KGValidator: A Framework for Automatic Validation of Knowledge Graph Construction

Jack Boylan[1], Shashank Mangla[1], Dominic Thorn[1], Demian Gholipour Ghalandari[1], Parsa Ghaffari[1] and Chris Hokamp[1]

[1]*Quantexa*

Abstract

This study explores the use of Large Language Models (LLMs) for automatic evaluation of knowledge graph (KG) completion models. Historically, validating information in KGs has been a challenging task, requiring large-scale human annotation at prohibitive cost. With the emergence of general-purpose generative AI and LLMs, it is now plausible that human-in-the-loop validation could be replaced by a generative agent. We introduce a framework for consistency and validation when using generative models to validate knowledge graphs. Our framework is based upon recent open-source developments for structural and semantic validation of LLM outputs, and upon flexible approaches to fact checking and verification, supported by the capacity to reference external knowledge sources of any kind. The design is easy to adapt and extend, and can be used to verify any kind of graph-structured data through a combination of model-intrinsic knowledge, user-supplied context, and agents capable of external knowledge retrieval.

Keywords

Text2KG, Knowledge Graph Evaluation, Knowledge Graph Completion, Large Language Models,

1. Introduction

Knowledge Graphs (KGs) are flexible data structures used to represent structured information about the world in diverse settings, including general knowledge [1], medical domain models [2], words and lexical semantics [3], and semantics [4]. Most KGs are incomplete [5], in the sense that there is relevant in-domain information that the graph does not contain. Motivated by this incompleteness, *knowledge graph completion* research studies methods for augmenting KGs by predicting missing links [6].

Challenges and Paradigms in KG Completion Evaluation: Evaluating KG completion models presents inherent challenges due to the natural incompleteness of most knowledge graphs (KGs) [5]. Traditional evaluation methods typically use a closed-world assumption (CWA), which deems absent facts to be incorrect, and may not effectively reflect the true capabilities of KG completion models [7, 8, 9]. Alternatively, the open-world assumption (OWA) offers a more realistic framework by recognizing that KGs are inherently incomplete [10]. However, OWA complicates evaluation due to the need for extensive manual annotation of

jackboylan@quantexa.com (J. Boylan); shashankmangla@quantexa.com (S. Mangla); dominicthorn@quantexa.com (D. Thorn); demiangholipour@quantexa.com (D. G. Ghalandari); parsaghaffari@quantexa.com (P. Ghaffari); chrishokamp@quantexa.com (C. Hokamp)

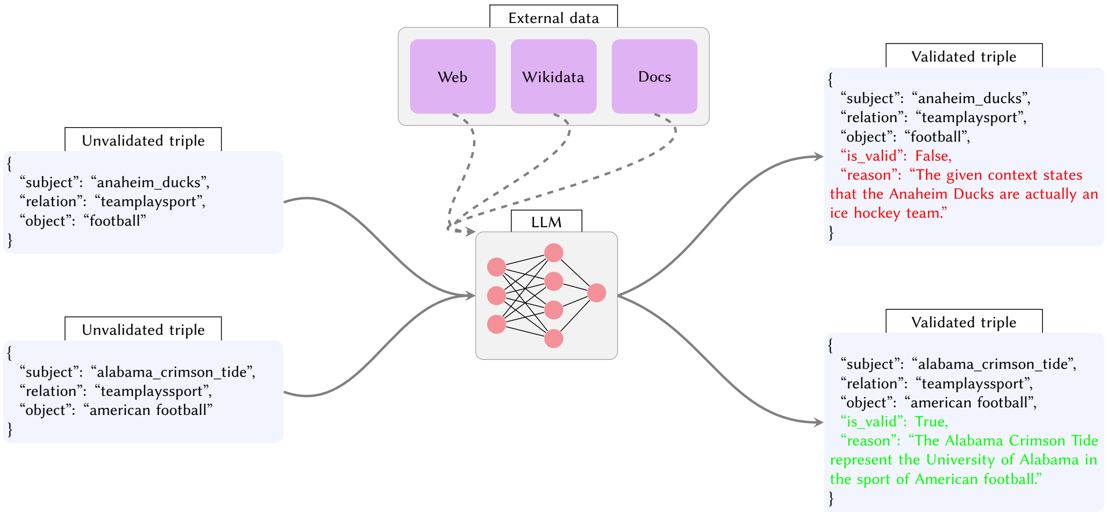

External data
Web
Wikidata
Docs
Unvalidated triple
{
"subject": "anaheim_ducks",
"relation": "teamplaysport",
"object": "football"
}
Unvalidated triple
{
"subject": "alabama_crimson_tide",
"relation": "teamplayssport",
"object": "american football"
}
LLM
Validated triple
{
"subject": "anaheim_ducks",
"relation": "teamplaysport",
"object": "football",
"is_valid": False,
"reason": "The given context states that the Anaheim Ducks are actually an ice hockey team."
}
Validated triple
{
"subject": "alabama_crimson_tide",
"relation": "teamplayssport",
"object": "american football",
"is_valid": True,
"reason": "The Alabama Crimson Tide represent the University of Alabama in the sport of American football."
}

Figure 1: Framework for Validating Knowledge Graph Triples.

unknown triples, leading to significant time and cost implications. Efforts to improve the efficiency of human-driven KG evaluation include strategies like cluster sampling, which aims to reduce costs by modeling annotation efforts more economically [11]. An illustration of these evaluation paradigms is shown in Figure 2.

KGValidator Framework: Motivated by these challenges, we introduce `KGValidator` as a flexible framework to evaluate KG Completion using LLMs. At its core, this framework validates the triples that make up a KG using context. This context can be the inherent knowledge of the LLM itself, a collection of text documents provided by the user, or an external knowledge source such as Wikidata or an Internet search (refer to Figure 1 for a high-level overview). Importantly, our framework does not require any gold references, which are often only available for popular benchmark datasets. This enables evaluation of a wider range of KGs using the same framework.

`KGValidator` makes use of the Instructor[1] library, Pydantic[2] classes, and function calling to control the generation of validation information. This ensures that the LLM follows the correct guidelines when evaluating properties, and outputs the correct data structures for calculating evaluation metrics. Our main contributions are:

- A simple and extensible framework based on open-source libraries that can be used to validate KGs with the use of LLMs[3].
- An evaluation of our framework against popular KG completion benchmark datasets to measure its effectiveness as a KG validator.

[1] https://github.com/jxnl/instructor
[2] https://docs.pydantic.dev/
[3] Unfortunately, IP restrictions currently prevent us from sharing our implementation, but we are happy to directly correspond with interested researchers who wish to reproduce our results

- An investigation of the impact of providing additional context to SoTA LLMs in order to augment evaluation capabilities.
- A straightforward protocol for implementing new validators using any KG alongside any set of knowledge sources.

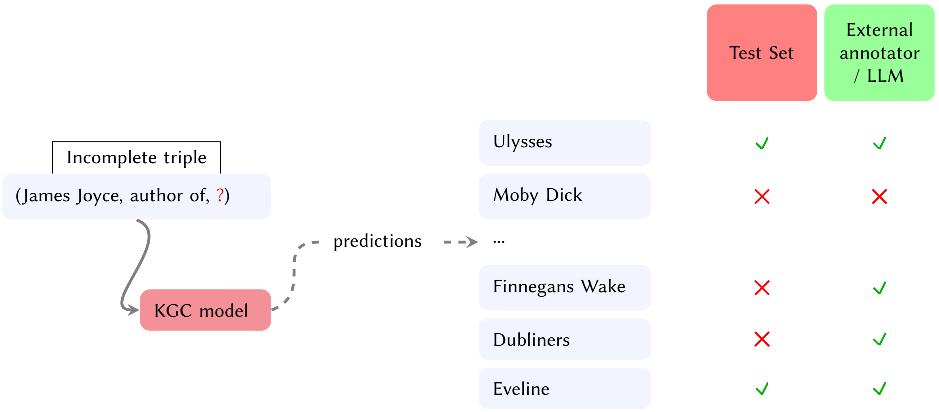


Figure 2: An example of the Closed-World Assumption in KG completion. Some of the triples predicted by a KG completion model are true in the real world (e.g. books written by James Joyce) but missing in the test set and would therefore be treated as false positives.

The rest of the paper is structured as follows: Section 2 discusses key related work, Section 3 covers our approach in detail, Section 4 presents several experiments designed to validate the framework, and Section 5 discusses results and possible extensions to this work.

2. Background

2.1. Knowledge Graph Construction

Knowledge Graphs can be represented as multi-relational directed *property graphs* [12], where nodes represent entities (for a general definition of *entity*), and edges are predicates or relations. Any KG can thus be rendered as a list of triples (*subject*, *relation*, *object*)[4], also called *statements*[5].

An early line of work on knowledge graph construction focused on the TAC 2010 **Knowledge Base Population (KBP)** shared task [13], which introduced a popular evaluation setting that separates knowledge base population into Entity Linking and Slot Filling subtasks. Early methods to address these tasks used pattern learning, distant supervision and hand-coded rules [14].

Knowledge Graph Completion (KGC) is a KG construction task that has gained popularity recently. It involves predicting missing links in incomplete knowledge graphs [9]. The subtasks include **triple classification**, where models assess the validity of (head, relation, tail) triples; **link prediction**, which proposes subjects or objects for incomplete triples; and **relation**

[4] several standards and formats exist for representing triples and optionally including additional metadata, including RDF, Turtle, N-triples, JSON-LD, and others.
[5] https://www.wikidata.org/wiki/Help:Statements

prediction [15], identifying relationships between subject and object pairs. Models for these tasks are frequently benchmarked against subsets of well-established knowledge bases such as WordNet [16], Freebase [17], and domain-specific KGs like UMLS [18].

Evaluation methodologies for KG completion primarily utilize ranking-based metrics. These include Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hits@K, which gauge a model's ability to prioritize correct triples over incorrect ones, offering a quantifiable measure of performance [15].

Outside these tightly defined tracks, various approaches have been proposed to construct or populate knowledge graphs. For example, NELL (Never-Ending Language Learner) [19] is a self-supervised system that was designed to interact with the internet over years to populate a growing knowledge base of topical categories and factual statements.

2.2. LLMs and Knowledge Graphs

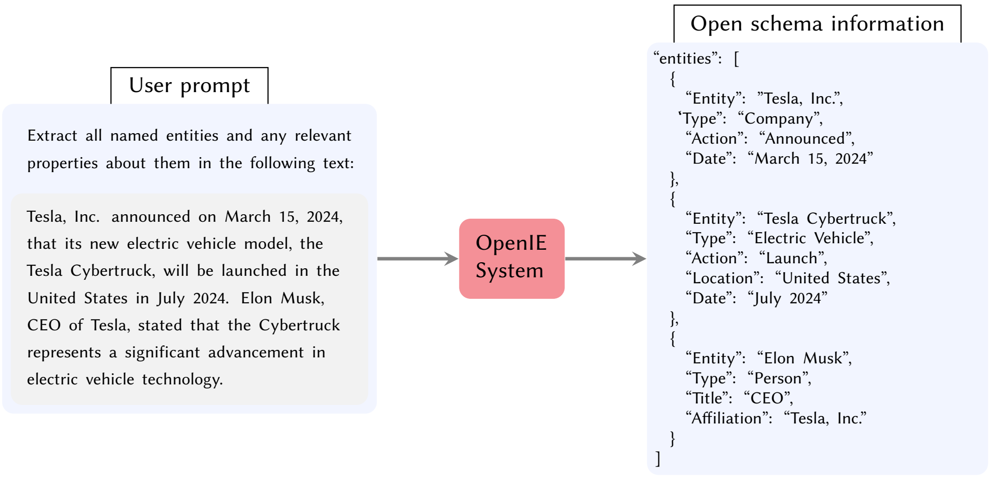
Studies have shown that pretrained language models (PLMs) possess factual and relational knowledge which makes them effective at downstream knowledge-intensive tasks such as open question-answering, fact verification, and information extraction [20, 21]. KG-BERT [22] uses PLMs for KG completion by fine-tuning BERT on all KG completion sub tasks, treating the problem as a sequence classification task.

Pretrain-KG [23] introduces a framework that enriches knowledge graph embedding (KGE) models with PLM knowledge during training, which proves to be particularly useful for low-resource scenarios of link prediction and triple classification.

Knowledge Graph Construction Using Generative AI With the proliferation of general-purpose LLMs [24], open information extraction (OpenIE) has become one of the most popular industry applications of generative AI [25]. OpenIE is closely related to knowledge graph construction, and so LLMs have naturally been applied to KG completion tasks such as link prediction and triple classification, proving to be successful in both fine-tuned [26] and zero-shot settings[27, 28]. The dominant paradigm is to include the desired schema of the output in the user prompt along with the input itself (refer to Figure 3).

Khorashadizadeh et al. demonstrate the capabilities of GPT 3.5 in the task of KG construction using an in-context learning approach [29]. Emphasis is placed on the importance of good prompt design under this setting. LLM2KB [30] fine-tunes open-source LLMs to predict tail entities given a head entity and relation, incorporating context retrieval from Wikipedia to enhance the relevance and accuracy of the predicted entities. Zhu et al. [31] investigate GPT-4's [32] capabilities for different steps of knowledge graph construction. They show that while GPT-4 exhibits modest performance on few-shot information extraction tasks, it excels as an inference assistant due to it's strong reasoning capabilities. Their experiments also show that GPT-4 generalizes well to new knowledge by creating a virtual knowledge extraction task.

Complementing these advancements, resources such as the Text2KG Benchmark [33] offer valuable tools for researchers to develop and test LLM-backed KG completion models. This benchmark, specifically designed for evaluating knowledge graph generation from unstructured text using guideline ontologies, marks a significant step towards standardizing and accelerating research in this field.



Extract all named entities and any relevant properties about them in the following text:

Tesla, Inc. announced on March 15, 2024, that its new electric vehicle model, the Tesla Cybertruck, will be launched in the United States in July 2024. Elon Musk, CEO of Tesla, stated that the Cybertruck represents a significant advancement in electric vehicle technology.

```
“entities”: [
  {
    “Entity”: ”Tesla, Inc.”,
  ‘Type”: “Company”,
    “Action”: “Announced”,
    “Date”: “March 15, 2024”
  },
  {
    “Entity”: “Tesla Cybertruck”,
    “Type”: “Electric Vehicle”,
    “Action”: “Launch”,
    “Location”: “United States”,
    “Date”: “July 2024”
  },
  {
    “Entity”: “Elon Musk”,
    “Type”: “Person”,
    “Title”: “CEO”,
    “Affiliation”: “Tesla, Inc.”
  }
]
```

Figure 3: An example of Open Information Extraction. Note that in OpenIE, the output schema is not fixed.

A comprehensive survey on the unification of LLMs and KGs [34] highlights the emergence of *KG-enhanced LLMs*, *LLM-augmented KGs*, and *Synergized LLMs and KGs*. Validation and evaluation of KGs with LLMs has been less explored, but is also a promising and important avenue for research.

2.3. Structuring and Validating Language Model Output

Constraining language models to produce outputs that conform to specific schemas is challenging but essential for applications like natural language to SQL (NL2SQL) [35, 36]. Recent developments include tools like Guidance[6], Outlines[7], JSONFormer[8], and Guardrails[9], which facilitate constrained decoding of structured outputs from large language models (LLMs). Additionally, *semantic validation* techniques like those enabled by the Instructor library use Pydantic classes to ensure outputs meet both structural and semantic accuracy. This advancement is crucial for tasks such as knowledge graph (KG) completion, where precision in data parsing significantly enhances model utility [28].

2.4. Knowledge-Grounded LLMs

The tendency of LLMs to hallucinate poses a significant challenge in their application to downstream tasks [37]. Retrieval-Augmented Generation (RAG) mitigates this by grounding LLM responses in verified information, significantly enhancing accuracy and reliability [38, 39]. RAG integrates a retrieval component that leverages external knowledge during the generation process, improving performance across various natural language processing tasks [40]. Additionally, role-playing approaches using LLMs have been developed to create detailed,

[6] https://github.com/guidance-ai/guidance
[7] https://github.com/outlines-dev/outlines
[8] https://github.com/1rgs/jsonformer
[9] https://github.com/guardrails-ai/guardrails

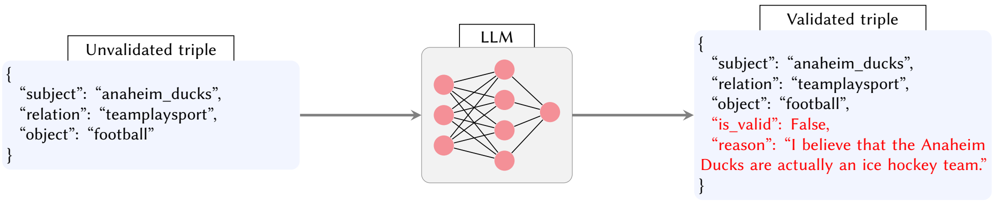
organized content similar to Wikipedia articles, drawing on trusted sources for factual grounding [41].

2.5. Knowledge Graph Evaluation

Evaluating automatically constructed knowledge graphs is challenging. Huaman et al. present a comprehensive evaluation of state-of-the-art validation frameworks, tools, and methods for KGs [42]. They highlight the challenges in validating KG assertions against real-world facts and the need for scalable, efficient, and effective semi-automatic validation approaches. Gao et al. [11] have highlighted the trade-offs between human annotation cost and meaningful estimates of accuracy. As discussed above, a common flaw reported in existing KG evaluation frameworks is use of a closed-world assumption. Specifically, this means treating unknown predicted triples as false [43]. Sun et al. [9] find that several recent KG completion techniques have reported significantly higher performance compared to earlier SoTA methods, in some cases due to the inappropriate evaluation protocols used. Cao et al. [44] suggest that triple classification evaluation under the closed-world assumption leads to trivial results. Additionally, Cao et al. note that current models lack the capacity to distinguish *false* triples from *unknown* triples. Yang et al. [10] confirm the existing gap between closed and open world settings in the performance of KG completion models.

3. Approach

We assume the existence of a triple-extractor model, which produces a stream of candidate *statements* from unstructured data feeds. The triple-extractor model could be implemented by a KG completion model, one or more LLMs with well-designed prompts, or by a more traditional information extraction pipeline consisting of several distinct models that perform parsing, named entity recognition, relationship classification, and other relevant sub-tasks. For each predicted triple from the stream, we wish to validate whether it is correct in the presence of context. Once a statement has been validated, it can be written into a knowledge graph or another data store, and statements that do not pass validation can be flagged for further review. A high-level overview of the validation stage is illustrated in Figure 4.



```
{
  “subject”: “anaheim_ducks”,
  “relation”: “teamplaysport”,
  “object”: “football”
}
```

```
{
  “subject”: “anaheim_ducks”,
  “relation”: “teamplaysport”,
  “object”: “football”,
  “is_valid”: False,
  “reason”: “I believe that the Anaheim
Ducks are actually an ice hockey team.”
}
```

Figure 4: Validating KGs with LLM Knowledge

In this work we use existing standard KGC datasets for our experiments, so in practice the candidate triples in this work are produced by streaming through existing datasets (see Section 4).

Possible sources of context for validation include:

- Knowledge accrued in the LLM parameters during pretraining.
- User-provided context in the form of document collections or reference KGs represented in string format.
- Agents that can interact with the world to search and retrieve information in various ways.

Further detail on use of context in our validator implementations is provided in Section 3.1.

Basic Settings for Validation: The first step is to obtain KG completion predictions in the format of a list of (h, r, t) triples, each consisting of a head entity h, a relation r and a tail entity t. All validators are instantiated in a zero-shot setting with an LLM backbone; this may be a model from OpenAI's model family, such as gpt-3.5-turbo-0125 [45, 32], or an open-source model from the Llama family [46]. Additionally, validators have access to various tools which allow them to query external knowledge sources.

Validation via Pydantic Models Pydantic is a data validation and settings management library which leverages Python type annotations. It allows for the creation of data models, where each model defines various fields with types and validation requirements. By using Python's type hints, Pydantic ensures that the incoming data conforms to the defined model structure, performing automatic validation at runtime.

KG triples are passed to the validator via the Instructor library, which uses a patched version of popular LLM API clients. This patch enables the request of structured outputs in the form of Pydantic classes. It is within these Pydantic classes that we specify the structural and semantic guidelines that the LLM must follow during validation. An example of this form of prompting is shown in Figure 7. Specifically, we request that, for every triple (h, r, t), the model must provide values for a number of fields:

1. `triple is valid`: A boolean indicating whether the proposed triple is generally valid, judged against any given context. The model can reply with `True`, `False`, or `"Not enough information to say"`.
2. `reason`: An open-form string describing why the triple is or is not valid.

3.1. Validation Contexts

This section discusses the contextual information that is available to different validator instantiations. We use *context* to mean all information that is available to a validator, including the information stored in trained model parameters.

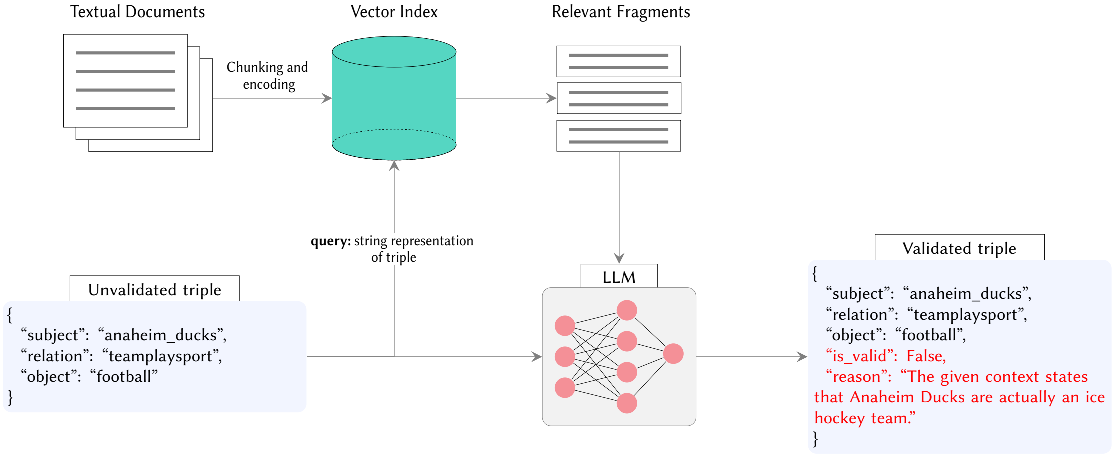
3.1.1. Validating with LLM Knowledge

This is the most straightforward method of triple validation. Given a triple (h, r, t), the objective is to classify the triple using the LLM's inherent knowledge about the world, learned during the pretraining stage, and stored in the model parameters. The process is illustrated in Figure 4 and an example can be found in appendix Figure 10. This is a powerful and simple way to verify triples with no additional data.

3.1.2. Validation using Textual Context(s)

Inspired by the success of Retrieval-Augmented Generation (RAG) in knowledge-intensive tasks such as question-answering [38], we implement tooling to retrieve relevant information from a reference text corpus (see Section 2.4). In this instance, the model is prompted with textual context alongside the candidate triple, as shown in Figure 5. This approach is particularly useful for a number of scenarios:

- When we wish to verify a set of triples about the same entity or group of entities and we have a collection of trustworthy sources within which we assume there will be evidence for or against the predicted triple, for example a given entity's Wikipedia page.
- When building KGs using private or domain-specific data feeds.



```
{
  "subject": "anaheim_ducks",
  "relation": "teamplaysport",
  "object": "football"
}
```

```
{
  "subject": "anaheim_ducks",
  "relation": "teamplaysport",
  "object": "football",
  "is_valid": False,
  "reason": "The given context states
that Anaheim Ducks are actually an ice
hockey team."
}
```

Figure 5: Validating KGs given Textual Context

This provided corpus can be of arbitrary length and can contain a collection of documents. The corpus will be recursively chunked and encoded by an embedding model from either the sentence transformers library [47] or OpenAI's family of embedding models [48], and a searchable index is created. A string representation for each triple is then constructed, and this is used to query the corpus index, which retrieves the most semantically similar chunks of text, according to cosine similarity. This forms the context against which the LLM will validate the given triple.

3.1.3. Validation using a Reference KG

We also consider validating proposed KG triples by cross-referencing against established, reliable KGs. Wikidata, with its expansive and well-structured repository of knowledge, serves as an ideal reference point for such validations, and will serve as the reference KG in our experiments. However, we note that any KG can be used as a reference by following the method outlined in 3.1.3.

The Wikidata knowledge graph is built from two top-level types: *Entities* and *Properties*:

Entities: Entities represent all items in the database. An item is a real-world object, concept, or event, such as “Earth” (Q2), “love” (Q316), or “World War II” (Q362). Items can be linked to each other to form complex statements via properties. In the context of KG completion, a statement can be thought of as a triple. Each entity is identified by a unique identifier, which is a Q-prefix followed by a sequence of numbers, e.g., Q42 for Douglas Adams.

Properties: Properties in Wikidata define the characteristics or attributes of items and establish relationships between them. They are the predicates in statements, linking subjects (items) to their object (value or another item). For example, in the statement “Douglas Adams (Q42) - profession (P106) - writer (Q36180)”,“profession” is the property that describes the relationship between “Douglas Adams” and “writer”.

Reference KG Implementation Our approach to integrating Wikidata as a source of contextual information is simple. Given triple *t*, an agent module searches Wikidata using the string of the subject as a query. The top Wikidata entity from the search API is returned – if no results are found for the query, a warning is thrown, and the validator will default to using its inherent knowledge. The Wikidata item is parsed to remove a list of trivial properties. Among Wikidata’s 11,000 Properties, over 7,000 of these are identifiers to external databases such as IMDb and Reddit [10]. In this work, we are not interested in verifying such information, and so we discard these properties.

A string representation of the Wikidata page is now passed through the same RAG pipeline as described in Section 3.1.2, from which relevant sections are retrieved and passed to the validator as context alongside each predicted triple *t*. This implementation is illustrated in appendix Figure 9.

3.2. Validation using Web Search

In some cases, the triples we wish to validate cannot be captured with a query to Wikidata, and we do not have a collection of textual information to provide the model with additional context. To overcome this, the validator is given access to collect information relevant to the triple via a web-searching agent. The triple is formatted as a string query. An agent then searches the web using the DuckDuckGo API[11]. The top results for the given query are parsed and stored as a collection of documents. The validation then follows the same pattern as Section 3.1.2, whereby relevant chunks of text are retrieved as context for triple validation. This method is illustrated in appendix Figure 8.

4. Experiments

We conduct a series of triple classification experiments to validate the effectiveness of an LLM-backed validator for KG Completion. Our experiments make use of a number of popular

[10] https://wikiedu.org/blog/2022/03/30/property-exploration-how-do-i-learn-more-about-properties-on-wikidata/
[11] https://github.com/deedy5/duckduckgo_search

benchmark KG datasets: UMLS [18], WN18RR [49], FB15K-237N, Wiki27k [8], and CoDeX-S [50]. FB15k-237N is derived from Freebase, and was obtained by removing the relations containing mediator nodes in FB15K-237. Wiki27K was created from Wikidata and manually annotated with real negative triples. UMLS is a medical ontology describing relations between medical concepts. WN18RR is a dataset about English morphology derived from WordNet. We investigate the performance of `gpt-3.5-turbo-0125` and `gpt-4-0125-preview` and present our results in Tables 1 , 2 and 3. Setup details and results for open-source LLM experiments can be found in Section A.3 and Table 4 in the appendix.

Table 1
Experiment results for FB15K-237N-150 and Wiki27K-150 datasets. Accuracy (Acc), precision (P), recall (R), and F1-score (F1) results for each method are reported. The best metrics for each dataset are marked in bold.

Model	FB15K-237N-150				Wiki27K-150			
	P	R	F1	Acc	P	R	F1	Acc
GPT 3.5 WorldKnowledge	0.58	**0.97**	0.73	0.63	0.63	**1.0**	0.77	0.71
GPT 3.5 Wikidata	0.75	0.77	0.76	0.76	0.74	0.73	0.74	0.74
GPT 3.5 WikipediaWikidata	0.85	0.69	0.76	0.79	0.84	0.86	0.85	0.85
GPT 3.5 Web	0.76	0.85	0.81	0.79	0.76	0.91	0.82	0.81
GPT 3.5 WikidataWeb	0.82	0.81	**0.82**	0.82	0.78	0.87	0.82	0.81
GPT 4 WorldKnowledge	0.87	0.72	0.79	0.81	0.95	0.76	0.84	0.86
GPT 4 Wikidata	0.89	0.64	0.74	0.78	0.97	0.75	0.84	0.86
GPT 4 WikipediaWikidata	0.90	0.59	0.71	0.76	0.97	0.77	0.86	0.87
GPT 4 Web	**0.92**	0.72	0.81	**0.83**	0.95	0.75	0.84	0.85
GPT 4 WikidataWeb	**0.92**	0.72	0.81	**0.83**	**1.0**	0.77	**0.87**	**0.89**

Table 2
Experiment results for WN18RR-150 and UMLS-150 datasets. Accuracy (Acc), precision (P), recall (R), and F1-score (F1) results for each method are reported. The best metrics for each dataset are marked in bold.

Model	WN18RR-150				UMLS-150			
	P	R	F1	Acc	P	R	F1	Acc
GPT 3.5 WorldKnowledge	0.54	0.97	0.70	0.58	0.5	**0.97**	0.66	0.5
GPT 3.5 Wikidata	0.53	**0.99**	0.69	0.56	0.51	0.87	0.64	0.52
GPT 3.5 WikipediaWikidata	0.54	**0.99**	0.69	0.57	0.53	0.88	0.66	0.55
GPT 3.5 Web	0.67	0.97	0.79	0.74	0.52	0.93	**0.67**	0.53
GPT 3.5 WikidataWeb	0.69	0.95	0.80	0.76	0.5	0.88	0.64	0.5
GPT 4 WorldKnowledge	0.99	0.92	**0.95**	**0.95**	0.57	0.77	0.66	0.59
GPT 4 Wikidata	0.99	0.91	0.94	**0.95**	**0.63**	0.69	0.66	**0.64**
GPT 4 WikipediaWikidata	0.99	0.91	0.94	**0.95**	0.62	0.67	0.64	0.63
GPT 4 Web	**1.0**	0.89	0.94	**0.95**	0.61	0.65	0.63	0.62
GPT 4 WikidataWeb	**1.0**	0.88	0.94	0.94	0.56	0.64	0.60	0.57

Table 3
Experiment results for CoDeX-150 dataset. Accuracy (Acc), precision (P), recall (R), and F1-score (F1) results for each method are reported. The best metrics are marked in bold.

Model	CoDeX-S-150			
	P	R	F1	Acc
GPT 3.5 WorldKnowledge	0.52	**0.97**	0.68	0.54
GPT 3.5 Wikidata	0.86	0.88	0.87	0.87
GPT 3.5 WikipediaWikidata	0.81	0.87	0.84	0.83
GPT 3.5 Web	0.74	0.84	0.79	0.77
GPT 3.5 WikidataWeb	0.87	**0.97**	**0.92**	**0.91**
GPT 4 WorldKnowledge	0.87	0.81	0.84	0.85
GPT 4 Wikidata	0.93	0.87	0.90	0.9
GPT 4 WikipediaWikidata	**0.94**	0.83	0.88	0.89
GPT 4 Web	0.85	0.84	0.85	0.85
GPT 4 WikidataWeb	0.93	0.85	0.89	0.89

4.1. Experiment Settings

Prompt as a Hyperparameter: We emphasize the notion of a prompt as a model hyperparameter, and manually tuning it to fit a subset of data is a form of over-fitting or evaluation set leakage. In this work, we thus formulate a generic model prompt, and apply this prompt to all benchmark datasets without further changes. We include the prompt in the appendix (see Figure 6).

Through the following experiments we attempt to answer the question: Given context, can our model judge whether an unseen triple (h, r, t) is correct?

We are primarily interested in observing the change in evaluation performance of an LLM when it has access to context under the following settings:

- **LLM Inherent Knowledge**: Evaluates the model's native understanding without external data sources.
- **Wikidata**: Uses structured data from Wikidata as the reference KG context.
- **Web**: Incorporates information retrieved directly from the internet.
- **WikidataWeb**: Combines data from both Wikidata and web sources.
- **WikipediaWikidata**: Utilizes a mix of Wikipedia and Wikidata to provide a comprehensive context.

API Cost and Rate-Limiting Constraints Due to OpenAI API constraints, we run experiments using a subset of 150 examples from each dataset. This is indicated by the `-150` suffix to each dataset name.

5. Discussion

5.1. Analysis

Our analysis reveals notable variations in performance across datasets, as evidenced by the results obtained using different validators powered by GPT-3.5 and GPT-4 language models. Specifically, the GPT-3.5 `World Knowledge` validator shows limited effectiveness on the FB15K-237N-150, Wiki27k-150, and CoDeX-S-150 datasets (as detailed in Tables 1 and 3). However, the introduction of contextual information from Wikidata and web searches gives a strong performance boost, with the performance on the CoDeX-S-150 dataset in particular improving accuracy from 0.54 to 0.91 when using the `WikidataWeb` validator.

GPT-4 configurations exhibit strong performance across the board, particularly excelling in the FB15K-237N-150 and Wiki27k-150 datasets, where GPT-4 achieves the highest accuracy of 0.83 and 0.89 respectively. However, both GPT-3.5 and GPT-4 models demonstrate less satisfactory results on the UMLS-150 dataset, as indicated in Table 2.

It is noteworthy that the incorporation of context from external knowledge sources, especially web searches and Wikidata, proves beneficial for both models. Despite this, the open-source Llama2 model performs poorly on this task, as shown in Table 4 and inference examples 11 and 12. We hypothesize that future open-source LLMs may perform much better than the those currently available.

GPT-4 validators display effectiveness on the WN18RR-150 dataset, both with and without supplemental context. This robust performance is hypothesized to stem from the model's superior grasp of English morphology and nuanced language comprehension, aligning with the linguistic focus of the WN18RR dataset.

5.2. Key Findings

Inherent Knowledge Insufficiency: In the case of GPT-3.5 and Llama2 70B, reliance solely on the inherent knowledge of LLM validators often leads to unquestioned acceptance of predicted triples. This indicates a limitation in the models' ability to challenge the veracity of the information, underscoring the need for external validation mechanisms. This corroborates with findings from prior studies which find that LLMs struggle to memorize knowledge in the long tail [51].

Challenge in Verifying Ambiguous Triples: Our evaluation of each dataset reveals that additional information is neccesary to verify many triples. For example, a positive triple in the UMLS dataset reads `("age_group", "performs", "social_behavior")`. Ambiguous triples in the UMLS and WN18RR datasets require understanding of specific ontologies, rendering web or Wikidata searches ineffective for retrieving relevant context. This complexity is contrasted by datasets like FB15K-237N and Wiki27k, which involve concrete entities or facts (e.g., people, locations) more amenable to validation through widely available external sources. For example, a positive example in FB15K-237N reads `"Tim Robbins", "The gender of [X] is [Y] .", "male",`

The Importance of Relevant Context: Performance is weaker on datasets requiring domain-specific knowledge, such as UMLS, where no model tested achieved satisfying results. This is attributed to the challenge of sourcing pertinent context for validation, as exemplified by the clinical domain triple from UMLS: `("research_device", "causes", "anatomical_abnormality")`. This highlights the critical role of context in enabling accurate validation, emphasizing the need for targeted search strategies to augment the model's knowledge base.

Limitations in Zero-Shot Triple Classification by Current Open-Source LLMs: Table 4 shows the performance of a version of the LLama-2-70B-chat model [46] on the triple verification task. Upon manual inspection, the model nearly always returns a `True` prediction for all triples, irrespective of the provided context, resulting in a recall of 1.0 and precision of about 0.5 across all settings. This tendency suggests that while the model can superficially engage with the context—evident from relevant factoids appearing in the `reason` field—it often resorts to fabricating agreeable responses rather than accurately assessing the triple's validity. Figures 11 and 12 illustrate this behaviour. It is worth noting that our experiments were conducted using a single open-source model; however, alternative models could potentially deliver superior performance. We propose this as an avenue for future research.

Adoption of Other Open-Source LLMs: At present, we find that only OpenAI and Llama models are usable with the Instructor framework. More recent models, such as Mixtral [52] and Gemma [53], are beginning to receive support under this library, but issues with constraining model output has delayed implementation. We are particularly interested in observing how other open-source models perform at this task in the future.

5.3. Ethical and Social Risks

Building on the framework by Weidinger et al. [54], we highlight key ethical and social risks associated with using LLMs for KG validation. LLMs, trained on large-scale internet datasets, may perpetuate biases [55], discriminating against marginalized groups and potentially reinforcing stereotypes within KGs. Additionally, the alignment of LLM outputs with human preferences can introduce biases favoring certain languages and perspectives [56]. Privacy concerns also arise from LLMs potentially leaking sensitive information [57]. Furthermore, the risk of spreading misinformation through inaccurate validation poses serious challenges, especially in sensitive domains like medicine or law. Lastly, the environmental impact of training and deploying LLMs, including significant carbon emissions and water usage, underscores the need for sustainable practices in LLM-driven KG validation [58, 59].

6. Conclusions

We have introduced a flexible framework for utilizing large language models for the validation of triples within knowledge graphs, capitalizing on both the inherent knowledge embedded within these models, and upon supplementary context drawn from external sources. As demonstrated in our experiments (Section 4), the approach significantly enhances the accuracy of zero-shot triple classification across several benchmark KG completion datasets, provided that the appropriate context can be retrieved from external sources.

Use Cases: From experimentation, LLMs have demonstrated the potential to be effective validators for KG completion methods. They also open up the possibility of updating existing KG datasets with new knowledge from external sources, ensuring their relevance as gold-standard benchmarks. A practical application of this is the development of automated systems, such as bots, designed to enrich platforms like Wikidata with real-world data. These bot contributions could be systematically verified by SoTA LLMs to ensure accuracy and relevance.

As of January 2024, Wikidata encompassed nearly 110 million pages, a figure increasing at an accelerating rate. The decade between 2014 and 2023 saw an annual average of 9.57 million new pages and 191.5 million edits, and cumulative annual growth rates of 12.83% and 12.16% respectively [60]. The volume and pace of such expansion highlights the challenge of relying on manual verification methods. Leveraging LLMs to flag incorrect or unsupported edits made by users or bots could be an excellent aid to the Semantic Web community.

Future Research: As the quality of general-purpose LLMs improves, this framework should become increasingly effective in validating KG completion models. Instructor has already begun work to support other open-source LLMs, which would enable even greater flexibility in validator configuration.

Enriching models with domain-specific context and graph structural features could boost their performance across diverse datasets. Moreover, fine-tuning strategies tailored to LLMs may unlock even better performance when a model is fine-tuning specifically for the KG validation task.

As discussed in Sections 1 and 2, a growing body of work studies knowledge graph creation and augmentation using generative models. Knowledge graph creation is outside the scope of this paper, but we plan to explore this in future work. Given an information extraction model which produces KG triples from raw text, our verification pipeline could be connected to the entity and property stores of an existing KG, and automatically update the KG with high-accuracy information extracted from textual data feeds such as news. We note this is likely to be easier for some domains then others, and current SoTA LLMs will probably not be good verifiers for domain specific KGs.

References

[1] D. Vrandečić, M. Krötzsch, Wikidata: a free collaborative knowledgebase, Commun. ACM 57 (2014) 78–85. URL: https://doi.org/10.1145/2629489. doi:10.1145/2629489.

[2] C. J. Koné, M. Babri, J. M. Rodrigues, Snomed ct: A clinical terminology but also a formal ontology, Journal of Biosciences and Medicines (2023). URL: https://api.semanticscholar.org/CorpusID:265433665.

[3] G. A. Miller, Wordnet: a lexical database for english, Commun. ACM 38 (1995) 39–41. URL: https://doi.org/10.1145/219717.219748. doi:10.1145/219717.219748.

[4] C. F. Baker, C. J. Fillmore, J. B. Lowe, The berkeley framenet project, in: COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics, 1998.

[5] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, W. Zhang, Knowledge vault: a web-scale approach to probabilistic knowledge fusion, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, Association for Computing Machinery, New York, NY, USA, 2014, p. 601–610. URL: https://doi.org/10.1145/2623330.2623623. doi:10.1145/2623330.2623623.

[6] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, Z. Duan, Knowledge graph completion: A review, Ieee Access 8 (2020) 192435–192456. URL: "https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9220143".

[7] R. Reiter, On Closed World Data Bases, Technical Report, CAN, 1977.

[8] X. Lv, Y. Lin, Y. Cao, L. Hou, J. Li, Z. Liu, P. Li, J. Zhou, Do pre-trained models benefit knowledge graph completion? a reliable evaluation and a reasonable approach, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3570–3581. URL: https://aclanthology.org/2022.findings-acl.282. doi:10.18653/v1/2022.findings-acl.282.

[9] Z. Sun, S. Vashishth, S. Sanyal, P. Talukdar, Y. Yang, A re-evaluation of knowledge graph completion methods, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 5516–5522. URL: https://aclanthology.org/2020.acl-main.489. doi:10.18653/v1/2020.acl-main.489.

[10] H. Yang, Z. Lin, M. Zhang, Rethinking knowledge graph evaluation under the open-world assumption, 2022. arXiv:2209.08858.

[11] J. Gao, X. Li, Y. E. Xu, B. Sisman, X. L. Dong, J. Yang, Efficient knowledge graph accuracy evaluation, 2019. arXiv:1907.09657.

[12] R. Angles, The property graph database model, in: Alberto Mendelzon Workshop on Foundations of Data Management, 2018. URL: https://api.semanticscholar.org/CorpusID:43977243.

[13] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, J. Ellis, Overview of the tac 2010 knowledge base population track, in: Third text analysis conference (TAC 2010), volume 3, 2010, pp. 3–3. URL: https://blender.cs.illinois.edu/paper/kbp2011.pdf.

[14] H. Ji, R. Grishman, Knowledge base population: Successful approaches and challenges, in: D. Lin, Y. Matsumoto, R. Mihalcea (Eds.), Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 1148–1158. URL: https://aclanthology.org/P11-1115.

[15] T. Shen, F. Zhang, J. Cheng, A comprehensive overview of knowledge graph completion, Knowledge-Based Systems (2022) 109597.

[16] C. Fellbaum, Wordnet, in: Theory and applications of ontology: computer applications, Springer, 2010, pp. 231–243.

[17] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM

SIGMOD international conference on Management of data, 2008, pp. 1247–1250.

[18] O. Bodenreider, The unified medical language system (umls): integrating biomedical terminology, Nucleic acids research 32 (2004) D267–D270.

[19] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, J. Welling, Never-ending learning, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15), 2015.

[20] T. Shin, Y. Razeghi, R. L. L. I. au2, E. Wallace, S. Singh, Autoprompt: Eliciting knowledge from language models with automatically generated prompts, 2020. arXiv:2010.15980.

[21] F. Petroni, T. Rocktäschel, P. Lewis, A. Bakhtin, Y. Wu, A. H. Miller, S. Riedel, Language models as knowledge bases?, 2019. arXiv:1909.01066.

[22] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, 2019. arXiv:1909.03193.

[23] Z. Zhang, X. Liu, Y. Zhang, Q. Su, X. Sun, B. He, Pretrain-KGE: Learning knowledge representation from pretrained language models, in: T. Cohn, Y. He, Y. Liu (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp. 259–266. URL: https://aclanthology.org/2020.findings-emnlp.25. doi:10.18653/v1/2020.findings-emnlp.25.

[24] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023. arXiv:2303.18223.

[25] D. Xu, W. Chen, W. Peng, C. Zhang, T. Xu, X. Zhao, X. Wu, Y. Zheng, E. Chen, Large language models for generative information extraction: A survey, 2023. arXiv:2312.17617.

[26] Y. Zhang, Z. Chen, W. Zhang, H. Chen, Making large language models perform better in knowledge graph completion, 2023. arXiv:2310.06671.

[27] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, J.-R. Wen, Structgpt: A general framework for large language model to reason over structured data, 2023. arXiv:2305.09645.

[28] L. Yao, J. Peng, C. Mao, Y. Luo, Exploring large language models for knowledge graph completion, 2024. arXiv:2308.13916.

[29] H. Khorashadizadeh, N. Mihindukulasooriya, S. Tiwari, J. Groppe, S. Groppe, Exploring in-context learning capabilities of foundation models for generating knowledge graphs from text, 2023. arXiv:2305.08804.

[30] A. Nayak, H. P. Timmapathini, Llm2kb: Constructing knowledge bases using instruction tuned context aware large language models, arXiv preprint arXiv:2308.13207 (2023). URL: https://arxiv.org/pdf/2308.13207.pdf.

[31] Y. Zhu, X. Wang, J. Chen, S. Qiao, Y. Ou, Y. Yao, S. Deng, H. Chen, N. Zhang, Llms for knowledge graph construction and reasoning: Recent capabilities and future opportunities, 2024. arXiv:2305.13168.

[32] OpenAI, :, J. Achiam, S. A. et al., Gpt-4 technical report, 2024. arXiv:2303.08774.

[33] N. Mihindukulasooriya, S. Tiwari, C. F. Enguix, K. Lata, Text2kgbench: A benchmark for ontology-driven knowledge graph generation from text, in: International Semantic Web Conference, Springer, 2023, pp. 247–265. URL: https://arxiv.org/pdf/2308.02357.pdf.

[34] S. Pan, L. Luo, Y. Wang, C. Chen, J. Wang, X. Wu, Unifying large language models and

knowledge graphs: A roadmap, IEEE Transactions on Knowledge and Data Engineering (2024) 1–20. URL: http://dx.doi.org/10.1109/TKDE.2024.3352100. doi:10.1109/tkde.2024.3352100.

[35] H. Kim, B.-H. So, W.-S. Han, H. Lee, Natural language to sql: where are we today?, Proc. VLDB Endow. 13 (2020) 1737–1750. URL: https://doi.org/10.14778/3401960.3401970. doi:10.14778/3401960.3401970.

[36] T. Guo, H. Gao, Content enhanced bert-based text-to-sql generation, ArXiv abs/1910.07179 (2019).

[37] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, P. Fung, Survey of hallucination in natural language generation, ACM Computing Surveys 55 (2023) 1–38. URL: http://dx.doi.org/10.1145/3571730. doi:10.1145/3571730.

[38] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, Q. Guo, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey, 2024. arXiv:2312.10997.

[39] S. J. Semnani, V. Z. Yao, H. C. Zhang, M. S. Lam, Wikichat: Stopping the hallucination of large language model chatbots by few-shot grounding on wikipedia, 2023. arXiv:2305.14292.

[40] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, D. Kiela, Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. arXiv:2005.11401.

[41] Y. Shao, Y. Jiang, T. A. Kanell, P. Xu, O. Khattab, M. S. Lam, Assisting in writing wikipedia-like articles from scratch with large language models, 2024. arXiv:2402.14207.

[42] E. Huaman, E. Kärle, D. Fensel, Knowledge graph validation, 2020. arXiv:2005.01389.

[43] J. Mayfield, T. W. Finin, Evaluating the quality of a knowledge base populated from text, in: AKBC-WEKEX@NAACL-HLT, 2012. URL: https://api.semanticscholar.org/CorpusID:1851959.

[44] Y. Cao, X. Ji, X. Lv, J. Li, Y. Wen, H. Zhang, Are missing links predictable? an inferential benchmark for knowledge graph completion, 2021. arXiv:2108.01387.

[45] A. Radford, K. Narasimhan, Improving language understanding by generative pre-training, 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[46] H. Touvron, L. Martin, K. S. et al., Llama 2: Open foundation and fine-tuned chat models, 2023. arXiv:2307.09288.

[47] N. Reimers, I. Gurevych, Sentence-bert: Sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019. URL: https://arxiv.org/abs/1908.10084.

[48] A. Neelakantan, T. Xu, R. P. et al., Text and code embeddings by contrastive pre-training, 2022. arXiv:2201.10005.

[49] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, 2018. arXiv:1707.01476.

[50] T. Safavi, D. Koutra, CoDEx: A Comprehensive Knowledge Graph Completion Benchmark, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 8328–8350. URL: https://aclanthology.org/2020.emnlp-main.669.

doi:10.18653/v1/2020.emnlp-main.669.

[51] A. Mallen, A. Asai, V. Zhong, R. Das, D. Khashabi, H. Hajishirzi, When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023. arXiv:2212.10511.

[52] A. Q. Jiang, A. Sablayrolles, A. R. et al., Mixtral of experts, 2024. arXiv:2401.04088.

[53] G. Team, T. Mesnard, C. H. et al., Gemma: Open models based on gemini research and technology, 2024. arXiv:2403.08295.

[54] L. Weidinger, J. Mellor, M. e. a. Rauh, Ethical and social risks of harm from Language Models, 2021. URL: http://arxiv.org/abs/2112.04359. doi:10.48550/arXiv.2112.04359, arXiv:2112.04359 [cs].

[55] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?, in: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, FAccT '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 610–623. URL: https://dl.acm.org/doi/10.1145/3442188.3445922. doi:10.1145/3442188.3445922.

[56] M. J. Ryan, W. Held, D. Yang, Unintended Impacts of LLM Alignment on Global Representation, 2024. URL: http://arxiv.org/abs/2402.15018. doi:10.48550/arXiv.2402.15018, arXiv:2402.15018 [cs].

[57] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, C. Raffel, Extracting Training Data from Large Language Models, 2021. URL: http://arxiv.org/abs/2012.07805. doi:10.48550/arXiv.2012.07805, arXiv:2012.07805 [cs].

[58] D. Mytton, Data centre water consumption, npj Clean Water 4 (2021) 1–6. URL: https://www.nature.com/articles/s41545-021-00101-w. doi:10.1038/s41545-021-00101-w, publisher: Nature Publishing Group.

[59] D. Patterson, J. Gonzalez, Q. Le, C. Liang, L.-M. Munguia, D. Rothchild, D. So, M. Texier, J. Dean, Carbon Emissions and Large Neural Network Training, 2021. URL: http://arxiv.org/abs/2104.10350. doi:10.48550/arXiv.2104.10350, arXiv:2104.10350 [cs].

[60] Wikimedia Foundation, Wikimedia statistics - wikidata, https://stats.wikimedia.org/, ???? Accessed: 2024-03-07.

[61] Tom Jobbins, Thebloke/llama-2-70b-chat-gguf, https://huggingface.co/TheBloke/Llama-2-70B-Chat-GGUF/, ???? Accessed: 2024-03-13.

[62] G. Gerganov, gguf.md, https://github.com/ggerganov/ggml/blob/master/docs/gguf.md, 2023. Accessed: [2024-03-15].

A. Appendix

A.1. Prompt Templates

```
@staticmethod
def validate_statement_with_no_context(entity_label, predicted_property_name,
 ↪  predicted_property_value):
    '''Validate a statement about an entity with no context

    a statement is a triple: entity_label --> predicted_property_name -->
     ↪  predicted_property_value
                        e.g Donald Trump --> wife --> Ivanka Trump

    '''
    resp: ValidatedTriple = client.chat.completions.create(
        response_model=ValidatedTriple,
        messages=[
            {
                "role": "user",
                "content": f"Using your vast knowledge of the world, " +
                        "evaluate the predicted Knowledge Graph triple for its accuracy by
                         ↪  considering:\n" +
                        "1. Definitions, relevance, and any cultural or domain-specific
                         ↪  nuances of key terms\n" +
                        "2. Historical and factual validity, including any recent updates
                         ↪  or debates around the information\n" +
                        "3. The validity of synonyms or related terms of the prediction\n"
                         ↪  +
                        "Approach this with a mindset that allows for exploratory analysis
                         ↪  and the recognition of uncertainty or multiple valid
                         ↪  perspectives. " +
                        "Use this approach to recognize a range of correct answers when
                         ↪  nuances and context allow for it." +
                        "If multiple relations are provided, the triple is valid if any of
                         ↪  them are valid. " +
                        f"\nSubject Name: {entity_label}" +
                        f"\nRelation: {predicted_property_name}" +
                        f"\nObject Name: {predicted_property_value}"
            }
        ],
        max_retries=3,
        temperature=0,
        model=MODEL,
    )
    return resp
```

Figure 6: The prompt used across all experiments. The LLM response is captured as a Pydantic model.

```
class ValidatedTriple(BaseModel, extra='allow'):
    predicted_subject_name: str
    predicted_relation: Union[str, List[str]]
    predicted_object_name: str

    reason: str = Field(
        ..., description="The reason why the predicted subject-relation-object triple is or
        ↪  is not valid."
    )
    triple_is_valid: Literal[True, False, "Not enough information to say"] = Field(
      ...,
        description="Whether the predicted subject-relation-object triple is generally
        ↪  valid, following the previously-stated approach. " +
                    "If multiple relations are provided, the triple is valid if any of them
                    ↪  is valid. " +
                    "Think through the context and the nuances of the terms before
                    ↪  providing your answer. " +
                    "If the context does not provide enough information, try to use your
                    ↪  common sense."
    )
```

Figure 7: The Pydantic model which will encapsulate the LLM Validator response.

A.2. Validators with Context Illustrations

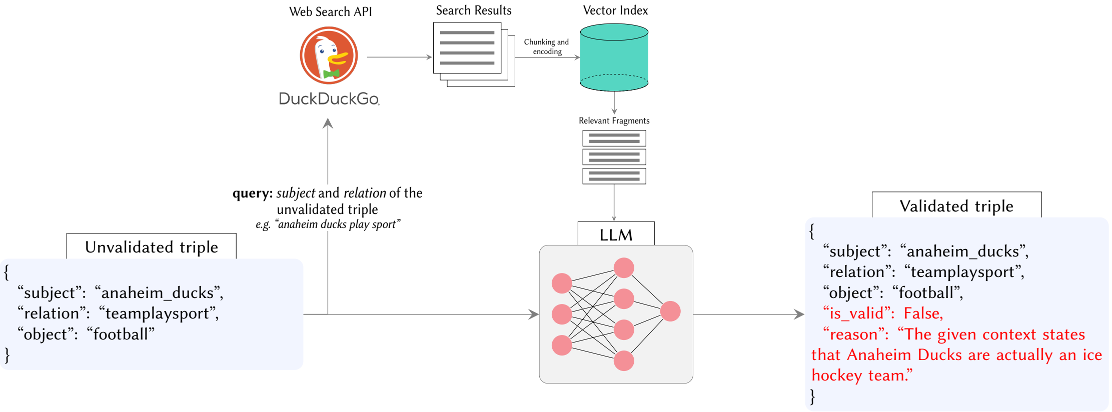

Web Search API
DuckDuckGo
Search Results
Chunking and encoding
Vector Index
Relevant Fragments
LLM
query: subject and relation of the unvalidated triple
e.g. "anaheim ducks play sport"
Unvalidated triple
{
"subject": "anaheim_ducks",
"relation": "teamplaysport",
"object": "football"
}
Validated triple
{
"subject": "anaheim_ducks",
"relation": "teamplaysport",
"object": "football",
"is_valid": False,
"reason": "The given context states that Anaheim Ducks are actually an ice hockey team."
}

Figure 8: Validating KGs using Web Search

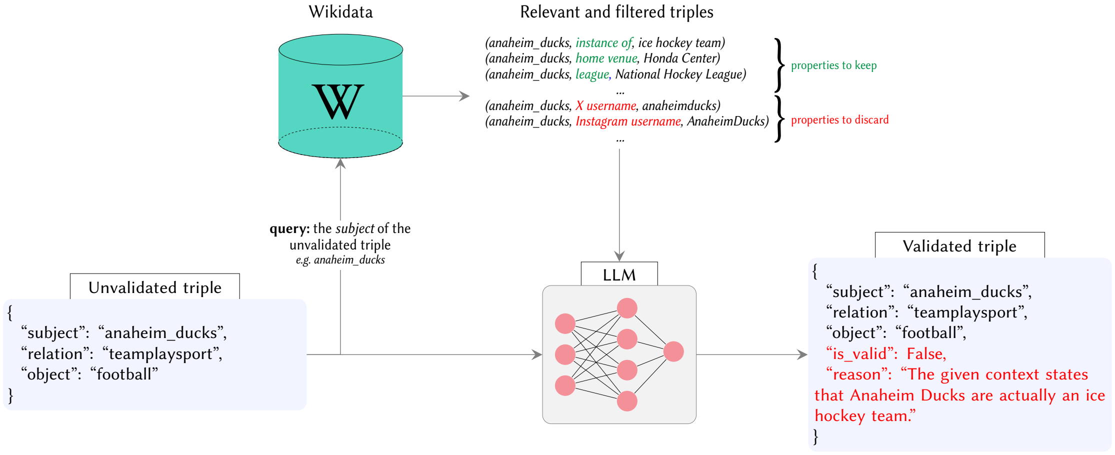

Wikidata
W
Relevant and filtered triples
(anaheim_ducks, instance of, ice hockey team)
(anaheim_ducks, home venue, Honda Center)
(anaheim_ducks, league, National Hockey League)
...
(anaheim_ducks, X username, anaheimducks)
(anaheim_ducks, Instagram username, AnaheimDucks)
...
properties to keep
properties to discard
query: the subject of the unvalidated triple
e.g. anaheim_ducks
Unvalidated triple
{
"subject": "anaheim_ducks",
"relation": "teamplaysport",
"object": "football"
}
LLM
Validated triple
{
"subject": "anaheim_ducks",
"relation": "teamplaysport",
"object": "football",
"is_valid": False,
"reason": "The given context states that Anaheim Ducks are actually an ice hockey team."
}

Figure 9: Validating KGs given a Reference KG such as Wikidata

A.3. Open-source Experimental Setup

To evaluate the capabilities of open-source LLMs in our framework, we employed a version of the LLama-2-70B-chat model [46]. We selected a model applying Q5_K_M quantization to LLama2 70b-chat, chosen for its minimal quantization levels and low reported impact on quality, provided by [61]. To implement this, we take advantage of Instructor's integration with llama-cpp-python[12], which supports quantized models in GGUF [62] format available on Hugging Face Hub[13]. Our experimental setup for open-source LLMs uses two NVIDIA A100 GPUs.

A.4. Open-source LLM Experimental Results

Table 4
Experiment results using Llama-2-70B-chat model for FB15K-237-N-150, CoDeX-S-150 and Wiki27K-150 datasets. Accuracy (Acc), precision (P), recall (R), and F1-score (F1) results for each method are reported. Scores for each dataset are separated by slashes, listed as FB15K-237-N-150/CoDeX-S-150/Wiki27K-150.

Model	P	R	F1	Acc
Llama-2 Web	0.52/0.51/0.46	1.0/1.0/1.0	0.68/0.67/0.63	0.54/0.52/0.49
Llama-2 WorldKnowledge	0.54/0.51/0.54	1.0/1.0/1.0	0.70/0.66/0.70	0.58/0.50/0.58
Llama-2 Wikidata	0.53/0.55/0.53	1.0/1.0/1.0	0.69/0.71/0.69	0.56/0.60/0.56
Llama-2 WikidataWeb	0.50/0.50/0.51	1.0/1.0/1.0	0.66/0.66/0.67	0.50/0.50/0.51
Llama-2 WikipediaWikidata	0.51/0.50/0.51	1.0/1.0/1.0	0.67/0.66/0.67	0.51/0.50/0.51

[12] https://github.com/abetlen/llama-cpp-python
[13] https://huggingface.co/docs/hub

A.5. Validator Inference Examples

```
>>> inp = [{"predicted_subject_name": "Heinrich Rudolf Hertz",
       "predicted_relation": [
        "occupation"
      ],
      "predicted_object_name": "theologian",}
    ]
>>> WorldKnowledgeKGValidator(**{'triples': inp})
>>> {
      "triple_is_valid": false,
      "reason": "Heinrich Rudolf Hertz was a German physicist who made significant
       ↪  contributions to the field of electromagnetism and is best known for his
       ↪  discovery of electromagnetic waves. There is no evidence or indication that he
       ↪  was a theologian. The predicted relation 'occupation' with the object
       ↪  'theologian' is not valid based on historical and factual information about
       ↪  Hertz's life and work.",
    }
```

Figure 10: Using LLM inherent knowledge to verify triple correctness.

```
{
  "predicted_subject_name": "Edward Norton",
  "predicted_relation": "The profession of Edward Norton is record producer",
  "predicted_object_name": "record producer",
  "triple_is_valid": true,
  "reason": "Edward Norton is indeed a record producer, as he has produced several films
   ↪  and documentaries through his production company, Class 5 Films.",
  "sources": [
    {
      "relevant_text": "['Edward Norton - IMDb Edward Norton - IMDb ![]()', 'Edward
       ↪  Norton\u2019s impact on the film industry goes beyond his on-screen performances.
       ↪  His dedication to storytelling, whether through acting, directing, or producing,
       ↪  reflects a commitment to meaningful and thought-provoking narratives.
       ↪  Norton\u2019s ability to tackle complex characters and engage with diverse
       ↪  genres has left an enduring mark on cinema, earning him a place among the most
       ↪  respected and accomplished figures in the entertainment world. As he continues
       ↪  to navigate the evolving landscape']"
    }
  ]
}
```

Figure 11: Example of Llama-2-70B-chat model's prediction, showing its lack of understanding of linguistic nuances. The model recognises that the relation is related to 'producer', and using its internal knownledge, includes in the reason that the subject "has produced several films and documentaries through his production company, Class 5 Films.", a relevant fact not mentioned in the provided context. However, the model incorrectly predicts that the triple is true, showing its lack of ability to discern between concepts like 'film producer' and 'record producer'.

```
{
  "predicted_subject_name": "Ricky Jay",
  "predicted_relation": "The gender of Ricky Jay is female",
  "predicted_object_name": "female",
  "triple_is_valid": true,
  "reason": "Ricky Jay was born as Richard Jay Potash, but he legally changed his name to
   ↪  Ricky Jay in 1982. Although he has been known to keep his personal life private, it
   ↪  is generally accepted that he identifies as male."
}
```

Figure 12: Example of Llama-2-70B-chat model's prediction, stating the correct gender and using the correct genedered pronouns in the 'reason', but failing to label the invalid triple about the subject's gender as false, exhibiting its lack of conceptual understanding of concepts like gender.