# Exploring the feasibility of ontology-based educational question generation with an LLM

**Franco Alberto Cardillo**
Institute for Computational Linguistics
National Research Council
Pisa, Italy
francoalberto.cardillo@cnr.it

**C. Maria Keet**
Computer Science Division,
Stellenbosch University
South Africa
Meaningfy S.à r.l.,
Lintgen, Luxembourg
cmkeet@sun.ac.za

**Zola Mahlaza**
Department of Computer Science,
University of Cape Town
South Africa
zmahlaza@cs.uct.ac.za

## Abstract

Educational question generation from structured knowledge, such as represented in knowledge graphs and ontologies, are typically generated by traditional NLP approaches. They face issues of template or grammar generation and the many corner cases to take into account. LLMs are also not assumed to perform well, but then due to hallucinations that are unacceptable for educational questions, and lack of training data in specialised domains of the ontologies may affect more complex question involving more and more expressive axioms. We assessed the question generation problem with 7 ontologies on 7 relatively small LLMs, all executed on GPU-enabled machines, through prompting from 0 to 10-shots on eight different types of educational questions. The first hypothesis, on LLMs producing grammatically better quality sentences but having more hallucinations than the NLP pipeline approach, was falsified, and the larger problem was the answer appearing in the generated question. The second hypothesis, on LLMs performing better with simple subsumption axioms than with 'complex' ones, was validated. We conclude that the tested LLMs are not well-suited for reliable ontology verbalisation as question generation, compared to tailored NLP pipelines.

## 1 Introduction

Educational question generation and answer explanation from ontologies has been proposed for in-context self-assessment in adaptive learning materials, alleviating manual crafting of homework and test-setting and marking of questions, which can be done for MCQs (Alsubait et al., 2016; Chaudhri et al., 2014; Kumar et al., 2018; Kurdi et al., 2020) as well as other types of educational questions (Raboanary et al., 2022a).

These systems can generate MCQ and non-MCQ questions when tailored to the ontology they are intended to be used with and have templates or grammars specified, which is time-consuming and not easily amenable to generalisation to at least multiple ontologies due to the many corner cases (Raboanary et al., 2022b) and due to unanticipated modelling or naming that can appear in the ontology. To the best of our knowledge, there are no LLMs fine-tuned for the task of ontology verbalization in general, nor for ontology-based question generation specifically (in the surface realisation step of the NLG pipeline (Reiter, 2024)). Retrofitting competency question generation from an ontology (Alharbi et al., 2024) are among the first explorations in this direction, albeit for a different task, as is the validation task a different one (Choi et al., 2026).

Our aim is to investigate the feasibility of LLM-based generation of *educational questions* from an ontology as a particular form of ontology verbalisation. The expectations are mixed and therefore our aim is to investigate the following two hypotheses:

H1: LLM-based educational question generation will have grammatically better sentences in the verbalisation, but due to 'hallucinations' be worse in conveying the semantics as represented in the axioms.

H2: The LLM-based approach performs better for the easier questions from simple short axioms, but worse for the complex ones from ontologies with larger, more expressive axioms.

We also expect to see compounding issues for question, answer, and explanation generation with the LLM-generated sentences as compared to the non-LLM baseline with rules, templates, and other NLP techniques, as the answer may not align with the question, and the explanation not with the question and answer, whereas they are more tightly integrated in the knowledge and rules-based approach. To be able to test that, however, we first need to asses H1 and H2, which are the focus of this paper.

We experiment with in-context learning of 7 LLMs on 7 ontologies, varying the number of in-context examples provided with the prompt. The



*Proceedings of the 19th International Natural Language Generation Conference: Long Papers*, pages 1236–1248
October 17-21, 2026 ©2026 Association for Computational Linguistics

results were subjected to common automated metrics and more detailed semantic analysis. Based on the results, H1 is falsified and H2 validated.

In the remainder of the paper, we present related work in Section 2, present the methods and materials in Section 3, the results in Section 4, and discuss and conclude in Sections 5 and 6.

## 2 Related work

Question generation from structured content can serve multiple purposes, and its input is typically a knowledge graph or an ontology. We focus on generating type-level questions from an ontology; e.g., "what does the rainfall reduction method involve?" rather than, say, "what is the name of the flooding site?" (an instance-level query) or "how do I prevent the sewer from being blocked?" (asks for instructions).

LLM-based automated question generation from ontologies has been pursued mainly for generating competency questions (CQs) to "retrofit" them on existing ontologies for which no CQs exist or they were not published alongside the ontology (Alharbi et al., 2024). More results have been obtained with rules-based approaches, such as question generation to validate contents (Mahlaza and Keet, 2020), multiple-choice question and answer options generation (Alsubait et al., 2016), and a rules and NLP approach to non-MCQ question, answer, and feedback generation (Raboanary et al., 2022b) or a combination of the two (Choi et al., 2026). CQs for scoping, reuse, and validation are different in structure than educational questions, although there is an overlap. For instance, "Is a giraffe a herbivore?" could be a scoping question and a primary school level review or test question. It can be generated from an ontology if it contains the vocabulary, and a 'yes' can be computed only if it contains the subsumption axiom $\mathsf{Giraffe} \sqsubseteq \mathsf{Herbivore}$. In an ontology validation mode, the absence of the axiom means the content still has to be added, whereas for an educational question generation mode, a system should only generate such a question if that axiom is present as well as a different, disjoint class so that also a 'false' in true/false can be generated, such as that the ontology also contains Carnivore and that it is asserted to be disjoint from Herbivore, or a subclass of carnivore that is also an animal.

In addition, and unlike competency questions, education questions cannot be permitted to have mistakes nor have any 'hallucinations' specifically.

Further, while for scoping, reuse, and validation of ontologies and knowledge graphs, one seeks to cover the whole actual or prospective ontology with questions in the same format for ease of checking and for ease of automated SPARQL query generation (see (Wisniewski et al., 2021)), automated generation of educational questions benefit from variety in phrasing and from 'interesting' distractors (Alsubait et al., 2016; Raboanary et al., 2022a). For instance, asking whether some particular species of animal or an instance thereof (e.g., a warthog, like Pumbaa) is a herbivore is harder to answer than a question that asks whether a safari bus is a herbivore. To achieve this, a question and an answer generally require more prerequisite knowledge in an ontology than for generating CQs. When delegating the generation to the LLM, it raises questions about fine-tuning or prompting and whether giving it multiple axioms will result in garbled output that may merge the content instead of preserving the semantics of each axiom.

Table 1: Ontologies tested in the experiments.

| Ontology | Description, type | Questions | Types |
|---|---|---|---|
| AWO (adapted) | African wildlife, tutorial | 1,153 | 8 |
| BioTop | Biological entities, top-domain | 149 | 1 |
| Copyright | Copyright, domain | 293 | 6 |
| Cultural-On | Cultural heritage, domain | 2,268 | 8 |
| Pizza | Pizzas, tutorial | 62,701 | 8 |
| Stuff | Amounts of matter, core | 340,412 | 8 |
| Olia | Linguistics, core | 68 | 4 |

## 3 Methods

The methods consist of three parts: the baseline materials, how to generate the questions with the LLMs, and how to evaluate them.

### 3.1 Baseline materials

The raw dataset we created was based on the code[1] by (Raboanary et al., 2022b) and is made up of 513,985 pairs of axioms and associated questions. For instance, the African Wildlife Ontology axiom `CarnivorousPlant SubclassOf(eats some Animal)` is paired with the question "Does a carnivorous plant eat some animal?". The axioms are sourced from seven ontologies expressed in OWL (Motik et al., 2009), which are described in Table 1. For each pair, one can recover the "question type"

[1] https://github.com/mkeet/AQuestGO

Table 2: Examples of all eight types of educational questions (of varying quality, from different ontologies), which were automatically generated by the NLP pipeline of (Raboanary and Keet, 2022).

| Axioms | Question | Question Type |
|---|---|---|
| ProcessQuality SubclassOf(Particular) | What is a process quality? | Definition |
| Perform SubclassOf(result only Performance) | What does a perform result? | op_What-1-part-1-rel |
| Gel SubclassOf(hasQuale only Solid), Gel SubclassOf(DispersionColloid) | Which dispersion colloid has a quale that is being a solid? | op_What-2-part-1-rel |
| Use SubclassOf(Action),Use SubclassOf(agent only LegalPerson) | Which disagree action has an agent that is only a legal person? | op_What-2-part-1-rel-quant-only |
| Warthog SubclassOf(eats some PlantParts),Warthog SubclassOf(Omnivore) | Which omnivore eats some plant parts? | op_What-2-part-1-rel-quant-some |
| OnionTopping SubclassOf(hasSpiciness only ValuePartition) | An onion topping has a spiciness that is a value partition. True or false? | op_Yes-No-2-part-1-rel |
| Impala SubclassOf(Live-on only Land) | True or false: An impala lives on only land. | op_Yes-No-2-part-1-rel+1-quant-only |
| Fiorentina SubclassOf(hasIngredient some Food) | A fiorentina has an ingredient that is some food. True or false? | op_Yes-No-2-part-1-rel+1-quant-some |

since that information is automatically generated by an NLP pipeline, when a set of prerequisites for the input axioms is satisfied. Examples of questions are shown in Table 2. There are at least 10 questions for each question type in every ontology.

Not all generated questions are of good quality, because there is more variability in the source ontologies than covered in the NLP pipeline. For instance, the vocabulary element 'result' in $\mathsf{Perform} \sqsubseteq \forall \mathsf{result}.\mathsf{Performance}$ is ambiguous: is it *has as result* or *result in*? This cannot be second-guessed in isolation, nor can all such corner cases be anticipated, which results in some grammatically sub-optimal questions. This is likewise for naming classes 'in context' versus a full name; e.g., American in the Pizza ontology is a subclass of Pizza and understood as such by a human looking at the hierarchy, but when it generates a question "Does each American have tomato as topping?", it raises eyebrows in a human evaluator. Refer to (Raboanary and Keet, 2022) for more issues.

Thus, the baseline is not perfect, and precisely a reason to investigate LLM-based educational question generation.

### 3.2 Generating questions: the input

This study investigates the use of LLMs for generating questions in natural language from OWL axioms, using the questions generated by a standard NLP pipeline of (Raboanary and Keet, 2022) (described in the previous section) as ground truth. The task is organised as an in-context learning problem, without any model fine-tuning. The LLM is prompted at 'inference time' with a set of desired input-output examples, hereafter referred to as few-shot examples.

Determining input and output examples for the LLM is less straight-forward as it may seem. First, regarding content input, OWL ontologies can be serialised in five official syntaxes and one can use or create unofficial notations, such as LaTeX or JSON, and it is unknown which one would be best for this task. For instance, $\mathsf{Pipe} \sqsubseteq \mathsf{Conduit}$ in functional style syntax is a compact

```
SubClassOf(SewerNet:Pipe SewerNet:Conduit)
```

but in the official exchange syntax RDF/XML it is a wordy:

```
<owl:Class rdf:about="http://sewernet.hsm.
   umontpellier.fr/SewerNet#Pipe">
  <rdfs:subClassOf rdf:resource="http://sewernet.
   hsm.umontpellier.fr/SewerNet#Conduit"/>
</owl:Class>
```

Second, it may or may not benefit from annotation of question type name and/or the templates belonging to the question type. Third, each question type has 1-7 template variants in the surface realisation, raising the question whether the LLM should be tested on generating all of them or a subset of the sentence structures. We decided against including the question template, because it is precisely one aspect one wants to avoid specifying manually.

Regarding output, we use texts generated by the NLP pipeline as ground truth. Although fine-tuned rules generated these texts, we cannot exclude that some of them are malformed and that the LLM might generate well-formed ones or replicate the malformed-ness. The alternative of manually written and checked sentences would be too many to do and precisely a problem we wish to solve, and there-

```
You are a helpful assistant that generates questions in natural language based on OWL axioms provided.

Based on the following OWL axiom, generate a question:
Axiom: Universe SubclassOf(Condition)
Question: What is universe?

Based on the following OWL axiom, generate a question:
Axiom: ThinAndCrispyBase SubclassOf(PizzaBase)
Question: What is a thin and crispy base?

Based on the following OWL axiom, generate a question:
Axiom: Grass SubclassOf(Plant)
Question:
```

Figure 1: Prompt example for $k = 2$. The first two questions are examples from the training ontologies, while the last axiom belongs to the test ontology and the LLM is asked to complete the text after "Question:".

fore not an option. A prompt example for $k = 2$, where $k$ is the number of in-context examples, is shown in Fig. 1.

### 3.3 Question generation: protocol

The experimental protocol is based on an ontology-based cross-validation, where each ontology is used once as *test* ontology and six times as *training* ontology. For each iteration (fold), the LLM is asked to generate questions for all the test axioms, i.e., relevant axioms extracted from the test ontology, based on $k \geq 0$ examples sampled from the training ontologies with the same question type. This works well for all ontologies except BioTop, for which only "Definition" type of question can be generated, and therefore the few-shot examples for the other nine question types do not (and cannot) contain question examples drawn from BioTop content.

As the number of pairs in the ontologies ranges from 68 to 340,412 (see Table 1), in order to balance the training examples and avoid the inclusion of many examples from the largest ontologies, we sample 10 questions per question type from each ontology, for a total of 430 examples.

We experimented using a number of examples $k \in \{0, 1, 2, 3, 6, 10\}$. When $k$ is greater than the number of training ontologies $k'$ with the correct question type, we first select one example from each of the $k'$ ontologies and then we sample $k - k'$ from all the $k'$ ontologies, with replacement. Then they are used in a prompt as illustrated in Fig. 1.

As $k$ increases across the experiments, the few-shots examples are included incrementally; e.g., when $k = 6$, the first three examples used in the prompt corresponds to the examples in the experiment with $k = 3$. Each experiment is repeated 10 times with different random seeds for sampling both the test pairs of ⟨axiom, question⟩ and the in-context example pairs. The LLM generation is greedy in order to ensure deterministic and repeatable results.

Table 3: LLMs used in the experiments.

| Family | Model ID | Param. | Type |
|---|---|---|---|
| Gemma | Gemma-2-9B-it | 9B | decoder-only |
| Llama | Llama3.1-8B-instruct | 8B | decoder-only |
| Mistral | Mistral-7B-instruct | 7B | decoder-only |
| Qwen | Qwen2.5-7B-instruct | 8B | decoder-only |
| | Qwen2.5-32B-instruct | 32B | |
| T5 | FLAN-T5-XXL | 11B | encoder-dec. |
| Phi | Phi-4 | 14B | decoder-only |

Table 3 shows the LLMs included in the experiments, which correspond to instruction-tuned versions of the corresponding base model, initialised with the pre-trained weights from Hugging Face[2]. Their sizes range from seven to 14 billion parameters, with the latter corresponding to the maximum that could be accommodated by our available computing resources without quantisation.

### 3.4 Evaluation

The main means of evaluation is automated evaluation, which is augmented with human analysis.

#### 3.4.1 Common automated evaluation metrics

For each axiom, the generated question is compared against the corresponding reference question produced by the NLP pipeline, used as ground truth. We use standard evaluation measures in NLP as computed by established software libraries. We use the following lexical and semantic measures. *Exact Match (EM)* checks if the generated question exactly matches the reference question, with EM $\in \{0, 1\}$. *BLEU* (Papineni et al., 2002): BLEU-$n$ measures the overlap of $n$-grams between the generated and reference texts. We report BLEU-3 and BLEU-4 scores, computed using the SacreBLEU library (Post, 2018), with BLEU-$n \in [0, 1]$. *ROUGE* (Lin, 2004) evaluates the overlap between generated and reference texts with an emphasis on recall. We report ROUGE-L, which measures the length of the longest common subsequence, using the implementation by Google Research[3], with ROUGE-L $\in [0, 1]$. *BERTScore* (Zhang et al., 2019) assesses semantic similarity by comparing contextualized embeddings obtained from a pre-trained BERT model. We use the implementation

[2] https://huggingface.com

[3] https://pypi.org/project/rouge-score/

provided by the authors[4], using its default settings and, specifically, the 'roberta-large' model. We use the $F_1$ score computed by BERTScore, which has range $[0, +1]$.

We compute these scores for all test questions, for each cross-validation fold and for all the models in Table 3. The models are compared using a composite score $\mathcal{S}$ defined for individual questions that combines the lexical and semantic similarity between the reference and the LLM-generated question: $\mathcal{S} = \alpha \cdot \text{ROUGE_L} + \beta \cdot \text{BERTScore_F}_1$, with $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$. Unless otherwise stated, we report $\mathcal{S}$ computed with $\alpha = 0.5$, $\beta = 0.5$.

We assess the overall performance of each model in terms of mean score $\mathcal{S}$ and mean rank. For each model, test ontology, value of $k$, and random seed, we first average the scores over the generated questions. To obtain the overall score, these values are then averaged over $k$ and test ontologies, and finally across random seeds. To compute the mean rank, the models are first ranked within each combination of random seed, $k$ and test ontology based on the pre-computed average scores over questions, with rank 1 corresponding to the best model. The resulting ranks are then averaged over $k$ and test ontologies, and finally across random seeds.

#### 3.4.2 Measuring grammaticality

We evaluated the grammatical quality of the resulting questions in a semi-automated manner. The overall goal is to determine whether the LLM-generated questions are grammatically better than the questions generated by the pipeline. We randomly selected 1000 questions produced by the pipeline and the LLMs, and made use of CoEdIT (Raheja et al., 2023) to correct the grammar of both types of questions. The first subgoal is to establish whether the training questions and the generated questions have a small number of changes that need to made to ensure that they are of high grammatical correctness. We measured the edit distance between the input and the corrected text and the average number of changes to be made to each question to improve its quality, and, second, from that, computed whether there is a significant difference between the two sets of questions, with respect to improving their grammatical quality.

#### 3.4.3 Automatically detecting 'hallucinations'

We analysed the questions on hallucinations with respect to the vocabulary used in the axioms. This focused on entity-related hallucinations by measuring whether the generated questions did not include more entities that those found in the axioms. This was measured by extracting the number of entities in the axiom and comparing them vs. those found in generated question using Llama3 8B-Instruct on 50 randomly extracted questions. The same texts were also manually evaluated by another collaborator, different from the extractor, by annotating them with yes/no and reasons for when the generated question was not acceptable. Since this was carried out by one person, no true inter-annotator agreement was computed. We did compare sentences flagged by the human and Llama3.

#### 3.4.4 Manual semantic analysis

Neither the automated metrics nor the the automated hallucination detection can assess whether the LLM-generated sentences are semantically correct, acceptable, borderline, or false with respect to the axioms that need to be verbalised.

As a first step towards the final evaluation, we checked the sentences generated from the training data to extract issues and to determine a threshold for acceptable output based on the combined lexical and semantics score $\mathcal{S}$. One collaborator selected 30 pairs of axioms with generated sentences that had a combined score between 0.70 and 0.85, which were annotated on semantic accuracy and type of issue by another collaborator. Based on this analysis, we set the threshold at $\theta = 0.80$ and classify questions with score $\mathcal{S} \geq \theta$ as "good" LLM-generated questions.

## 4 Results

We first present the results of the automated metrics to determine the best LLM, then grammaticality, hallucinations, and challenges observed.

### 4.1 Automated metrics

The results are presented in two parts: (i) the complete set of generated questions, providing a global assessment of the model performance across all the random seeds, the test ontologies and the number $k$ of in-context examples used in the prompt; (ii) a subset of generated questions containing only those whose score exceeds a quality threshold.

Fig. 2 shows the overall average score $\mathcal{S}$ for all the models, computed as described in Section 3. The models Phi-4 and FLAN-T5-XXL reach the best and worst performance, respectively, while the others have an approximately similar performance.

[4] https://github.com/Tiiiger/bert_score

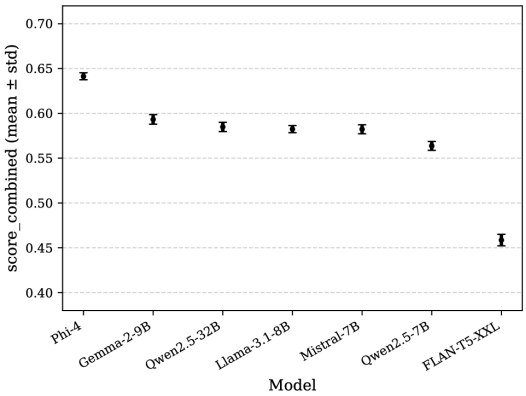
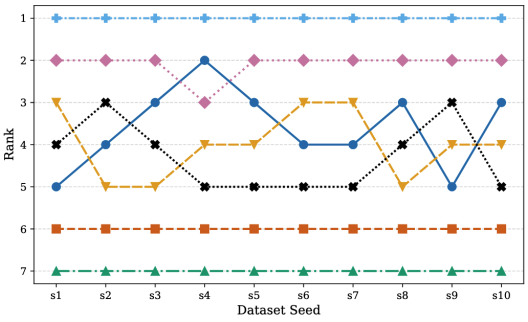


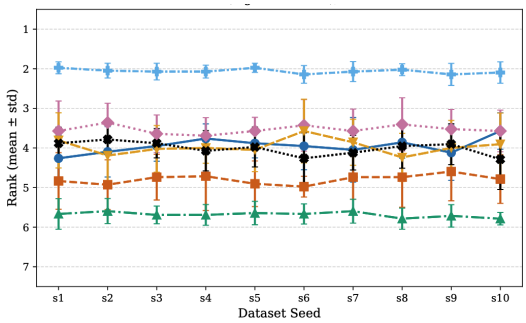

Figure 2: Overall combined score $\mathcal{S}$, hierarchically averaged across seeds, ontologies, and number of examples.

Rather than using the scores to assess statistically significant differences in text generation, we rely on rank-based tests within a complete block design, with seeds acting as blocks and models as treatments. To assess how robust the model performance is across the different datasets, Fig. 3a reports the ranking of the models for each random seed, based on their corresponding overall scores. We then average these ranks across the random seeds and test whether the differences among models are statistically significant, using the Friedman statistics (Demšar, 2006), obtaining $Q = 55.41$ with $p = 3.82 \times 10^{-10}$. Since there are only 10 groups (random seeds), we performed also a randomized permutation test based on 100,000 iterations. In this second test, the model ranks are first independently shuffled within each seed, then used to compute the Friedman statistics $Q_i$ for each iteration $i$. The permutation-based $p$-value is estimated as $p_{\text{perm}} = \frac{1+|\{i.Q_i \geq Q\}|}{1+N}$, where $N = 100,000$. The permutation-based test confirmed statistically significant differences with $p_{\text{perm}} = 1 \times 10^{-5}$. To establish which pairs of models differ statistically we used a post-hoc pair-wise Nemenyi test, based on the average ranks of the model across the seeds, shown in Fig. 8. Two models are considered significantly different when their average ranks differ more than a Critical Difference (CD), whose value for the top ranked model corresponds to the height of the gray area in Fig. 8.

The previous analyses are based on an overall score resulting from an average over all the experimental combinations and dataset seeds. Fig. 3a shows the average rank computed by first ranking the model for each value of $k$, then averaging. This alternative averaging leads to the same order as shown in Fig. 3a, but with less pairwise differ-




(a) Model ranked on the average score by random seed




(b) Model average rank by random seed, with models first ranked for each value of $k$




Figure 3: Models ranked.

ences that are statistically significant (Friedman $Q = 55.567, p = 3.56 \times 10^{-10}$), as highlighted by the gray area in Fig. 9 relative to the top ranked model.

To assess the effect of the ontology on the model performance, we compare the combined score $\mathcal{S}$ across all the test ontologies. First, we characterize how difficult each ontology is by plotting the average score of each model in Fig. 4, where it is apparent that the difficulty is broadly consistent across ontologies, with the possible exception of CopyrightAll. However, a Friedman test on the model ranks using the ontologies as blocks and the models as treatments does not detect any significant difference ($Q = 1.028, p = 0.985$).

Concerning the number $k$ of few-shots examples in the prompt, Fig. 5 reports the mean combined score $\mathcal{S}$ for each model and value of $k$. The performance increased as more examples are added, with large gains occurring within the first three examples. Since the prompts were built incrementally, we used a Friedman test for each model, treating each ontology and dataset seed pair as a block. When the Friedman test proved significant, a post-hoc Nemenyi test was used to identify the pairs of $k$ that differed. Overall, $k$ significantly affects

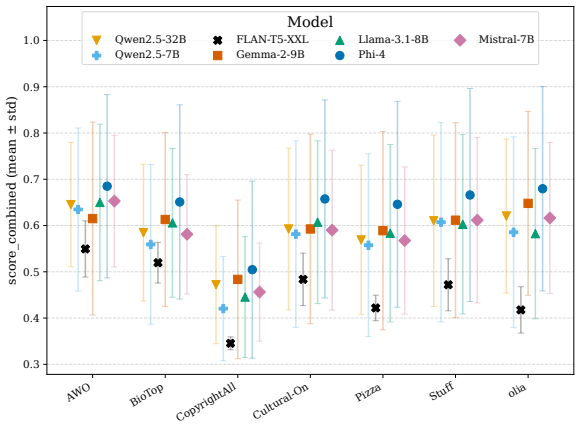



Figure 4: Average $\mathcal{S}$ per ontology.

the performance: except for a few cases and for a limited subset of models, the differences among all the pairs tested significant.

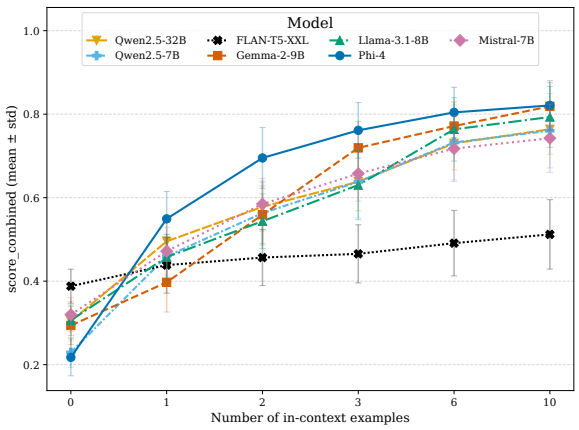



Figure 5: Average $\mathcal{S}$ for each value of $k$, by model.

The exact-match scores are low across all models, indicating a very low performance under this evaluation criterion. The average performance never exceeds 0.29, irrespective of the number of examples provided to the models. FLAN-T5-XXL is notable for not demonstrating a noticeable increase in performance with the increase in the number of examples. When there are no examples, the model has the highest average (i.e., 0.03) but this performance gap is quickly outdone by the decoder-only models when there are two or more examples provided since the performance of the other models increases at a faster rate (see Fig 5).

The models' performance via ROUGE, a more flexible metric vs. the all-or-nothing exact match metric, demonstrates that the models perform well and better when the number of examples provided increases. Similar to the exact match, the decoder-only models show the greatest increase in performance with more examples, with FLAN-T5-XXL the smallest increase; e.g., FLAN-T5-XXL starts with an average of 0.39 and it only increases to 0.5, at best, when the number of examples increases. To demonstrate the utility of more examples for decoder-only models, Gemma-9b starts at a lower baseline of 0.27 (zero-shot) and its performance climbs to an average of 0.83 when provided with 10 examples.

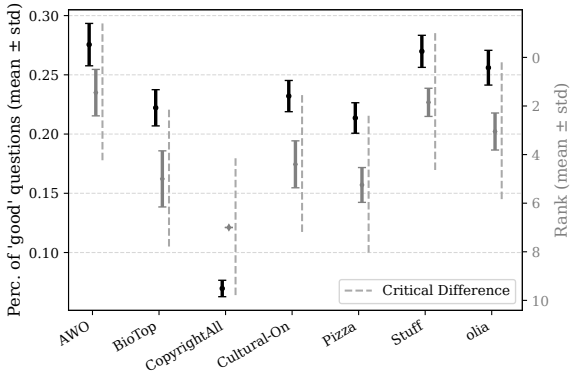


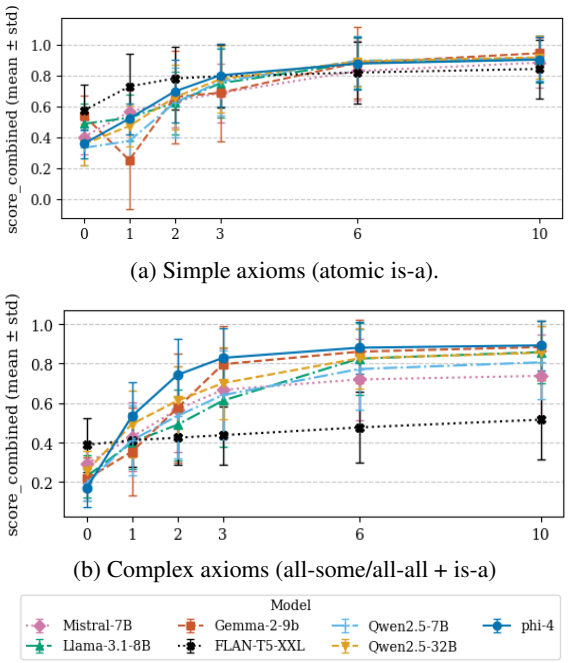

Figure 6: Pct. of "good" outputs and ranks by ontology

To assess whether the models and the ontologies differ significantly when considering only "good" outputs, we compared them in terms of the average ranking across dataset seeds using a Friedman test, resulting in statistically significant differences. Fig. 6, supported by Fig. 10, report the average percentage of good generated questions on the primary y-axis and the average rank across seeds on the secondary y-axis. The two figures also show the critical differences used in the post-hoc Nemenyi test, that allow to identify statistically significant pair-wise differences.

Fig. 7 shows the mean combined score $\mathcal{S}$ by axiom complexity (simple, intermediate, complex). Across all three levels, the largest gains still occur within the first three examples. Notably, FLAN-T5-XXL improves at a similar, if not larger, rate only on simple axioms and levels off quickly on intermediate and complex ones vs. the other models.

## 4.2 Grammaticality and hallucinations

With regards to grammaticality, we observed that the average number of changes required to improve the pipeline-generated question is 3 edits while it is 5 for LLM-generated questions. We were able to establish that the number of changes required to ensure that both question types are grammatically correct are not normally distributed. A Wilcoxon signed-rank test confirmed that there is a signif-




(a) Simple axioms (atomic is-a).

(b) Complex axioms (all-some/all-all + is-a)

Figure 7: Performance by simple versus complex axioms, clearly affecting FLAN-T5-XXL negatively and Phi-4 not.

icant difference between the two question types (with test statistic = 30 912, p-value $\leq 0.001$, and $\alpha = 0.05$). Of the sampled questions, 5% of the questions generated by LLMs required more than 10 edits to improve their quality when compared to the pipeline questions.

Regarding hallucinations, we found that LLMs generally do not hallucinate entities. On average, the axioms contain 2.6 entities, whereas the pipeline and LLM use fewer (1.9 and 1.6 respectively). This difference is statistically significant (established via a Mann-Whitney U test, p = 0.0005 for pipeline text and p < 0.000003 for the LLMs). Even so, human evaluation of the sample of 50 texts shows that 40% of the LLM generated texts are not semantically valid and 7 (14%) exhibited hallucinations, such as adding *levels* of spiciness and of specificity, and *performing* an action versus a pay condition.

### 4.3 Challenges

A manual reading through a section of the LLM-generated sentences revealed subtle problematic issues that are non-trivial to solve. For instance, an original sentence was "Which insect participates in aviating?" and Mistral generated "Which insect participates in aviation?", which had a ROUGE score of 1: the aviating from the ontology should have been 'flying', but 'aviation' certainly is wrong in context, because in the ontology it does not relate to insects splattering on the cockpit windscreen. The American that should have been named 'American Pizza' (recall Section 3.1) is another example of infelicitous naming of entities, which is beyond control of LLM-based generation other than offering its own autocomplete, such as llama's "What type of topping is typically found in an American dish?". Likewise regarding missing context and the Pizza ontology, a text was "What is a fruit topping?", which is too vague: does it ask for a definition, and topping what, or does it ask for some fruit, and if so, how could the user know the answer according to the Pizza Ontology is SultanaTopping (according to Italian recipes, pizzas cannot have fruit as topping)?

An LLM may generate a better-looking sentence that does not score well on similarity. For instance, the input `Rent SubclassOf(condition some PayAction)`: the LLM-generated one was "What condition must be met in order for a rent to be considered a pay action?", but the problem here is that the answer—a pay action—is incorporated in the question, so that it asks for another condition beyond the pay action, whereas the pay action is the condition it should have asked for. Other examples exhibiting the answer-in-the-question problem are "Which plant parts are leaves?" cf. an intended "Are leaves parts of plants?" (generated from `Leaf SubclassOf(PlantParts)`) and the "What type of pizza base is a deep pan base?" for `DeepPanBase SubclassOf(PizzaBase)` instead of having selected a different question type, like a true/false or yes/no question, alike a "Is a deep pan base a pizza base?" from the NLP pipeline, and blatant inclusion of the answer with 'Answer' in the question, such as the "What is the relationship between a plant part and a plant? (Answer: A plant part is a part of a plant.)". Simply checking for number of entities in the axioms and in the generated question will not catch all such issues: all entities are present in the true/false and yes/no questions, and even inclusion of the answer might still be borderline acceptable, as for `Cinema SubclassOf(CulturalEntity)` from which "What is a cinema? Is it a type of cultural entity?" was generated. More entities in the question than in the axioms is an indication of a possible hallucination, as is an entity with a noun that does not appear in the axiom. For instance, for `Sol SubclassOf(DispersionColloid)` (from Stuff)

mistralai generated "Is a solar system (Sol) a type of dispersion colloid?".

We checked the prevalence of issues in a random selection of 30 sentences that had a combined score between 0.7 - 0.85 to determine the cut-off point for acceptable output, and only 12 were deemed semantically acceptable. Twelve generated questions had the answer incorporated in the question, three had grammatical issues, and three were too vague in the sense of missing context (see supplemental material for details).

## 5 Discussion

Given the results obtained, we now can return to the hypotheses posed in the Introduction. The LLM-generated sentences had more edits than the pipeline-generated questions, falsifying the first part of H1. The LLMs did hallucinate some words, but the manual inspection revealed the more prevalent, and problematic, issue of including the answer in the question, despite it certainly not being given in the prompts as such.

Second, hypothesis H2 on that the LLMs perform better for the simple axioms of the pattern `A SubclassOf(B)` compared to `A SubclassOf(R only B)` and to `A SubclassOf(R some B), A SubclassOf(B)`: the results show clearly that H2 is validated for 0-shot and 10-shot, and especially so for FLAN-T5-XXL on multiple shots for medium and complex axioms (recall Fig. 7 in Section 4.1). This limits the prospects for robust in-context LLM-based feedback generation, which typically use at least as much, or more, axioms.

A limitation is the quality of the ground truth, as explained in Section 3.1. The alternatives to the one chosen were considered worse. Manually checking and rewriting the 210000 generated sentences would not only take too much time, a logician fluent in English would have to do it in order to ensure semantic accuracy, which is therefore not easily crowdsourced either and would then still need a round of verification on semantic accuracy. Refining the input, i.e., adjusting the vocabulary of the ontology, is more realistic in task execution, but it incurs the costs of greatly limiting generalisability of the approach. Ontologies use different naming practices and not always good ones, which is unlikely to change, and a verbalisation system must be able to handle that.

As mentioned in the introduction, if the question generation reaches as an acceptable quality, for example through a fine-tuning step of the LLMs reserved for future work, the first aim is to extend it to feedback generation, for which the content determination step exists but surface realisation is rudimentary (Raboanary and Keet, 2026). Given the results obtained here, while it cannot suffer from leaking answers, it will from increased complexity because, on average, more axioms are part of the feedback. The second aim is to do this multilingually in at least French, Italian, Arabic using multilingual instruction-tuned models such as Gemma 3, Qwen 2.5 or Aya Expanse, and, once the translation is complete, also isiXhosa for the domain ontology developed in one of the projects of the authors.

## 6 Conclusions

The results showed that the Phi-4 LLM performed best in generating educational questions from ontologies among the 7 LLMs evaluated, albeit not good. Grammar of the LLM-generated educational questions appeared not significantly better, and the text suffered more from answer-in-the-question than from hallucinations. Question generation from simple is-a axioms exhibited better performance than more complex axiom sets.

There are a number of options to give LLMs another chance, such as fine-tuning the models, prompting large and top-of-the-range expensive LLMs, by varying the prompt text and the input format, as well as modifying the ontologies to one particular naming scheme.

## Acknowledgments & Disclaimer

We are grateful for the assistance by Toky Raboanary in extracting the axioms from the ontologies and generating the ground truth sentences. This work was financially supported in part by the National Research Foundation (NRF) of South Africa (Grant Number and CPRR23040389063). This research was also partially supported by the EU H2020 project STARWARS (GA No. 101086252), a type of action HORIZON TMA MSCA Staff Exchanges. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, which can not be held responsible for them. We gratefully acknowledge the HPC4AI Center at the University of Turin, Italy, for providing the computational resources used to conduct the experiments.

## Ethical Considerations

There are no ethical considerations other than the myriad of issues with the use of LLMs.

## Supplemental materials

The supplementary materials are available from https://github.com/mkeet/AQuestGO/tree/master/LLMgeneratedQs

## References


Reham Alharbi, Valentina Tamma, Floriana Grasso, and Terry Payne. 2024. An experiment in retrofitting competency questions for existing ontologies. In *Proceedings of the 39th ACM/SIGAPP Symposium on Applied Computing*, pages 1650–1658.

Tahani Alsubait, Bijan Parsia, and Ulrike Sattler. 2016. Ontology-based multiple choice question generation. *KI - Künstliche Intelligenz*, 30(2):183–188.

Vinay K. Chaudhri, Peter E. Clark, Adam Overholtzer, and Aaron Spaulding. 2014. Question generation from a knowledge base. In *Proceedings of the 19th International Conference on Knowledge Engineering and Knowledge Management, EKAW 2014*, volume 8876 of *LNAI*, pages 54–65, Linköping, Sweden. Springer.

Hyojun Choi, Seokju Hwang, and Kyong-Ho Lee. 2026. Vspo: Validating semantic pitfalls in ontology via llm-based cq generation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI'26)*, volume 40, pages 18977–18984.

Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Vishwajeet Kumar, Kireeti Boorla, Yogesh Meena, Ganesh Ramakrishnan, and Yuan-Fang Li. 2018. Automating reading comprehension by generating question and answer pairs. In *Proceedings (Part III) of the 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD 2018*, volume 10939 of *LNAI*, pages 335–348, Victoria, Australia. Springer.

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A systematic review of automatic question generation for educational purposes. *International Journal of Artificial Intelligence in Education*, 30(1):121–204.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Zola Mahlaza and C. Maria Keet. 2020. OWLSIZ: An isiZulu CNL for structured knowledge validation. In *Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, pages 15–25, Dublin, Ireland (Virtual). Association for Computational Linguistics.

Boris Motik, Peter F. Patel-Schneider, and Bijan Parsia. 2009. OWL 2 web ontology language structural specification and functional-style syntax. W3c recommendation, W3C. http://www.w3.org/TR/owl2-syntax/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Toky Raboanary, Steve Wang, and C Maria Keet. 2022a. Generating answerable questions from ontologies for educational exercises. In *Proc of MTSR'21*, volume 1537 of *CCIS*, pages 28–40. Springer.

Toky Hajatiana Raboanary and C. Maria Keet. 2022. An architecture for generating questions, answers, and feedback from ontologies. In *Metadata and Semantic Research - 16th Research Conference, MTSR 2022, London, UK, November 7-11, 2022, Revised Selected Papers*, volume 1789 of *Communications in Computer and Information Science*, pages 135–147. Springer.

Toky Hajatiana Raboanary and C. Maria Keet. 2026. Ontology-mediated framework for generating answerable questions and feedback from ontologies. In *Proceedings of the 23rd European Semantic Web Conference (ESWC'26)*, volume 16549 of *LNCS*, pages 120–138. Springer. Dubrovnik, Croatia, May 10-14, 2026.

Toky Hajatiana Raboanary, Steve Wang, and C Maria Keet. 2022b. Towards the generalisation of the generation of answerable questions from ontologies for education. *International Journal of Metadata, Semantics and Ontologies*, 16(1):86–103.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.

Ehud Reiter. 2024. *Natural Language Generation*. Springer Nature.

Dawid Wisniewski, Jedrzej Potoniec, and Agnieszka Lawrynowicz. 2021. Seequery: An automatic method for recommending translations of ontology competency questions into SPARQL-OWL. In *CIKM*

*'21: The 30th ACM International Conference on Information and Knowledge Management*, pages 2119–2128. ACM. Virtual Event, Queensland, Australia, November 1 - 5, 2021.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

# A Appendix

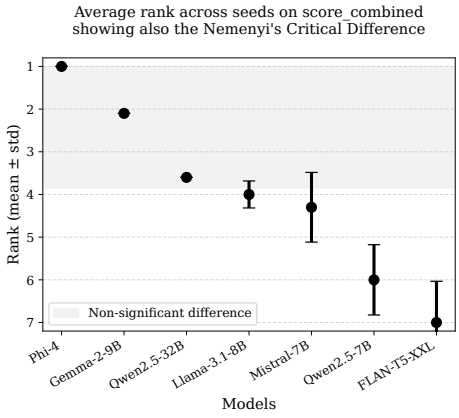



Figure 8: Average ranks across the random seeds. The shaded gray areas represent the range of non-significant differences with respect to the top ranked model when all the models or only the top-2 models are included in the experimental groups: any model whose average rank falls within that range cannot be considered statistically different to the top one.

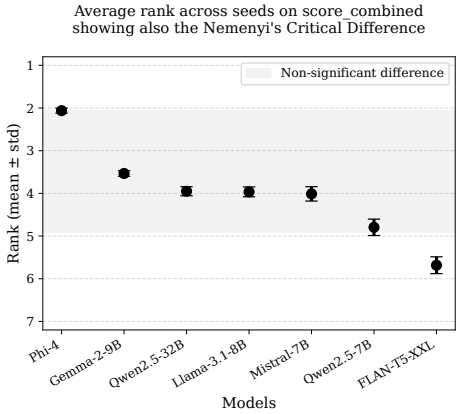



Figure 9: Average rank over seeds, with models first ranked for each value of $k$ with shaded areas corresponding to non-significant pairwise differences with respect to the best model.

Tables 4-7 contain the raw scores and the figures a visualisation thereof, which served as input to calculate the results shown in Figures 2-6.

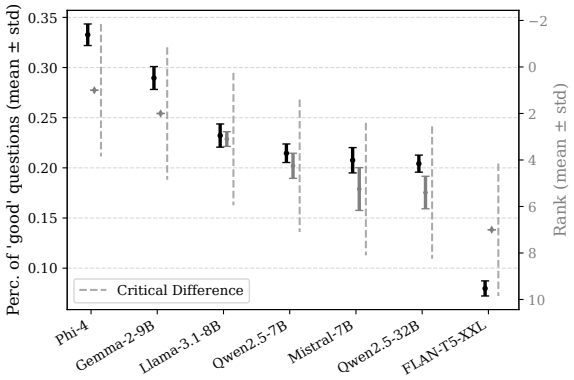



Figure 10: Pct. of "good" outputs and ranks by model

Table 4: Combined score (mean $\pm$ standard deviation) for each model across the different values of $k$

| | 0 | 1 | 2 | 3 | 6 | 10 |
|---|---|---|---|---|---|---|
| Qwen/Qwen2.5-32B-Instruct | $0.30 \pm 0.01$ | 0.50 | $0.58 \pm 0.01$ | $0.64 \pm 0.01$ | $0.73 \pm 0.01$ | $0.76 \pm 0.01$ |
| Qwen/Qwen2.5-7B-Instruct | $0.23 \pm 0.01$ | $0.46 \pm 0.01$ | $0.56 \pm 0.01$ | $0.64 \pm 0.01$ | $0.73 \pm 0.01$ | $0.76 \pm 0.01$ |
| google/flan-t5-xxl | $0.39 \pm 0.01$ | $0.44 \pm 0.01$ | $0.46 \pm 0.01$ | $0.47 \pm 0.01$ | $0.49 \pm 0.01$ | $0.51 \pm 0.01$ |
| google/gemma-2-9b-it | $0.29 \pm 0.01$ | $0.40 \pm 0.01$ | $0.56 \pm 0.01$ | $0.72 \pm 0.01$ | $0.77 \pm 0.01$ | $0.82 \pm 0.01$ |
| meta-llama/Llama-3.1-8B-Instruct | 0.31 | $0.46 \pm 0.01$ | $0.54 \pm 0.01$ | $0.63 \pm 0.01$ | $0.76 \pm 0.01$ | 0.79 |
| microsoft/phi-4 | 0.22 | $0.55 \pm 0.01$ | $0.70 \pm 0.01$ | $0.76 \pm 0.01$ | $0.80 \pm 0.01$ | 0.82 |
| mistralai/Mistral-7B-Instruct-v0.3 | 0.32 | $0.47 \pm 0.01$ | $0.58 \pm 0.01$ | $0.66 \pm 0.01$ | $0.72 \pm 0.01$ | $0.74 \pm 0.01$ |

Table 5: Rouge-L score (mean $\pm$ standard deviation) for each model across the different values of $k$

| | 0 | 1 | 2 | 3 | 6 | 10 |
|---|---|---|---|---|---|---|
| Qwen/Qwen2.5-32B-Instruct | $0.27 \pm 0.01$ | $0.48 \pm 0.01$ | $0.57 \pm 0.01$ | $0.64 \pm 0.01$ | $0.74 \pm 0.01$ | $0.78 \pm 0.01$ |
| Qwen/Qwen2.5-7B-Instruct | $0.22 \pm 0.01$ | $0.44 \pm 0.01$ | $0.56 \pm 0.01$ | $0.64 \pm 0.01$ | $0.75 \pm 0.01$ | $0.78 \pm 0.01$ |
| google/flan-t5-xxl | $0.39 \pm 0.01$ | $0.43 \pm 0.01$ | $0.45 \pm 0.01$ | $0.45 \pm 0.01$ | $0.48 \pm 0.01$ | $0.50 \pm 0.01$ |
| google/gemma-2-9b-it | $0.27 \pm 0.01$ | $0.41 \pm 0.01$ | $0.57 \pm 0.01$ | $0.73 \pm 0.01$ | $0.79 \pm 0.01$ | 0.83 |
| meta-llama/Llama-3.1-8B-Instruct | 0.28 | $0.44 \pm 0.01$ | $0.53 \pm 0.01$ | $0.63 \pm 0.01$ | 0.78 | 0.81 |
| microsoft/phi-4 | 0.21 | $0.54 \pm 0.01$ | $0.70 \pm 0.01$ | $0.77 \pm 0.01$ | $0.82 \pm 0.01$ | 0.84 |
| mistralai/Mistral-7B-Instruct-v0.3 | $0.32 \pm 0.01$ | $0.45 \pm 0.01$ | $0.58 \pm 0.01$ | $0.66 \pm 0.01$ | $0.72 \pm 0.01$ | $0.75 \pm 0.01$ |

Table 6: BERTScore $F_1$ (mean $\pm$ standard deviation) for each model across the different values of $k$

| | 0 | 1 | 2 | 3 | 6 | 10 |
|---|---|---|---|---|---|---|
| Qwen/Qwen2.5-32B-Instruct | $0.34 \pm 0.01$ | 0.51 | $0.59 \pm 0.01$ | $0.64 \pm 0.01$ | $0.72 \pm 0.01$ | $0.75 \pm 0.01$ |
| Qwen/Qwen2.5-7B-Instruct | $0.24 \pm 0.01$ | $0.48 \pm 0.01$ | $0.57 \pm 0.01$ | $0.63 \pm 0.01$ | $0.72 \pm 0.01$ | $0.74 \pm 0.01$ |
| google/flan-t5-xxl | $0.39 \pm 0.01$ | $0.45 \pm 0.01$ | $0.47 \pm 0.01$ | $0.48 \pm 0.01$ | $0.50 \pm 0.01$ | $0.52 \pm 0.01$ |
| google/gemma-2-9b-it | $0.32 \pm 0.01$ | $0.39 \pm 0.01$ | $0.55 \pm 0.01$ | $0.71 \pm 0.01$ | $0.75 \pm 0.01$ | $0.80 \pm 0.01$ |
| meta-llama/Llama-3.1-8B-Instruct | 0.33 | $0.47 \pm 0.01$ | $0.55 \pm 0.01$ | $0.63 \pm 0.01$ | $0.75 \pm 0.01$ | 0.77 |
| microsoft/phi-4 | 0.23 | $0.56 \pm 0.01$ | $0.69 \pm 0.01$ | $0.75 \pm 0.01$ | $0.79 \pm 0.01$ | $0.80 \pm 0.01$ |
| mistralai/Mistral-7B-Instruct-v0.3 | 0.32 | $0.49 \pm 0.01$ | $0.59 \pm 0.01$ | $0.66 \pm 0.01$ | $0.71 \pm 0.01$ | $0.74 \pm 0.01$ |

Table 7: Exact match score (mean $\pm$ standard deviation) for each model across the different values of $k$

| | 0 | 1 | 2 | 3 | 6 | 10 |
|---|---|---|---|---|---|---|
| Qwen/Qwen2.5-32B-Instruct | 0 | 0 | $0.03 \pm 0.01$ | $0.08 \pm 0.01$ | $0.16 \pm 0.02$ | $0.19 \pm 0.02$ |
| Qwen/Qwen2.5-7B-Instruct | 0 | 0.01 | $0.04 \pm 0.01$ | $0.09 \pm 0.01$ | $0.17 \pm 0.02$ | $0.19 \pm 0.02$ |
| google/flan-t5-xxl | 0.01 | $0.03 \pm 0.01$ | $0.04 \pm 0.01$ | $0.05 \pm 0.01$ | $0.06 \pm 0.01$ | $0.07 \pm 0.01$ |
| google/gemma-2-9b-it | 0 | 0.01 | $0.07 \pm 0.01$ | $0.16 \pm 0.02$ | $0.25 \pm 0.02$ | $0.29 \pm 0.02$ |
| meta-llama/Llama-3.1-8B-Instruct | 0 | 0.01 | $0.03 \pm 0.01$ | $0.09 \pm 0.01$ | $0.22 \pm 0.02$ | $0.25 \pm 0.02$ |
| microsoft/phi-4 | 0 | 0.01 | $0.10 \pm 0.01$ | $0.17 \pm 0.02$ | $0.24 \pm 0.02$ | $0.27 \pm 0.02$ |
| mistralai/Mistral-7B-Instruct-v0.3 | 0 | 0.01 | $0.04 \pm 0.01$ | $0.08 \pm 0.01$ | $0.14 \pm 0.02$ | $0.17 \pm 0.01$ |

Table 8: Examples from the manual evaluation.

| **Axioms** | **Generated question** | Sem. | LLM hall. | Comment |
|---|---|---|---|---|
| Phloem SubclassOf(PlantParts) | What is a phloem? | no | | grammar: not a |
| Sell SubclassOf(Exploit),Sell SubclassOf(condition some PayAction) | Which action must be performed in order to exploit a sale? | no | yes | almost |
| hasConjunct SubclassOf(Relation),hasConjunct SubclassOf(hasParent some CoordinatingConjunction) | Which relation has a parent that is some coordinating conjunction? | yes | | |
| Tasty-plant SubclassOf(eaten-by some Carnivore) | What is a tasty plant that is eaten by some carnivore? | no | | answer in question |
| Warthog SubclassOf(Participate-In some Walk),Warthog SubclassOf(Animal) | Which animal participates in walking? | yes | | |
| SemanticAsset SubclassOf(isPartOf only CulturalInstituteOrSite) | What is a semantic asset part of? | yes | | |
| NutTopping SubclassOf(Food) | What is a food that is a nut topping? | no | | answer in the question |
| Parsnip SubclassOf(proper-part-of some Plant),Parsnip SubclassOf(PlantParts) | Which plant part is a proper part of a plant and is a parsnip? | no | | answer in the question |
| NonspecificPronoun SubclassOf(hasSpecificity some Nonspecific) | Does a nonspecific pronoun have some level of nonspecificity? | no | yes | level is hallucinated |