AutoVU-KG: Automated Validation and Updates for Knowledge Graphs with Web-Search-Augmented LLMs

Amel Gader and Alsayed Algergawy
Chair of Data and Knowledge Engineering, University of Passau, Passau, Germany

Keywords: Knowledge Graphs, Web-augmented LLMs, KG Update.

Abstract: Knowledge Graphs (KGs) offer a powerful framework for representing and managing structured information in many applications. However, when it comes to frequently changing facts, KGs often lag behind real-world updates. Large Language Models (LLMs) hold promise for enriching and updating KGs, but their capabilities are limited by static training cutoffs and a tendency to hallucinate or produce outdated information. To address these concerns, we introduce AutoVUKG: **Auto**mated **V**alidation and **U**pdates for **K**nowledge **G**raphs with Web-Search-Augmented LLMs. Our approach comprises: a classification module that identifies facts likely to change and therefore needing updates; An LLM-driven validation and update pipeline, enhanced with real-time web retrieval to ground assertions in current external sources, and an entity matching and alignment component that ensures updates maintain internal consistency within the KG. Evaluation on subsets of Wikidata demonstrates that the proposed approach achieves high accuracy and significantly outperforms vanilla LLMs. Additionally, it reduces the number of outdated facts by up to 60% on one of the datasets. The source code is available at https://github.com/amal-gader/autovu-kg.

1 INTRODUCTION

A Knowledge Graph (KG) is a structured data model used for knowledge representation and organization. It serves as the backbone of web-scale knowledge and supports a variety of downstream applications, such as recommender systems, question answering, and information retrieval (Peng et al., 2023). To ensure these applications produce accurate and relevant results, it is essential that the underlying KGs remain high-quality and are continuously updated, especially given the rapid pace at which information evolves and the increasing dependence on KGs across domains.

One of the most prominent open-source KGs is Wikidata (Vrandečić and Krötzsch, 2014), a huge, free knowledge base that is built and edited collaboratively through crowdsourcing and community-driven contributions. Wikidata supports applications like Wikipedia[1] and many semantic web tools. Despite its scale and utility, Wikidata often contains outdated information, largely due to its reliance on manual updates by users. With millions of entities and facts, many of which are dynamic in nature, manual maintenance becomes increasingly impractical(Shenoy et al., 2022).

A dynamic fact refers to a piece of information that changes over time such as the current president of a country or an organization, the stock price of a firm, or the list of drugs used for a specific treatment. In contrast, static facts, like the birthplace of a football player or the capital of a country, remain generally unchanged over time. Dynamic facts are especially common in financial and medical domains, where updates may be required on a weekly, daily, or even real-time basis.

Since their emergence, Large Language Models (LLMs) including closed-source models such as GPT-4[2], and open-source models like LLaMA (Touvron et al., 2023), and DeepSeek[3] have shown impressive performance across a wide range of Natural Language Processing (NLP) tasks. Their integration into tasks such as KG construction, completion, and refinement has shown great promise (Zhu et al., 2024). The synergy between KGs and LLMs has been explored bidirectionally, with growing focus on LLM-enhanced KGs, where LLMs are used to enrich, verify, or complete knowledge representations (Agrawal

[1] https://en.wikipedia.org/wiki/Main_Page

[2] https://platform.openai.com/docs/models/gpt-4.1

[3] https://www.deepseek.com/

Gader, A. and Algergawy, A.
AutoVU-KG: Automated Validation and Updates for Knowledge Graphs with Web-Search-Augmented LLMs.
DOI: 10.5220/0013704600004000

In *Proceedings of the 17th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2025) - Volume 1: KDIR*, pages 257-265

et al., 2023; Feng et al., 2023; Wei et al., 2024; Yao et al., 2025).

However, LLMs also come with notable limitations. They are prone to hallucinations, generating false information and are constrained by a fixed knowledge cutoff, which limits their usefulness in dynamic or time-sensitive applications (Liu et al., 2024; Mousavi et al., 2024; Sriramanan et al., 2024). These limitations reduce their reliability as autonomous agents for updating and maintaining KGs, Figure 1 depicts an example of an outdated dynamic fact on *Wikidata* which an LLM may not be able to update.

To address this challenge, Retrieval-Augmented Generation (RAG) systems have emerged as a hybrid solution. RAG architectures combine the generative power of LLMs with real-time retrieval from external, up-to-date sources such as web search engines, document databases, or structured repositories to mitigate knowledge gaps (Asai et al., 2024; Xie et al., 2024).

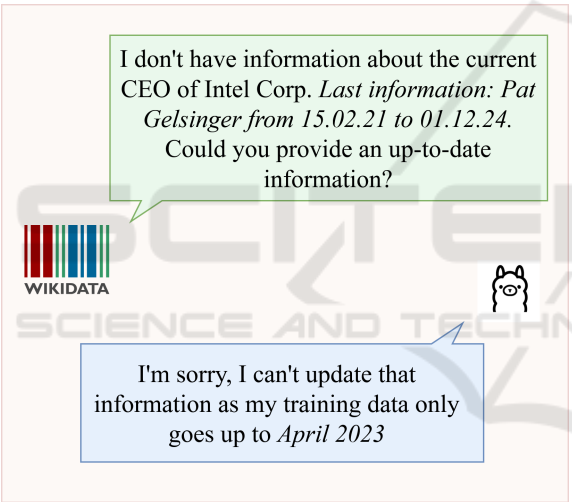

Figure 1: A motivating example illustrating the limitations of LLMs in handling outdated facts in knowledge graphs. The prompt was submitted to LLaMA3-70B, DeepSeek, and GPT-3.5 at the time of the experiment, and all models returned similar responses. (Note: Answers may differ if models have since been updated.)

Building on this idea, we propose a framework that leverages LLMs augmented with web-retrieved information to check, validate, and update outdated or erroneous facts in KGs. In our approach, the LLM acts as a reasoning agent, resolving conflicts that may arise in retrieved data by analyzing associated metadata such as timestamps, titles, and source credibility. Our main contributions are:

- We present a lightweight, practical framework, with publicly available code for the automated validation and updating of dynamic facts in encyclopedic knowledge graphs such as Wikidata.
- We provide a set of SPARQL queries to extract time-sensitive facts, which serve as benchmarks for evaluating update methods.
- We evaluate the framework on these benchmarks and conduct a comparative analysis of different model configurations in terms of update accuracy and efficiency.

2 RELATED WORK

In this section, we review the most relevant research related to our study. First, we examine works that explore the integration of web search with Large Language Models (LLMs). Next, we discuss previous efforts aimed at automating the process of Knowledge Graph (KG) updating.

2.1 Web-augmented LLMs

Multiple studies have aimed to train LLMs to mimic the human-like web search behavior. One of the earliest efforts is WebGPT by OpenAI (Nakano et al., 2022), which introduced a web-browsing environment for a fine-tuned version of GPT-3. The model was trained using two types of data: demonstrations; consisting of human-generated web search sessions used during supervised learning, and comparisons; which involve human feedback on model-generated answers and are used in the reinforcement learning phase to optimize performance.

Two other approaches follow a similar paradigm. WebGLM (Liu et al., 2023) builds on the WebGPT framework by using a more efficient model architecture and replacing costly human feedback with user likes from online Q&A as a quality signal. Similarly, AutoWebGLM (Lai et al., 2024) adopts a two-step pipeline: an interaction step, where information is retrieved from the web, followed by an action step, in which the language model generates a response to the query. Its training also relies on reinforcement learning.

UNI-WEB (Li et al., 2023) belongs to the same line of research on web-enhanced LLMs. A key strength of this work is the introduction of a self-assessment mechanism that allows the model to evaluate its confidence in its own answers, with uncertainty quantified using entropy. When the confidence is low, it queries web search APIs to retrieve additional information and improve its answers.

2.2 Automate the Update of KGs

Extensive research has been dedicated to augmenting LLMs with web search; however, fewer studies have addressed the challenge of automating updates to knowledge graphs.

A notable study is (Tang et al., 2019) which aims to leverage continuous news streams to dynamically update and enrich KGs. The proposed approach involves training an encoder-decoder model, where the encoder integrates relational graph attention mechanisms with text-based attention to ensure the message from the news snippet pass along the KG structure. The encoder is used to generate representations for entities, and for each entity pair, a multi-layer perceptron classifier evaluates the probability of a new link being added or an existing link being removed. To determine the specific type of relation between entities, the model employs DistMult (Yang et al., 2015) as the decoder.

Another related work is (Babaiha et al., 2023); it focuses on enriching biomedical kGs by automatically extracting causal relationships from biomedical literature. The approach begins with a keyword-based search of PubMed[4] abstracts using search APIs to retrieve relevant literature. Information extraction is then performed on the collected abstracts, followed by the training of an NLP-based extractor relying on named entity recognition and relation extraction techniques to identify meaningful biomedical entities and their interactions. Extracted relations are evaluated by human experts.

Both methods start with information retrieval from external sources and update the KG when a relevant change is detected. However, our goal is to build a system capable of proactively identifying and updating all potentially outdated facts in the knowledge graph.

We plan to merge the goals of the aforementioned works by updating KGs with web-augmented LLMs. We are aware of a related work that shares the same objective (Hatem et al., 2024), but we are taking a different approach for knowledge retrieval and exploring additional aspects.

3 METHODOLOGY

In this section, we define the task and detail the different steps of our framework.

[4]https://pubmed.ncbi.nlm.nih.gov/

3.1 Preliminaries

Knowledge Graphs (KGs) evolve continuously over time. As noted by (Polleres et al., 2023), this evolution can be analyzed by treating time either as *data* (explicitly encoded in the triples) or as *metadata* (attached to the triples externally).

3.1.1 KG Evolution Dimensions

Temporal Knowledge Graphs explicitly incorporate time into the triple structure, typically represented as quadruples (h, r, t, τ), where h, r, and t denote the head entity, relation, and tail entity of the factual triple, and τ represents the associated timestamp or time interval that indicates the validity period of a fact.

In contrast, *Time-varying Knowledge Graphs* represent temporal information as metadata, indicating the transaction time or the time of insertion, rather than embedding it within the triple itself. Time-varying KGs can be further categorized into:

- **Dynamic KGs**, which preserve the full history of changes as a set $\mathcal{G} = \{(h, r, t, \tau_i) \mid \tau_i \in \mathbb{T}\}$, where τ_i is a metadata field representing the time at which the triple was added or modified.
- **Versioned KGs**, which store discrete snapshots of the graph at specific time points as $\mathcal{G} = \{G_{t_1}, G_{t_2}, \ldots, G_{t_n}\}$.

3.1.2 KG Update Paradigms

To update a KG, different strategies can be employed. For instance, *Wikidata* primarily relies on community-driven contributions, where users manually edit facts. Bots can assist with certain routine tasks (e.g., property validation), but a fully automated update mechanism is not in place. We propose a framework in which Web-augmented Large Language Models (LLMs) automatically perform KG updates by retrieving, validating, and integrating new facts from external sources.

Given an incoming new fact (h, r, t_1) observed at time τ_1, the update strategy depends on the KG type:

- For **dynamic KGs**, the system retains historical information. If a prior version (h, r, t_0) was added at τ_0, the updated KG becomes:
$$\mathcal{G}_{\tau_1} = \mathcal{G}_{\tau_0} \cup \{(h, r, t_1, \tau_1)\}$$
- For **versioned KGs**, older facts are replaced in the new snapshot. The KG at τ_1 is updated as:
$$\mathcal{G}_{\tau_1} = (\mathcal{G}_{\tau_0} \setminus \{(h, r, t_0)\}) \cup \{(h, r, t_1)\}$$

In this work, we assume a versioned KG setting. The dynamic scenario can be addressed analogously, differing in that new facts are created with associated timestamps rather than removing outdated entries.

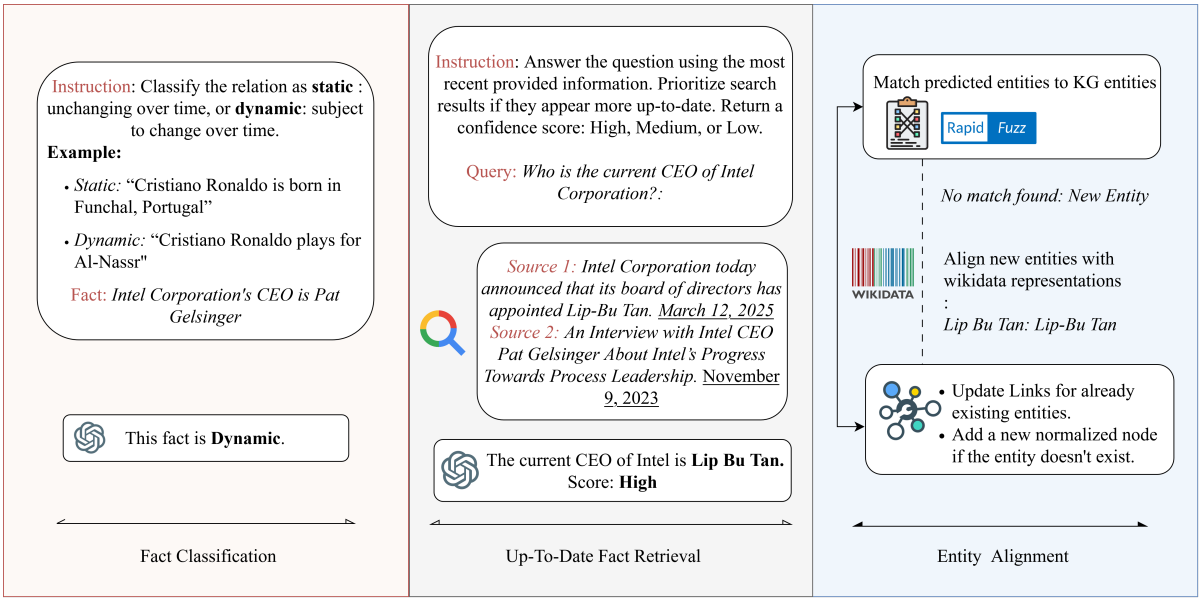

Figure 2: AutoVU-KG: Overview of the Proposed Framework. The pipeline consists of three main components: (1) classification of facts as static or dynamic, (2) extraction of up-to-date facts from the web, and (3) alignment and integration of entities into the KG.

3.2 AutoVU-KG: Proposed Framework

Our overarching goal, as aforementioned, is to propose a robust framework for validating and updating facts in Knowledge Graphs using a web-augmented LLM-based approach. The process begins with the classification of facts into static and dynamic categories. We then extract a subgraph containing only dynamic facts. For each of these, we retrieve relevant context from the web and attach appropriate metadata. This information is incorporated into the LLM prompt to assess or generate updated facts. Finally, newly predicted entities and relationships are integrated into the Knowledge Graph, and existing links are updated accordingly. Figure 2 describes the main steps of the pipeline.

3.2.1 Classification Module

Knowledge Graphs (KGs) often contain both static and dynamic facts, with the latter being subject to change over time. The goal of our framework is to track and update these dynamic facts as they evolve. Rather than examining every fact in the KG, we classify facts based on their associated relation type. Our assumption is that the relation can indicate whether a fact is likely to change. For instance, the relation *born_in* is typically static, while *occupation* or *plays_for* are dynamic, as they can change over time. This classification step is crucial because it significantly narrows the scope of facts that need validation, reducing both computational cost and reliance on web retrieval. Although a language model agent could theoretically decide when to fetch web data, our classification-based approach proves to be more efficient and produces more accurate results. Consider, for instance, the HumanWiki dataset (Rosso et al., 2021), which contains 221 relations in total. By classifying these relations, we can limit the search space to just 47.8%, focusing only on facts associated with the 145 dynamic relations out of the 221.

For this task, we use a backbone pre-trained Large Language Model (LLM), we feed the relation to classify and add examples as few-shots to the prompt with a clear definition of static and dynamic relations as depicted in Figure 2.

3.2.2 Web-Augmented LLMs for up-to-date Fact Retrieval

It has been widely acknowledged that Large Language Models (LLMs) are limited by their fixed training cutoff and can significantly benefit from access to external sources. In time-sensitive tasks, LLMs often struggle to provide up-to-date information and may hallucinate facts (Liu et al., 2024; Sriramanan et al., 2024; Mousavi et al., 2024). To address this, we integrate Google Search APIs[5] to retrieve relevant, real-time context. We use standard search engines, like Google, since they have strong ranking capabili-

[5]https://serper.dev/

ties, and they take freshness of the data into account[6]. We reformulate a natural language query, emphasizing the need for recent information, e.g., "Who is the current CEO of *company_name*?" or "Who is the current governor of *National_bank_name*?", this query is used to retrieve relevant passages which may include candidate answers or indicators. We concatenate the top-k results (with k being a tunable hyperparameter based on the model's maximum input context length). From each retrieved result, we extract the webpage title, publication date (if available), and a summarized version of the content. The title and date are crucial, as they help the model assess the credibility and recency of the source, which is especially important when dealing with conflicting or recently updated information.

To reduce hallucinations and enhance the reliability of the results, we prompt the model to provide a confidence score for each prediction, that could be High, Medium, or Low. Prior research has shown that LLMs are often capable of estimating their own certainty which also known as black-box Confidence Elicitation (Cash et al., 2024; Xiong et al., 2023). We leverage this self-assessed confidence score to determine which facts to update in the knowledge graph, prioritizing changes supported by high-confidence predictions.

3.2.3 KG Update: Entity Matching and Alignment

After receiving a response from the model, we extract both the predicted tail entity t_{pred} and the associated confidence score c. The prediction corresponds to the tail entity in a triple (h,r,t), where the head h and relation r are given by the original query. We then assess the confidence score c to determine how to proceed. If c is high, we check whether t_{pred} already exists in the current knowledge graph $\mathcal{G}$ using fuzzy matching to handle minor name variations.

If no matching entity is found in $\mathcal{G}$, we normalize t_{pred} using the Wikidata API[7] to retrieve its canonical identifier, and a new node is created accordingly. If a matching entity exists, the graph is updated by replacing the existing triple (h,r,t_{old}) with the new one (h,r,t_{pred}), where t_{old} is the original tail entity. No update is performed if t_{pred} is identical to t_{old}. In cases where the confidence score c is not high, the prediction is flagged for manual review to ensure data integrity. Algorithm 1.

[6]https://developers.google.com/search/docs/appearance/ranking-systems-guide

[7]https://www.wikidata.org/w/api.php

Input: t_{pred}, t_{old}, c, $\mathcal{G}$
Output: $\mathcal{G}_{updated}$
if *c is* ***high*** **then**
  Perform fuzzy matching with existing entities in $\mathcal{G}$;
  if $t_{pred} \notin \mathcal{G}$ **then**
    Normalize t_{pred} using Wikidata API;
    Create new node for t_{pred};
  end
  else if $t_{pred} \in \mathcal{G}$ ***and*** $t_{pred} \neq t_{old}$ **then**
    Update $\mathcal{G}$: replace (h,r,t_{old}) with (h,r,t_{pred});
  end
  else if $t_{pred} = t_{old}$ **then**
    return No update required;
  end
else
  Flag for manual review;
end

Algorithm 1: Entity Matching and Alignment in KG Update.

Table 1: Subset Statistics and Outdated Fact Rates.

Subset	Size	% Outdated Facts
CEOs (Companies)	339	46.4
Bank Governors	25	52.2
Nat'l Football Teams	50	100.0
Int'l Org. Leaders	325	61.6

4 EXPERIMENTAL EVALUATION

In this section we present our experimental setup, implementation details and findings.

4.1 Settings

Datasets. The HumanWiki dataset, introduced by (Rosso et al., 2021), is derived from Wikidata by extracting facts involving entities of type human (wd:Q5). This knowledge graph contains 221 distinct relations. Using our classification module, we categorize these relations into static and dynamic. For our experiments, we focus on a subset of dynamic relations, namely: chief executive officer (wdt:P169), chairperson (wdt:P488), and officeholder (wdt:P1308).

To construct focused sample datasets, we extract subgraphs composed of facts where these dynamic relations are used as predicates. This process is carried out using the Wikidata Query Service (WDQS)[8], which supports SPARQL-based querying over the

[8]https://query.wikidata.org/

Wikidata knowledge graph. We design SPARQL queries to retrieve the most recent and relevant facts corresponding to these roles.

We extract four distinct subgraphs: (1) Current CEOs of companies (2) Current governors of central banks (3) Current leaders of international organizations (4) Top-ranked national football teams.

Table 1 summarizes the statistics of the extracted subsets used in our experiments. The National Football Teams Ranking subset is included to illustrate the varying degrees of data dynamicity, with 100% of the records requiring updates. For the remainder of our experiments, we focus on the first three datasets, as they are less prone to rapid changes.

Models. For the classification module, we employ the pre-trained large language model (LLM) LLaMA3.1-70B (Touvron et al., 2023). For the new fact retrieval module, we evaluate three different LLMs: (1) LLaMA3.1-70B, (2) DeepSeek-R1-Distill-Llama-70B, a distilled variant of the R1 model based on LLaMA3.3-70B-Instruct, and (3) GPT-4o-mini, a cost-efficient, lightweight model. Our open source models are augmented with web search capabilities via the Serper API[9], retrieving up to 10 relevant passages along with their metadata (including title and publication date). The GPT-4o-mini model leverages a recently introduced web search preview tool with a low search context size [10]. The models are respectively denoted as follows: *llama3.1*, *r1.llama*, and *gpt-4o-mini*.

Evaluation. We evaluate model performance using the standard accuracy metric, which is the ratio of correct predictions to the total number of predictions. Each prediction is manually assessed by one human annotator, who verifies its correctness and temporal relevance by consulting reliable web sources.

4.2 Main Results

Table 2 presents the results of the retrieval step, comparing the performance of the three models across the three datasets, both with and without integrated web search. As expected, the models perform poorly without web access, often returning outdated or incorrect answers, or generic responses such as ”unknown” or ”not available” since the queries concern the current status of entities.

Introducing contextual web snippets significantly improves performance. For instance, the *r1.llama* model achieves an accuracy boost of up to 48.8% on the *CEOs* dataset. On the *Bank Governors* dataset, the *gpt-4o-mini* model reaches 100% accuracy, likely due to the dataset's small size and the recency of the reference sources used for verification.

On the *Int'l Org. Leaders* dataset, the models perform similarly, with *gpt-4o-mini* leading slightly by 0.9% over *r1.llama*, which itself outperforms *llama3.1* by 1.4%. Notably, *r1.llama* slightly outperforms *gpt-4o-mini* on the *CEOs* dataset.

The stronger results on the *CEOs* dataset, compared to the *Int'l Org. Leaders* dataset, may be attributed to the higher public visibility of CEOs, making them more likely to appear in pretraining data. In contrast, the leaders of international organizations are less well-known, presenting a greater challenge for the models. Without web search, performance on this dataset is particularly low, ranging from 18.9% to 27.4% with *r1.llama* performing best and *llama3.1* the worst, reflecting the difficulty the models face in retrieving accurate information about these lesser-known figures.

4.3 Model Comparison

As shown in Table 3, the models differ notably in their response times and costs. Specifically, *r1.llama* takes approximately 19 seconds on average to generate an answer, compared to about 2.5 seconds for *llama3.1* and 4 seconds for *gpt-4o-mini*. This difference can be attributed to the format of the returned output: *r1.llama* includes a detailed explanation of the reasoning behind its answer. Although we instructed the model to omit this explanation, it still provides it within special tokens `<think>` and `</think>`, which adds to the processing time. The Google Search Serper API[11], which we use alongside the *r1.llama* and *llama3.1* models, costs $1 per 1,000 queries. In contrast, leveraging the web search feature provided by OpenAI with the *gpt-4o-mini* model incurs a cost of around $30 per 1,000 queries, broken down into $25 for the search itself and approximately $5 for model input and output tokens.

In terms of the percentage of correctly updated facts relative to the total number of outdated facts, all three models perform well across the datasets. However, *gpt-4o-mini* shows an edge on the *Int'l Org. Leaders* dataset with 64.8%, while *r1.llama* performs best on the *CEOs* dataset with 90.4%. For the *Bank Governors* dataset, all models achieve the same up-

[9] https://serper.dev/

[10] https://platform.openai.com/docs/guides/tools-web-search

[11] https://serper.dev/

Table 2: Fresh Fact Retrieval Accuracy (%) of Models Across Datasets With and Without Web Search Integration.

Subset	Model	+web	-web
CEOs	llama3.1	86.7	45.7
	r1.llama	**92.6**	43.8
	gpt-4o-mini	91.1	**47.9**
Bank Governors	llama3.1	95.7	52.2
	r1.llama	95.5	52.2
	gpt-4o-mini	**100.0**	**60.9**
Int'l Org. Leaders	llama3.1	70.1	18.9
	r1.llama	72.5	**27.4**
	gpt-4o-mini	**73.4**	21.5

Table 3: Efficiency and Update Effectiveness of Models Across Datasets.

Model	Correct Updates (%)			Time (s)	Cost/$1K
	Bank Gov.	CEOs	Int'l Leaders		
llama3.1	100.0	81.5	59.3	2.5	$1
r1.llama	100.0	90.4	62.8	19.0	$1
gpt-4o-mini	100.0	84.1	64.8	4.0	$30

date accuracy of 100%, effectively updating all outdated facts in that sample.

Query: Who is the current CEO of LISI?
Initial LLM Prediction (w/o web):
`Florent Germain`, Confidence: *Medium*
Web Search Results:

- *Governance - LISI Group*: **Lionel Rivet** listed as CEO
- *LinkedIn*: **Emmanuel Viellard** as *Directeur Général*
- *Craft.co*: **Emmanuel Neildez** – CEO of LISI Aerospace
- *Bloomberg Markets (2023)*: Viellard confirmed as CEO since 2016
- *FII.FR*: Viellard listed as CEO of LISI SA
- *LISI Automotive*: **François Liotard** as division CEO

Model Reasoning:

Some names refer to division heads (e.g., Aerospace or Automotive). Emmanuel Viellard appears consistently across the most recent and credible sources as the CEO of the overall LISI Group.

Final Answer: **Emmanuel Viellard**, Confidence: *High*

Figure 3: Summarized reasoning by *r1.llama* in handling conflicting CEO data.

4.4 Confidence Elicitation Reliability

The confidence score serves as a key indicator of the model's certainty in its predictions. To assess the reliability of these scores, we analyze the proportion of false predictions made with high confidence versus those made with low confidence, as presented in Table 4.

Our analysis reveals that *llama3.1* exhibits the lowest rate of high-confidence false predictions (False-High) and the highest rate of low-confidence false predictions (False-Low). In contrast, *gpt-4o-mini* shows the highest False-High rate, reaching 90.7% on the *Int'l Org. Leaders* dataset, which suggests a tendency toward overconfidence. This variation can be explained by the number of references each model uses when generating its final predictions. Both *r1.llama* and *llama3.1* consider around 10 different passages from diverse sources, which may sometimes conflict and thus provide the model with cues of uncertainty. Conversely, *gpt-4o-mini*, limited by a smaller search context, relies on only one or a few references, often resulting in higher confidence scores.

As part of our analysis of confidence elicitation and interpretability, we present a case study illustrating the model's reasoning process (Figure 3). In this example, the *r1.llama* model correctly identifies the CEO when provided with contextual information

Table 4: Confidence score reliability: percentage of false predictions with low confidence and true predictions with high confidence

Model	Dataset	% False-High	% False-Low
r1.llama	CEOs	54.5	4.5
	Int'l Org. Leaders	51.7	19.1
llama3.1	CEOs	26.7	13.3
	Int'l Org. Leaders	27.1	38.5
gpt-4o-mini	CEOs	86.7	3.3
	Int'l Org. Leaders	90.7	0.0

from the web. Despite encountering conflicting candidate names across sources, the model uses accompanying text and metadata such as titles and publication dates to make an accurate prediction with high confidence.

4.5 Limitations and Discussion

In this work, we present a proof of concept for automating Knowledge Graph (KG) updates using web-augmented large language models (LLMs). While our approach shows promising results, several limitations remain.

First, the datasets used in our experiments are relatively small and do not reflect the scale of real-world KGs, which can contain billions of triples. However, our classification step helps narrow the scope by focusing on a subset of relations, making the task more manageable. In practice, practitioners can further prioritize validation by targeting older facts, high-impact entities, or frequently queried relations.

Second, certain cases present notable challenges for the models, particularly when entity representations evolve over time. For instance, "OL Group" was rebranded as "The Eagle Football Group", which can lead to confusion in entity matching. Similarly, ambiguous abbreviations complicate disambiguation. In the CEOs dataset, "Ada" originally refers to Ada Motors, yet the web results also included unrelated entities sharing the same abbreviation, like the American Diabetes Association.

Third, the confidence elicitation mechanism used in this study has shown limitations in reliability. More robust alternatives such as entropy-based uncertainty estimation, attention-based confidence scores, or ensemble methods could be employed, particularly in domains where precision and trustworthiness are critical.

Fourth, while *gpt-4o-mini* generally outperforms the other models in terms of accuracy, it presents challenges related to transparency and control. Its reliance on internal sources within the OpenAI ecosystem limits our ability to inspect or influence the references used during inference. In contrast, models like *r1.llama* and *llama3.1* are augmented using external web search APIs, offering greater control and traceability. Additionally, *gpt-4o-mini* is significantly more expensive compared to the other alternatives.

In summary, the key challenges faced by our approach involve scaling to real-world KG sizes and ensuring the reliability and interpretability of both the web context and the model's reasoning process.

5 CONCLUSIONS

Our proposed framework, AutoVu-KG, highlights the potential of web-augmented large language models (LLMs) for automating the validation and update of knowledge graphs (KGs). Due to their inherent training data cutoffs, LLMs alone cannot be fully trusted for up-to-date or accurate predictions. To address this, we integrate real-time external sources, such as the web, to enhance their reliability. Our experiments demonstrate that open-source solutions can match, and perform on par with closed-source counterparts, while offering significant advantages in cost and interpretability. This work underscores the powerful capabilities of LLM agents when combined with mechanisms for external control and enrichment, particularly in the domains of data management, quality assurance, and knowledge graph evolution. As a future direction, we aim to scale our approach to larger and domain-specific knowledge graphs.

ACKNOWLEDGEMENTS

This work was partially supported by the University of Passau through Project Kapital 1527 (Title 42951 UT 02).

REFERENCES

Agrawal, G., Kumarage, T., Alghamdi, Z., and Liu, H. (2023). Can knowledge graphs reduce hallucinations in llms?: A survey. *arXiv preprint arXiv:2311.07914*.

Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2024). Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*.

Babaiha, N. S., Elsayed, H., Zhang, B., Kaladharan, A., Sethumadhavan, P., Schultz, B., Klein, J., Freudensprung, B., Lage-Rupprecht, V., Kodamullil, A. T., Jacobs, M., Geissler, S., Madan, S., and Hofmann-Apitius, M. (2023). A natural language processing system for the efficient updating of highly curated pathophysiology mechanism knowledge graphs. *Artificial Intelligence in the Life Sciences*, 4:100078.

Cash, T. N., Oppenheimer, D. M., and Christie, S. (2024). Quantifying uncertainty: Testing the accuracy of llms' confidence judgments.

Feng, C., Zhang, X., and Fei, Z. (2023). Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs. *arXiv preprint arXiv:2309.03118*.

Hatem, S., Khoriba, G., Gad-Elrab, M. H., and ElHelw, M. (2024). Up to date: Automatic updating knowledge graphs using llms. *Procedia Computer Science*, 244:327–334. 6th International Conference on AI in Computational Linguistics.

Lai, H., Liu, X., Iong, I. L., Yao, S., Chen, Y., Shen, P., Yu, H., Zhang, H., Zhang, X., Dong, Y., and Tang, J. (2024). Autowebglm: A large language model-based web navigating agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 5295–5306, New York, NY, USA. Association for Computing Machinery.

Li, J., Tang, T., Zhao, W. X., Wang, J., Nie, J.-Y., and Wen, J.-R. (2023). The web can be your oyster for improving large language models.

Liu, F., Liu, Y., Shi, L., Huang, H., Wang, R., Yang, Z., Zhang, L., Li, Z., and Ma, Y. (2024). Exploring and evaluating hallucinations in llm-powered code generation.

Liu, X., Lai, H., Yu, H., Xu, Y., Zeng, A., Du, Z., Zhang, P., Dong, Y., and Tang, J. (2023). Webglm: Towards an efficient web-enhanced question answering system with human preferences.

Mousavi, S. M., Alghisi, S., and Riccardi, G. (2024). Dyknow: Dynamically verifying time-sensitive factual knowledge in llms.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. (2022). Webgpt: Browser-assisted question-answering with human feedback.

Peng, C., Xia, F., Naseriparsa, M., and Osborne, F. (2023). Knowledge graphs: Opportunities and challenges. *Artificial Intelligence Review*, 56(11):13071–13102.

Polleres, A., Pernisch, R., Bonifati, A., Dell'Aglio, D., Dobriy, D., Dumbrava, S., Etcheverry, L., Ferranti, N., Hose, K., Jiménez-Ruiz, E., et al. (2023). How does knowledge evolve in open knowledge graphs? *Transactions on Graph Data and Knowledge*, 1(1):11–1.

Rosso, P., Yang, D., Ostapuk, N., and Cudré-Mauroux, P. (2021). Reta: A schema-aware, end-to-end solution for instance completion in knowledge graphs. In *Proceedings of the Web Conference 2021*, pages 845–856.

Shenoy, K., Ilievski, F., Garijo, D., Schwabe, D., and Szekely, P. (2022). A study of the quality of wikidata. *Journal of Web Semantics*, 72:100679.

Sriramanan, G., Bharti, S., Sadasivan, V. S., Saha, S., Kattakinda, P., and Feizi, S. (2024). Llm-check: Investigating detection of hallucinations in large language models. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C., editors, *Advances in Neural Information Processing Systems*, volume 37, pages 34188–34216. Curran Associates, Inc.

Tang, J., Feng, Y., and Zhao, D. (2019). Learning to update knowledge graphs by reading news. In Inui, K., Jiang, J., Ng, V., and Wan, X., editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2632–2641, Hong Kong, China. Association for Computational Linguistics.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). Llama: Open and efficient foundation language models.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

Wei, Y., Huang, Q., Kwok, J. T., and Zhang, Y. (2024). Kicgpt: Large language model with knowledge in context for knowledge graph completion. *arXiv preprint arXiv:2402.02389*.

Xie, W., Liang, X., Liu, Y., Ni, K., Cheng, H., and Hu, Z. (2024). Weknow-rag: An adaptive approach for retrieval-augmented generation integrating web search and knowledge graphs.

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. (2023). Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Yang, B., tau Yih, W., He, X., Gao, J., and Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases.

Yao, L., Peng, J., Mao, C., and Luo, Y. (2025). Exploring large language models for knowledge graph completion. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Zhu, Y., Wang, X., Chen, J., Qiao, S., Ou, Y., Yao, Y., Deng, S., Chen, H., and Zhang, N. (2024). Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities. *World Wide Web*, 27(5):58.