*, 46(2):391–411. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/bjet.12113.

Bellotti, F., Kapralos, B., Lee, K., Moreno-Ger, P., e Berta, R. (2013). Assessment in and of Serious Games: An Overview. *Advances in Human-Computer Interaction*, 2013(1):136864.

Berners-Lee, T., Hendler, J., e Lassila, O. (2001a). The Semantic Web. *Scientific American*, 284(5):34–43.

Berners-Lee, T., Hendler, J., e Lassila, O. (2001b). The semantic web: A new form of web content that is meaningful to computers will unleash a revolution of new possibilities. *ScientificAmerican.com*.

Clark, P., Tafjord, O., e Richardson, K. (2020). Transformers as Soft Reasoners over Language. arXiv:2002.05867 [cs].

Elfotouh, A. M. A., Nasr, E. S., e Gheith, M. H. (2017). Towards a comprehensive serious educational games' ontology. In *Proceedings of the 3rd Africa and Middle East Conference on Software Engineering*, AMECSE '17, page 25–30, New York, NY, USA. Association for Computing Machinery.

Goodfellow, I., Bengio, Y., e Courville, A. (2016). *Deep Learning*. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., e Bengio, Y. (2014). Generative Adversarial Nets. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., e Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Hat, R. (2023). What is generative AI? https://www.redhat.com/en/topics/ai/what-is-generative-ai. Accessed: Apr. 04, 2024.

Horrocks, I., Parsia, B., Patel-Schneider, P., e Hendler, J. (2005). Semantic web architecture: Stack or two towers? In Fages, F. e Soliman, S., editors, *Principles and Practice of Semantic Web Reasoning*, pages 37–41, Berlin, Heidelberg. Springer Berlin Heidelberg.

Lassila, O., Swick, R. R., et al. (1998). Resource description framework (rdf) model and syntax specification.

Mao, X., Yu, W., Yamada, K. D., e Zielewski, M. R. (2024). Procedural Content Generation via Generative Artificial Intelligence. _eprint: 2407.09013.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., e Lowe, R. (2022). Training language models to follow instructions with human feedback. _eprint: 2203.02155.

Rashid, M., Torchiano, M., Rizzo, G., Mihindukulasooriya, N., e Corcho, Ó. (2019). A quality assessment approach for evolving knowledge bases. *Semantic Web*, 10(2):349–383.

Shadbolt, N., Berners-Lee, T., e Hall, W. (2006). The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101.

Summerville, A., Snodgrass, S., Guzdial, M., Holmgård, C., Hoover, A. K., Isaksen, A., Nealen, A., e Togelius, J. (2018). Procedural Content Generation via Machine Learning (PCGML). _eprint: 1702.00539.

van Ossenbruggen, J., Hardman, L., e Rutledge, L. (2001). Hypermedia and the semantic web: A research agenda. *Journal of Digital information*, 3(1).

Vrandečić, D. e Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.