Natural Language Querying for Humanities Knowledge Graphs: A Case Study on the GOLEM Knowledge Graph

Jose Maldonado-RodrÃŋguez[1,†], Arianna Graciotti[2,†], Valentina Presutti[2,†] and Federico Pianzola[1,†]

[1]University of Groningen, Oude Kijk in Het Jatstraat 26, 9712 EK Groningen, Netherlands
[2]LILEC, University of Bologna, Via Cartoleria 5, 40124 Bologna, Italy

Abstract

Large-scale Knowledge Graphs (KGs) are increasingly relevant for humanities research, yet querying them via SPARQL poses challenges for non-technical users. While Text-to-SPARQL studies predominantly target popular KGs such as Wikidata or DBpedia, domain-specific KGs remain underexplored. This paper introduces a bilingual (English-Spanish) dataset designed for evaluating automatic text-to-SPARQL translation on GOLEM, a humanities KG containing metadata and extracted features from fanfiction stories hosted on Archive of Our Own (AO3). The dataset includes 477 manually crafted natural language questions paired with gold SPARQL queries, augmented to 1,895 questions through automatic paraphrasing. We benchmark several Large Language Models (LLMs) with prompt-based approaches, particularly examining in-context learning methods that select prompt examples based on semantic similarity, which yield the best results. Error analysis highlights entity linking as essential for improving query generation. This work provides practical insights and opens pathways for future research on natural language interfaces for querying domain-specific KGs in Digital Humanities. The dataset and output of our experiments are available at: https://github.com/GOLEM-lab/GOLEM_Text-to-SPARQL.

Keywords

Humanities Knowledge Graphs, Large Language Models, Text-to-SPARQL, In-context learning

1. Introduction

Large-scale Knowledge Graphs (KGs) [1] offer structured resources for humanities research. Querying these KGs typically requires proficiency in SPARQL, posing a barrier to non-technical users. Recent studies in natural language (NL) interfaces for KG Question Answering (KGQA) mainly target popular KGs (e.g., Wikidata, DBpedia), neglecting custom domain-specific KGs.

This paper introduces a novel bilingual dataset for the text-to-SPARQL task, comprising English and Spanish NL questions answerable by querying GOLEM [2], a humanities KG containing metadata and features extracted from multilingual fanfiction stories. Leveraging this dataset, we evaluate the effectiveness of Large Language Models (LLMs) in translating NL questions into SPARQL queries through prompt-based methods.

Contributions of this work include:

- A novel bilingual (English and Spanish) dataset containing 477 NL questions, each paired with manually curated SPARQL queries, that can be used to interrogate the GOLEM KG SPARQL endpoint. The dataset is augmented into 1,895 NL questions by automatically generated paraphrases.
- An evaluation of LLM performance in automatically generating SPARQL queries from NL questions, with a focus on In-Context Learning [3] techniques.

SemDH 2025: Second International Workshop of Semantic Digital Humanities. Co-located with ESWC 2025, June 02, 2025, Portoroz, Slovenia.

*Corresponding author.
†These authors contributed equally.
j.e.maldonado.rodriguez@student.rug.nl (J. Maldonado-RodrÃŋguez); arianna.graciotti@unibo.it (A. Graciotti); valentina.presutti@unibo.it (V. Presutti); f.pianzola@rug.nl (F. Pianzola)
0009-0004-7918-809X (A. Graciotti); 0000-0002-9380-5160 (V. Presutti); 0000-0001-6634-121X (F. Pianzola)

We find that few-shot learning methods improve performance by selecting prompt examples based on semantic similarity to the input question. We provide a granular error analysis to identify current method limitations, laying the basis for future improvements.

The paper is structured as follows. In Section 2, we review existing text-to-SPARQL datasets, summarise recent approaches leveraging LLM-based prompting methods for NL-to-SPARQL translation, and position our work relative to these studies. Section 3 introduces GOLEM KG, describes the methodology for dataset construction, and provides a qualitative and quantitative characterisation of the resulting bilingual dataset. We then detail our experimental settings, including prompting strategies, LLMs tested, computational infrastructure, and evaluation metrics. In Section 4, we present experimental results along with an in-depth error analysis. Finally, we discuss conclusions and directions for future work. Our dataset and the output of our experiments are available at: https://github.com/GOLEM-lab/GOLEM_Text-to-SPARQL.

2. Related Work

The transformation of NL questions into SPARQL queries (Text-to-SPARQL) is widely studied in Knowledge Graph Question Answering (KGQA). However, existing Text-to-SPARQL methods do not generalise well, as most research efforts rely on training or fine-tuning models [4, 5] primarily on mainstream KGs (e.g., Wikidata, DBpedia), restricting their transferability to domain-specific KGs. Existing benchmarks focus on widely-used general KGs, with limited coverage of custom KGs: QALD-9 plus [6], QALD-10 [7], KQA Pro [8], LC-QuAD 2.0 [9] are based on Wikidata; LC-QuAD [10], QALD-9 on DBpedia; WebQuestionsSP [11], GraphQuestions [12], GrailQA [13], CFQ [14] on Freebase; DBLP-QUAD [15] on scholarly data.

With the emergence of LLMs, Text-to-SPARQL became intensively explored as an NL interface for KGs [16]. Prompt-based techniques fostered the development of Text-to-SPARQL resources and experiments beyond mainstream KGs toward domain-specific use cases in healthcare, scholarly data, and cultural heritage. Sivasubramaniam et al. [17] experimented in the medical sector with electronic health records (EHRs) and showed that SPARQL is underrepresented in LLM pre-training data, highlighting its complexity and demonstrating SPARQL as the most complex language for LLM-based query generation across different settings (zero-shot, few-shot, presence or absence of KG schema information in prompts). Sequeda et al. [18] focused on enterprise SQL schemas in the insurance sector. They identified an advantage in deriving KG representations from relational databases. Their zero-shot approach involves prompting LLMs with the OWL ontology describing the KG schema alongside instructions to generate corresponding SPARQL queries for given NL inputs. They report improved LLM performance as compared to direct database querying. The lack of KGQA benchmark datasets applicable beyond mainstream KGs is also addressed in [19]. The authors introduce Spider4SPARQL, a benchmark dataset comprising 10,181 manually curated NL questions paired with 5,693 distinct SPARQL queries at varying levels of complexity. Mountantonakis et al. [20] propose an LLM-based method to translate NL questions into SPARQL queries targeting cultural heritage KGs aligned with the ISO standard CIDOC-CRM ontology. They construct a benchmark comprising 100 NL questions paired with corresponding SPARQL queries applied to two real-world KGs representing artworks in the cultural heritage domain.

Having discussed available datasets, we now turn to recent prompt-based Text-to-SPARQL approaches leveraging LLMs. Zahera et al. [21] employed Chain-of-Thought prompting with in-context learning and semantically similar examples, enhancing query precision and syntactic flexibility. Dabramo et al. [22] proposed Dynamic Few-Shot Learning, combining semantic similarity with in-context learning, achieving robust results across KGQA benchmarks. Meyer et al. [23] presented LLM-KG-Bench, a framework evaluating the baseline SPARQL SELECT capabilities of LLMs on standard KGQA datasets. Avila et al. [24, 25] introduced Auto-KGQA and Auto-KGQAGPT, frameworks autonomously selecting smaller KG fragments to reduce token usage in prompts without performance loss, validated through experiments utilizing GTP3.5 Turbo, GTP-4, and GPT-4 Turbo.

In this work, we test LLMs' capability on the Text-to-SPARQL task. In particular, we evaluate

the flexibility of in-context learning with semantic search, avoiding costly fine-tuning and extensive collection of NL-SPARQL examples. With this objective, we adapt Dynamic Few-Shot Learning [22] to automatically generate SPARQL queries from NL questions that can be answered by interrogating a custom KG in the Digital Humanities realm, the GOLEM KG [2]. We partially address the limitations of [22], for example, their focus on English-only input, by testing on a bilingual dataset (English and Spanish) and experimenting with smaller LLMs. Our approach aims to reduce technical barriers in KGQA, facilitate NL interfaces, and broaden KGs' accessibility for humanities researchers beyond their knowledge of the SPARQL querying language.

3. Methodology

The dataset construction has been curated manually by a Language and Communication Technologies master student (author of this paper) as part of a curricular research internship. The annotator is proficient in English and is a native Spanish speaker. This section describes our approach, introducing GOLEM KG [2], which is the core of our case study. We cover how we constructed the dataset, the prompting strategies adopted, and our evaluation framework.

3.1. The GOLEM Knowledge Graph

The GOLEM KG contains metadata and extracted features of fanfiction stories in various languages from the popular online platform Archive of Our Own (AO3) [26]. Metadata about works are provided by fanfiction authors and include common metadata such as author, title, publication date, as well as a wide array of additional information, such as content tags, characters appearing in the story, and their relationships. In addition, information on narrative and stylistic elements, as well as reader response data (such as syntactic complexity and lexical richness), has been added. The formal ontology used to model the data combines existing ontologies with new classes and properties specific to the domain of narrative and fiction [2]. For example, the *lrm:F1_Work* class is defined by LRMoo, an extension of the ISO standard for cultural heritage CIDOC-CRM [1]; there are DCMI Metadata Terms, e.g. *dct:title* or *dct:creator* [2]; and new classes like *gc:G1_Character* (a *crm:E89_Propositional_Object*). A description of other predicates used in the KG can be found in [27]. GOLEM KG is available via a search interface[3] and a public SPARQL endpoint[4].

3.2. Dataset construction

Table 1
Distribution of questions by language and type in our proposed dataset. Note that the set of paraphrased questions includes all the instantiated questions.

Language	#template questions	#instantiated questions	#paraphrased questions
EN	39	236	1172
ES	39	241	723
All	78	477	1,895

Since the data in the GOLEM KG is multilingual, and potential end-users may want to query the KG in their native language, we aim to develop a multilingual dataset. We constructed NL questions in English and Spanish and paired them with their corresponding SPARQL query. We formulated the questions based on typical queries that end users interested in fanfiction might ask of the KG. Our methodology can be extended to additional languages. Dataset statistics are reported in Table 1. The following paragraphs describe how the dataset was constructed based on the included question types.

[1] https://cidoc-crm.org//extensions/lrmoo/html/LRMoo_v1.0.html
[2] https://www.dublincore.org/specifications/dublin-core/dcmi-terms/
[3] http://search.golemlab.eu:3006/en/
[4] http://graph.golemlab.eu:8890/sparql

3.2.1. Template questions

The first steps towards creating a multilingual dataset of NL question-SPARQL query pairs in English and Spanish involved producing 39 generic *template questions* per language. Spanish questions are translations of English questions. These templates use placeholders instead of specific KG entities. There are several types of placeholders, each referring to one GOLEM predicate:

- *[[fandom]]*, corresponding to `https://golemlab.eu/graph/fandom`;
- *[[story]]*, corresponding to `https://golemlab.eu/graph/title`;
- *[[character]]*, corresponding to `https://golemlab.eu/graph/character`;
- *[[keyword]]*, corresponding to `https://golemlab.eu/graph/keyword`.

Below is one example of a template question per language:

- *How many [[fandom]] stories are there?*
- *¿CuÃąntas historias de [[fandom]] hay publicadas?*

Among the template questions, some do not contain placeholders because they do not require the inclusion of any KG entities. Below is one example per language:

- *How many stories are tagged as explicit?*
- *¿CuÃąntas historias estÃąn marcadas como explÃŋcitas?*

Before execution on the KG's SPARQL endpoint, these generic template questions require an instantiation step, described in the following paragraph.

3.2.2. Instantiated questions

We instantiate the template questions by replacing placeholders with actual entities from the KG when required. These entities are obtained through a series of simple queries on the KG, performing one query per placeholder type. Below is an example of a query built for this purpose:

```
prefix golem: <https://golemlab.eu/graph/>
SELECT DISTINCT ?fandom
WHERE {
    ?story golem:fandom ?fandom .
}
```

The resulting lists of instances are then used at random to replace the placeholders in the dataset, both in the NL question and its corresponding SPARQL query.

We refer to the resulting questions as *instantiated questions*, which can directly query the KG via the SPARQL endpoint. We created 10 instantiated questions for each template question. Each instantiated question has been paired with a manually crafted SPARQL query. If the corresponding SPARQL query generated an empty answer, the sample was removed. Below, we report two examples:

- *How many 1984 - George Orwell stories are there?*

```
prefix golem: <https://golemlab.eu/graph/>
prefix dc: <http://purl.org/dc/terms/>
prefix gc: <https://ontology.golemlab.eu/>
SELECT (COUNT(DISTINCT ?story) as ?uploads)
WHERE {
    ?story golem:fandom "1984 - George Orwell" .
}
```

- *¿CuÃąntas historias de Valley of Tears (TV) hay publicadas?*

```
prefix golem: <https://golemlab.eu/graph/>
prefix dc: <http://purl.org/dc/terms/>
prefix gc: <https://ontology.golemlab.eu/>

SELECT (COUNT(DISTINCT ?story) as ?uploads)
WHERE {
    ?story golem:fandom "Valley of Tears (TV)" .
}
```

The template questions that do not contain any placeholders do not need to undergo any instantiation process and are directly associated with the corresponding SPARQL queries, as in the example below:

- *How many stories are tagged as explicit?*
- *¿CuÃąntas historias estÃąn marcadas como explÃŋcitas?*

```
prefix golem: <https://golemlab.eu/graph/>
prefix dc: <http://purl.org/dc/terms/>
prefix gc: <https://ontology.golemlab.eu/>

SELECT COUNT(?story) as ?explicit_stories
WHERE {
    ?story golem:rating "Explicit" .
}
```

The two questions in English and Spanish reported in the above example are semantically equivalent (the Spanish one is the translation of the English one) and are therefore paired with the same SPARQL query.

The SPARQL queries corresponding to the instantiated questions are executed against the GOLEM KG SPARQL endpoint, and their responses are collected and evaluated.

3.2.3. Paraphrased questions

Instantiated questions are paraphrased via data augmentation using `deepseek-r1-7b`. However, Spanish questions are frequently misparaphrased into English. We address this by verifying the correctness of the paraphrased questions' language using a cross-check that implements the language detection library `lingua`[5]. Paraphrases detected by `lingua` in languages differing from the manually annotated source are discarded.[6] Below, we list some examples of paraphrased questions:

- *What is the number of George Orwell 1984 works available?*
- *How many George Orwell's 1984 pieces exist?*
- *¿CuÃąntas narrativas de *Valley of Tears* (TV) existen ?*

The prompt used in the paraphrasing step can be seen in A.1.1.

3.3. Experimental Setting

In this section, we first introduce the prompting approaches tested (Zero-shot, Naive Few-shot, and an adaptation of Dynamic Few-shot Learning [22]). Then, we introduce the models used. Finally, we describe the evaluation metrics chosen.

[5] https://github.com/pemistahl/lingua-py

[6] We discarded 457 paraphrases: 25 expected in English but detected otherwise; 450 expected in Spanish but detected otherwise.

3.3.1. Prompting approaches

Zero-shot In the Zero-shot (ZS) prompting approach, we include in the prompt only the task instructions and the list of predicates from the KG. We report an example of ZS prompt per language in Appendix A.1.2.

Naive few-shots In the Naive Few-shot (NFS) prompting approach, we include in the prompt the task instructions and the list of predicates from the KG. Plus, we include three random examples in the prompt, taken from the same language as the question being processed. We report an example of an NFS prompt per language in Appendix A.1.3.

Dynamic Few-shot Learning (adapted) In the adapted Dynamic Few-shot Learning (a-DFSL) prompting approach, we include in the prompt the task instructions and the list of predicates from the KG. Additionally, adapting what is described in [22], we also include examples from our dataset. To select the examples, we encode all dataset questions using `paraphrase-multilingual-mpnet-base-v2` as a sentence encoder model. The input question is then encoded and compared against the dataset using cosine similarity. The 3 most similar questions are selected as examples. To simulate real-world complexity, questions sharing the same SPARQL query as the input are excluded. Example selection is constrained to samples in the same language as the input question. Contrary to [22], we do not provide in the prompt the gold relation and the gold entity to be used in the target query. We report prompt examples for an English and a Spanish question in Appendix A.1.4.

3.3.2. Models

paraphrase-multilingual-mpnet-base-v2 This sentence embedding model[7] is used in the scope of this work to perform the semantic search required to select the most similar examples to the input sentence for inclusion in the a-DFSL prompt. We chose this model because of its multilingual specialisation.

deepseek-coder-v2 This open-weight Mixture-of-Experts (MoE) LLM[8] specializes in coding and mathematical tasks. We pull it from its Ollama repository in its 16B-parameter version. In our work, it is used to transform the NL questions into SPARQL queries via different prompting approaches.

deepseek-r1:7b and :70b We use these reasoning open-weight LLMs to transform NL questions into SPARQL queries via different prompting approaches. We pull them from the related Ollama repository[9] in their 7B and 70B parameters versions.

llama3.1:70b We use this open-weight LLM to transform NL questions into SPARQL queries using different prompting approaches. We pull it from its Ollama repository[10].

3.3.3. System

Experiments were conducted on a server equipped with an Intel i9-11900KF CPU, 128GB RAM, and the GPUs NVIDIA GeForce RTX 3090 (24GB VRAM).

[7]Available at https://huggingface.co/sentence-transformers/paraphrase-multilingual-mpnet-base-v2
[8]Available at https://ollama.com/library/deepseek-coder-v2
[9]Available at https://ollama.com/library/deepseek-r1
[10]Available at https://ollama.com/library/llama3.1:70b

3.3.4. Metrics

We evaluate performance by comparing the results obtained by executing the gold SPARQL queries in our dataset against the GOLEM KG with those obtained by executing the automatically generated SPARQL queries on the same KG on a per-sample basis. For each sample, an exact string match between the two responses yields a true positive (TP). If the generated response is non-empty yet differs from the ground truth, a false positive (FP) and a false negative (FN) are recorded; if the generated response is empty, only FN is incremented.

4. Evaluation

In this section, we present the results of our experiments, and we perform a detailed analysis of the errors.

Table 2
Performance of the models tested across the different prompting strategies on the instantiated questions dataset. The best Precision, Recall, and F1 Score are in bold, and the second best are underlined.

Model	Prompting Approach	Language	Precision	Recall	F1 Score	Processing time (hh:mm:ss)
deepseek-coder-v2	Zero-shot	All	0.0331	0.0168	0.0223	0:44:43
		EN	0.0196	0.0085	0.0119	
		ES	0.0429	0.0249	0.0315	
	Naive few-shots	All	0.3727	0.2977	0.3310	0:15:54
		EN	0.4190	0.3178	0.3614	
		ES	0.3317	0.2780	0.3025	
	a-DFSL	All	<u>0.8790</u>	<u>0.8532</u>	<u>0.8660</u>	0:15:51
		EN	<u>0.8874</u>	<u>0.8686</u>	<u>0.8779</u>	
		ES	<u>0.8707</u>	<u>0.8382</u>	<u>0.8541</u>	
deepseek-r1:7b	a-DFSL	All	0.5640	0.4990	0.5295	1:06:05
		EN	0.6119	0.5678	0.5890	
		ES	0.5123	0.4315	0.4685	
llama3.1:70b	a-DFSL	All	**0.8998**	**0.8658**	**0.8825**	9:50:10
		EN	**0.9000**	**0.8771**	**0.8884**	
		ES	**0.8996**	**0.8548**	**0.8766**	

Table 3
Performance on the paraphrased questions dataset.

Model	Prompting Approach	Language	Precision	Recall	F1 Score	Processing time (hh:mm:ss)
deepseek-coder-v2	a-DFSL	All	0.5349	0.4364	0.4807	2:27:00
		EN	0.5308	0.4334	0.4772	
		ES	0.5416	0.4412	0.4863	

4.1. Results

Table 2 summarises experiment results on the instantiated question dataset for all the tested models and prompting strategies, including inference times. We evaluated `deepseek-coder-v2` under three prompt conditions: ZS, NFS, and a-DFSL. a-DFSL significantly outperforms ZS and NFS. a-DFSL gains validate our and [22] hypothesis: similarity-based example selection improves SPARQL generation in few-shot settings. We tested `deepseek-r1:7b` with an a-DFSL prompting approach, and we observed significantly lower performance than `deepseek-coder-v2` and higher processing time. `Llama3.1:70b` outperforms `deepseek-coder-v2` due to its larger parameter size but incurs significantly higher processing time as compared to the best run of `deepseek-coder-v2` (with the a-DFSL prompting approach).

Precision consistently exceeds recall across all models and prompting approaches, indicating more frequent query failures or empty results rather than incorrect answers. For all the models tested across all prompting approaches (except the zero-shot approach, which is less relevant due to its extremely low performance), performance on the English-language subset is better than on the Spanish-language subset.

Thus, we select `deepseek-coder-v2` (a-DFSL) for further experiments on the augmented dataset (Table 3). Performance notably declines on augmented data due to entity alterations introduced by paraphrasing. Entity mismatches between paraphrases and KG canonical forms cause the SPARQL query to obtain erroneous results. Such results closely align with the ablation studies performed in the original DFSL study [22], which reported lower accuracy on QALD-9 DB (49.59% accuracy) when the prompt did not include the gold entities and relations. Future work should include entity linking to explicitly incorporate entities' canonical forms in prompts.

4.2. Error Analysis

Table 4
Error type breakdown for each mismatch found in deepseek-coder-v2 (a-DFSL) run on the instantiated questions dataset.

Error Type	Count	%
Wrong predicate	20	28,5%
Failed KG Entity Recognition	15	21.5%
Incomplete query	12	17%
Evaluation method errors	12	17%
Failed execution	11	16%
Total	70	100%

We conducted a systematic error analysis on all 70 errors produced by our best-performing model (deepseek-coder-v2), which optimises the accuracy-runtime tradeoff. Notably, our analysis could not spot the "triple-flip" errorâĂŤa well-documented phenomenon in text-to-SPARQL conversion where subject and object positions are reversed in generated triples, resulting in empty or incorrect query results. The identified errors were instead categorised into two primary classes: (i) *Failed execution*, covering queries unable to execute successfully against the SPARQL endpoint (e.g., `QueryBadFormed`, timeouts), and (ii) *Incorrect results*, referring to queries executed successfully but returning answers differing from the gold annotations. The second macro-category was further divided into specific error types (see Table 4), namely *Failed KG Entity Recognition*, where the model incorrectly recognized entities from the NL question; *Wrong Predicate*, where the model selected an incorrect predicate among those provided in the prompt; *Incomplete Query*, where generated queries lacked sufficient complexity to retrieve all expected results; and *SPARQL Syntax Error*, involving syntactic mistakes in the generated query that led to incorrect results. Additionally, we identified errors attributable to the evaluation approach itself (*Evaluation Method Errors*), where queries returned correct results but were mistakenly flagged as wrong due to minor variations in variable naming or result grouping. Future work will address these evaluation inaccuracies.

Table 5 illustrates each error type among queries executed successfully but returning incorrect results. In the first example, comparing the gold SPARQL query with the generated one reveals that both queries are structurally identical, differing only in the KG entity used as the object of predicate `golem:title`. The generated query incorrectly uses the entity `"Wolfstar prompts"` instead of the correct entity `"(fanart) Wolfstar prompts"`, causing an empty (incorrect) result.

Another common issue is represented by queries of the error type *Incomplete Query*, exemplified by subjective NL questions such as "What is *Forbidden Like The Forest* about?", reported as the second example in the table. The annotator associated this question with a highly articulated SPARQL query, which retrieves multiple detailed elements (e.g., keywords, romantic categories, content warnings, collections, series, and summaries). In contrast, the model produced a minimal query, interpreting the

Table 5
Examples of Errors in Generated SPARQL Queries.

Question	Gold SPARQL Query	Generated SPARQL Query	Error Type
What series is (fanart) Wolfstar prompts a part of?	`SELECT DISTINCT ?series WHERE ?story golem:title "(fanart) Wolfstar prompts" . ?story golem:series ?series .`	`SELECT DISTINCT ?series WHERE ?story golem:title "Wolfstar prompts" . ?story golem:series ?series .`	Failed KG Entity Recognition
What is Forbidden Like The Forest about?	`SELECT DISTINCT ?series ?collection ?content_warning ?romantic_category (GROUP_CONCAT(DISTINCT ?keyword; separator=", ") AS ?keywords) ?summary WHERE ?story golem:title "Forbidden Like The Forest" . ?story golem:keyword ?keyword . ?story golem:romanticCategory ?romantic_category . ?story golem:series ?series . ?story golem:collections ?collection . ?story golem:contentWarning ?content_warning . ?story golem:summary ?summary .`	`SELECT DISTINCT ?about WHERE ?story golem:title "Forbidden Like The Forest" . ?story golem:summary ?about .`	Incomplete Query
¿CuÃąntos autores han publicado alguna historia?	`SELECT (COUNT(DISTINCT ?author) as ?uploads) WHERE ?story golem:author ?author .`	`SELECT (COUNT(DISTINCT ?author) as ?authorsCount) WHERE ?story dc:creator ?author .`	Wrong predicate

ambiguous phrase "about" simply as retrieving the story's summary. This discrepancy highlights a modelling issue where subjective interpretations by annotators may lead to overly complex gold standard queries compared to the minimalistic outputs generated by the model. In future work, annotation guidelines will be refined to address this kind of subjectivity.

A final error type illustrated in the table is *Wrong Predicate*, exemplified by the fourth and last query pair. The gold SPARQL query correctly uses the predicate `golem:author`, whereas the model-generated query mistakenly selects the predicate `dc:creator`. In such cases, the model erroneously selects a predicate from among those provided in the prompt or hallucinates predicates that do not exist in the KG, leading to incorrect or incomplete results.

To analyse the errors of the run on the paraphrased question dataset, reported in Table 3, we selected a sample of 30 erroneous cases. In this sample, we do not consider those errors due to the failed execution of the query. Questions were chosen to maximise semantic diversity. Semantic embeddings of questions were computed using the same model used for selecting a-DFSL examples. A greedy selection method then iteratively picked the most semantically distinct questions, measured by cosine distance, ensuring coverage of the broadest possible range of error cases. This approach allowed us to efficiently maximise

Table 6
Error type breakdown for a sample of 30 errors found in deepseek-coder-v2 (a-DFSL) run on the paraphrased questions dataset.

Error Type	Num.	%
Misleading paraphrase	10	33%
Failed KG Entity Recognition	9	30%
Wrong predicate	7	23%
Evaluation method errors	4	13%
Total	30	100%

the variety of errors considered while minimising redundancy in our analysis.

As we report in Table 4.2, unlike the analysis of instantiated questions, no instances of *Incomplete Query* errors emerged, possibly due to the smaller sample size. However, we identified a new error category, *Misleading Paraphrase*, grouping queries arising from flawed or unclear paraphrased questions. Typical examples include nonsensical, mixed-language, or malformed paraphrases such as *"It never entered my mind's key points or topics"*, *"¿CÃşmo many kudos se le otorgan a If only?"*, or *"¿QuÃl pÃąjaros aparecen en If You Could Be Anywhere?"*. Future analyses will focus on addressing this error type through improved paraphrase quality control.

5. Limitations and Future Work

One limitation of our dataset construction is that the questions were not directly informed by domain experts or end-users from computational literature or fanfiction communities. Instead, they were primarily inspired by information already present in the KG, which could potentially introduce bias and limit method effectiveness. Future work should involve gathering explicit requirements from target end-users to create more representative queries. Additionally, the dataset augmentation process produced paraphrases with varying linguistic quality, including language inconsistencies such as mixed-language outputs (e.g., English instead of expected Spanish). To address this, future efforts should integrate automatic quality control mechanisms, such as leveraging LLMs as evaluators, or conduct manual assessments to enhance dataset quality. Some collected NL questions exhibit subjective interpretations, for example, queries about the content of stories, which lead to subjective gold SPARQL annotations. Future work will refine the annotation guidelines to minimise ambiguity.

Another limitation concerns the systemâĂŹs generalizability beyond the semi-templated scope of the constructed dataset. While the tested methodology demonstrates effectiveness within the instantiated questions, its performance degrades on paraphrased variants, indicating limited robustness to linguistic variation. Furthermore, its behaviour in more open-ended or less structured query scenarios remains underexplored. Future work will involve evaluating system performance on more naturalistic and user-authored queries.

We acknowledge the explicit predicate listing as a simplification. In deployed applications, these would be stored as pre-defined variables hidden from end-users. Future work should examine how prompt complexity affects performance and explore more compact schema representations.

The best-performing prompting approach (a-DFSL) is a simplified version of the original DFSL [22] implementation. Future improvements include a more mature, sophisticated prompting strategy, beginning with automatic entity linking. Entity linking by either LLM or specialized entity linkers can identify and inject KG entities in their canonicalised form directly into the prompts, reducing recognition errors. More advanced examples of retrieval methods will also be explored.

The current evaluation strategy compares answers from generated and gold SPARQL queries, but has limitations. Queries returning identical results with different groupings or slight variable differences are incorrectly marked as errors. Future work will refine the evaluation method for greater flexibility and accuracy.

6. Conclusion

In this work, we introduced a bilingual dataset of NL in English and Spanish questions paired with corresponding SPARQL queries. The dataset targets the GOLEM KG, containing metadata and features extracted from fanfiction hosted on the Archive of Our Own (AO3) platform. The dataset comprises instantiated questions, manually crafted and automatically populated with KG entities, and an augmented version generated via automatic paraphrasing. We used the dataset to benchmark various LLMs on the text-to-SPARQL task, exploring several prompting strategies. An adapted, simplified version of DFSL [22], which selects prompt examples via semantic similarity to input questions, demonstrated superior performance. Error analysis revealed that integrating entity linking is critical to improving query generation quality. This case study in Digital Humanities provides practical insights and suggests pathways for future research on NL interfaces for querying knowledge graphs through text-to-SPARQL methods.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

Acknowledgements

This work is part of the Graphs and Ontologies for Literary Evolution Models (GOLEM) project funded by the European Commission. Jose Maldonado-RodrÃŋguez is supported by the Erasmus Mundus Masters Program in Language and Communication Technologies (LCT), EU grant no. 2019-1508. Arianna Graciotti is supported by the European UnionâĂŹs Horizon 2020 research and innovation programme under grant agreement No 101004746.

References

[1] A. Hogan, E. Blomqvist, M. Cochez, C. DâĂŹamato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab, A. Zimmermann, Knowledge graphs, ACM Comput. Surv. 54 (2021). URL: https://doi.org/10.1145/3447772. doi:10.1145/3447772.

[2] F. P. Franziska Pannach, Luotong Cheng, The golem knowledge graph: Exploring fanfiction narratives through structured data, in: W. Haverals, M. Koolen, L. Thompson (Eds.), Proceedings of the Computational Humanities Research Conference 2024, volume 3834 of *CEUR Workshop Proceedings*, CEUR Workshop Proceedings (CEUR-WS.org), Aarhus, Denmark, 2024, pp. 462–471. URL: https://ceur-ws.org/Vol-3834/paper80.pdf.

[3] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang, X. Sun, L. Li, Z. Sui, A survey on in-context learning, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 1107–1128. URL: https://aclanthology.org/2024.emnlp-main.64/. doi:10.18653/v1/2024.emnlp-main.64.

[4] D. Banerjee, P. A. Nair, J. N. Kaur, R. Usbeck, C. Biemann, Modern baselines for sparql semantic parsing, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '22, Association for Computing Machinery, New York, NY, USA, 2022, p. 2260âĂŞ2265. URL: https://doi.org/10.1145/3477495.3531841. doi:10.1145/3477495.3531841.

[5] J. Qi, C. Su, Z. Guo, L. Wu, Z. Shen, L. Fu, X. Wang, C. Zhou, Enhancing sparql query generation for knowledge base question answering systems by learning to correct triplets, Applied Sciences 14 (2024). URL: https://www.mdpi.com/2076-3417/14/4/1521. doi:10.3390/app14041521.

[6] A. Perevalov, D. Diefenbach, R. Usbeck, A. Both, Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers, in: 2022 IEEE 16th International Conference on Semantic Computing (ICSC), 2022, pp. 229–234. doi:10.1109/ICSC52841.2022.00045.

[7] L.-A. Kaffee, S. Razniewski, P. Vougiouklis, R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. MÃűller, J. Huang, J. Reineke, A.-C. N. Ngomo, M. Saleem, A. Both, Qald-10 âĂŞ the 10th challenge on question answering over linked data: Shifting from dbpedia to wikidata as a kg for kgqa, Semantic Web 15 (2024) 2193–2207. URL: https://doi.org/10.3233/SW-233471. doi:10.3233/SW-233471. arXiv:https://doi.org/10.3233/SW-233471.

[8] S. Cao, J. Shi, L. Pan, L. Nie, Y. Xiang, L. Hou, J. Li, B. He, H. Zhang, KQA pro: A dataset with explicit compositional programs for complex question answering over knowledge base, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 6101–6119. URL: https://aclanthology.org/2022.acl-long.422/. doi:10.18653/v1/2022.acl-long.422.

[9] M. Dubey, D. Banerjee, A. Abdelkawi, J. Lehmann, Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia, in: C. Ghidini, O. Hartig, M. Maleshkova, V. Svátek, I. Cruz, A. Hogan, J. Song, M. Lefrançois, F. Gandon (Eds.), The Semantic Web – ISWC 2019, Springer International Publishing, Cham, 2019, pp. 69–78.

[10] P. Trivedi, G. Maheshwari, M. Dubey, J. Lehmann, Lc-quad: A corpus for complex question answering over knowledge graphs, in: C. d'Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange, J. Heflin (Eds.), The Semantic Web – ISWC 2017, Springer International Publishing, Cham, 2017, pp. 210–218.

[11] J. Berant, A. Chou, R. Frostig, P. Liang, Semantic parsing on Freebase from question-answer pairs, in: D. Yarowsky, T. Baldwin, A. Korhonen, K. Livescu, S. Bethard (Eds.), Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 1533–1544. URL: https://aclanthology.org/D13-1160/.

[12] Y. Su, H. Sun, B. Sadler, M. Srivatsa, I. Gür, Z. Yan, X. Yan, On generating characteristic-rich question sets for QA evaluation, in: J. Su, K. Duh, X. Carreras (Eds.), Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Austin, Texas, 2016, pp. 562–572. URL: https://aclanthology.org/D16-1054/. doi:10.18653/v1/D16-1054.

[13] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan, Y. Su, Beyond i.i.d.: Three levels of generalization for question answering on knowledge bases, in: Proceedings of the Web Conference 2021, WWW '21, Association for Computing Machinery, New York, NY, USA, 2021, p. 3477âĂŞ3488. URL: https://doi.org/10.1145/3442381.3449992. doi:10.1145/3442381.3449992.

[14] D. Keysers, N. SchÃďrli, N. Scales, H. Buisman, D. Furrer, S. Kashubin, N. Momchev, D. Sinopalnikov, L. Stafiniak, T. Tihon, D. Tsarkov, X. Wang, M. van Zee, O. Bousquet, Measuring compositional generalization: A comprehensive method on realistic data, 2020. URL: https://arxiv.org/abs/1912.09713. arXiv:1912.09713.

[15] D. Banerjee, S. Awale, R. Usbeck, C. Biemann, DBLP-QuAD: A question answering dataset over the DBLP scholarly knowledge graph, in: BIR 2023: 13th International Workshop on Bibliometric-enhanced Information Retrieval at ECIR 2023, 2023.

[16] H. Khorashadizadeh, F. Z. Amara, M. Ezzabady, F. Ieng, S. Tiwari, N. Mihindukulasooriya, J. Groppe, S. Sahri, F. Benamara, S. Groppe, Research trends for the interplay between large language models and knowledge graphs, 2024. URL: https://arxiv.org/abs/2406.08223. arXiv:2406.08223.

[17] S. Sivasubramaniam, C. E. Osei-Akoto, Y. Zhang, K. Stockinger, J. Fuerst, Sm3-text-to-query: Synthetic multi-model medical text-to-query benchmark, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, volume 37, Curran Associates, Inc., 2024, pp. 88627–88663. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/

a182a8e6ebc91728b6e6b6382c9f7b1e-Paper-Datasets_and_Benchmarks_Track.pdf.
[18] J. Sequeda, D. Allemang, B. Jacob, A benchmark to understand the role of knowledge graphs on large language model's accuracy for question answering on enterprise sql databases, 2023. URL: https://arxiv.org/abs/2311.07509. arXiv:2311.07509.
[19] C. Kosten, P. CudrÃĺ-Mauroux, K. Stockinger, Spider4sparql: A complex benchmark for evaluating knowledge graph question answering systems, in: 2023 IEEE International Conference on Big Data (BigData), 2023, pp. 5272–5281. doi:10.1109/BigData59044.2023.10386182.
[20] M. Mountantonakis, Y. Tzitzikas, Generating sparql queries over cidoc-crm using a two-stage ontology path patterns method in llm prompts, J. Comput. Cult. Herit. 18 (2025). URL: https://doi.org/10.1145/3708326. doi:10.1145/3708326.
[21] H. M. Zahera, M. Ali, M. A. Sherif, D. Moussallem, A.-C. N. Ngomo, Generating sparql from natural language using chain-of-thoughts prompting, in: SEMANTICS, 2024, pp. 353–368. URL: https://doi.org/10.3233/SSW240028.
[22] J. D'Abramo, A. Zugarini, P. Torroni, Dynamic few-shot learning for knowledge graph question answering, 2024. URL: https://arxiv.org/abs/2407.01409. arXiv:2407.01409.
[23] L.-P. Meyer, J. Frey, F. Brei, N. Arndt, Assessing SPARQL Capabilities of Large Language Models, in: NLP4KGC: 3rd International Workshop on Natural Language Processing for Knowledge Graph Creation in conjunction with SEMANTiCS 2024 Conference, 2024.
[24] C. V. S. Avila, M. A. Casanova, V. M. P. Vidal, A framework forÂăquestion answering onÂăknowledge graphs using large language models, in: A. Meroño Peñuela, O. Corcho, P. Groth, E. Simperl, V. Tamma, A. G. Nuzzolese, M. Poveda-Villalón, M. Sabou, V. Presutti, I. Celino, A. Revenko, J. Raad, B. Sartini, P. Lisena (Eds.), The Semantic Web: ESWC 2024 Satellite Events, Springer Nature Switzerland, Cham, 2025, pp. 168–172.
[25] C. V. S. Avila, V. M. Vidal, W. Franco, M. A. Casanova, Experiments with text-to-sparql based on chatgpt, in: 2024 IEEE 18th International Conference on Semantic Computing (ICSC), 2024, pp. 277–284. doi:10.1109/ICSC59802.2024.00050.
[26] C. Fiesler, S. Morrison, A. S. Bruckman, An archive of their own: A case study of feminist hci and values in design, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, CHI '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 2574âĂŞ2585. doi:10.1145/2858036.2858409.
[27] F. Pannach, X. Yang, N. V. Solissa, Z. Yu, A. V. Cranenburgh, M. V. D. Ree, F. Pianzola, The golem triple store: A graph-based representation of narrative and fiction, in: B. Sartini, J. Raad, P. Lisena, A. M. PeÃśuela, M. Beetz, I. Blin, P. Cimiano, J. de Berardinis, S. Gottschalk, F. Ilievski, N. Jain, J. Kim, M. KÃijmpel, E. Motta, I. Tiddi, J.-P. TÃűberg (Eds.), ESWC 2024 Workshops and Tutorials Joint Proceedings, volume 3749 of *CEUR Workshop Proceedings*, CEUR Workshop Proceedings (CEUR-WS.org), Hersonissos, Greece, 2024. URL: https://hdl.handle.net/11370/f5b70d22-cc55-4dac-a5a4-155c1e515b4c.

A. Appendix

A.1. Prompting approaches

In the sections below, we report the prompts used in the different experimental settings tested in this paper.

A.1.1. Data augmentation prompt examples

ENGLISH

```
Generate 10 paraphrases for the following text, making the responses short and query-like. Please do not say anything after </think>, only the paraphrases. If there are any words within double square brackets (such as [[story]] or [[character]] ), please do not modify them.:
```

```

How many 1984 - George Orwell stories are there?
```

Listing 1: English question ata augmentation prompt example

SPANISH

```
Generate 10 paraphrases for the following text, making the responses short and query-
    like. Please do not say anything after </think>, only the paraphrases. If there are
    any words within double square brackets (such as [[story]] or [[character]] ),
    please do not modify them.:

Â£CuÃąntas historias de Valley of Tears (TV) hay publicadas?
```

Listing 2: English question ata augmentation prompt example

A.1.2. ZS Prompt Examples

ENGLISH

```
Your task is to translate a question in natural language into a SPARQL query for the
    GOLEM knowledge graph.
The query must follow these guidelines:
1. SPARQL queries must include the following prefix:
   prefix golem: <https://golemlab.eu/graph/>
2. Enclose SPARQL queries within <SPARQL> </SPARQL> tags.
3. You must generate 1 query(ies).
4. It is very important that you use only the predicates provided below.

###
Predicates:
https://golemlab.eu/graph/numberOfComments
https://golemlab.eu/graph/numberOfKudos
https://golemlab.eu/graph/publicationStatus
https://golemlab.eu/graph/dateModified
https://golemlab.eu/graph/characters
https://golemlab.eu/graph/collections
https://golemlab.eu/graph/fandom
https://golemlab.eu/graph/publisher
https://golemlab.eu/graph/rating
https://golemlab.eu/graph/series
https://golemlab.eu/graph/story_id
https://golemlab.eu/graph/summary
https://golemlab.eu/graph/numberOfChapters
https://golemlab.eu/graph/datePublished
https://golemlab.eu/graph/keyword
https://golemlab.eu/graph/contentWarning
https://golemlab.eu/graph/numberOfWords
https://golemlab.eu/graph/socialRelationships
https://golemlab.eu/graph/datePackaged
https://golemlab.eu/graph/romanticCategory
https://golemlab.eu/graph/noOfPairings
https://golemlab.eu/graph/topPartner
https://golemlab.eu/graph/topPartnerPairings
https://golemlab.eu/graph/averageWordLength
https://golemlab.eu/graph/MSTTR
https://golemlab.eu/graph/bigramEntropy
https://golemlab.eu/graph/wordEntropy
https://golemlab.eu/graph/fleschGrade
https://golemlab.eu/graph/fleschEase
```

```
https://golemlab.eu/graph/SMOG
https://golemlab.eu/graph/ARI
https://golemlab.eu/graph/daleChallNew
https://golemlab.eu/graph/nominalRatio
https://golemlab.eu/graph/nominalRatioInverse


Question: How many 1984 - George Orwell stories are there?

Query:
```

Listing 3: English question ZS Prompt Example

SPANISH

```
Your task is to translate a question in natural language into a SPARQL query for the
    GOLEM knowledge graph.
The query must follow these guidelines:
1. SPARQL queries must include the following prefix:
   prefix golem: <https://golemlab.eu/graph/>
2. Enclose SPARQL queries within <SPARQL> </SPARQL> tags.
3. You must generate 1 query(ies).
4. It is very important that you use only the predicates provided below.

###
Predicates:
https://golemlab.eu/graph/numberOfComments
https://golemlab.eu/graph/numberOfKudos
https://golemlab.eu/graph/publicationStatus
https://golemlab.eu/graph/dateModified
https://golemlab.eu/graph/characters
https://golemlab.eu/graph/collections
https://golemlab.eu/graph/fandom
https://golemlab.eu/graph/publisher
https://golemlab.eu/graph/rating
https://golemlab.eu/graph/series
https://golemlab.eu/graph/story_id
https://golemlab.eu/graph/summary
https://golemlab.eu/graph/numberOfChapters
https://golemlab.eu/graph/datePublished
https://golemlab.eu/graph/keyword
https://golemlab.eu/graph/contentWarning
https://golemlab.eu/graph/numberOfWords
https://golemlab.eu/graph/socialRelationships
https://golemlab.eu/graph/datePackaged
https://golemlab.eu/graph/romanticCategory
https://golemlab.eu/graph/noOfPairings
https://golemlab.eu/graph/topPartner
https://golemlab.eu/graph/topPartnerPairings
https://golemlab.eu/graph/averageWordLength
https://golemlab.eu/graph/MSTTR
https://golemlab.eu/graph/bigramEntropy
https://golemlab.eu/graph/wordEntropy
https://golemlab.eu/graph/fleschGrade
https://golemlab.eu/graph/fleschEase
https://golemlab.eu/graph/SMOG
https://golemlab.eu/graph/ARI
https://golemlab.eu/graph/daleChallNew
https://golemlab.eu/graph/nominalRatio
```

```
https://golemlab.eu/graph/nominalRatioInverse


Question: Â£CuÃąntas historias de Valley of Tears (TV) hay publicadas?

Query:
```

Listing 4: Spanish question NFS Prompt Example

A.1.3. NFS Prompt Examples

ENGLISH

```
Your task is to translate a question in natural language into a SPARQL query for the
    GOLEM knowledge graph.
The query must follow these guidelines:
1. SPARQL queries must include the following prefix:
   prefix golem: <https://golemlab.eu/graph/>
2. Enclose SPARQL queries within <SPARQL> </SPARQL> tags.
3. You must generate 1 query(ies).
4. It is very important that you use only the predicates provided below.
5. Examples are provided below for guidance.

###
Predicates:
https://golemlab.eu/graph/numberOfComments
https://golemlab.eu/graph/numberOfKudos
https://golemlab.eu/graph/publicationStatus
https://golemlab.eu/graph/dateModified
https://golemlab.eu/graph/characters
https://golemlab.eu/graph/collections
https://golemlab.eu/graph/fandom
https://golemlab.eu/graph/publisher
https://golemlab.eu/graph/rating
https://golemlab.eu/graph/series
https://golemlab.eu/graph/story_id
https://golemlab.eu/graph/summary
https://golemlab.eu/graph/numberOfChapters
https://golemlab.eu/graph/datePublished
https://golemlab.eu/graph/keyword
https://golemlab.eu/graph/contentWarning
https://golemlab.eu/graph/numberOfWords
https://golemlab.eu/graph/socialRelationships
https://golemlab.eu/graph/datePackaged
https://golemlab.eu/graph/romanticCategory
https://golemlab.eu/graph/noOfPairings
https://golemlab.eu/graph/topPartner
https://golemlab.eu/graph/topPartnerPairings
https://golemlab.eu/graph/averageWordLength
https://golemlab.eu/graph/MSTTR
https://golemlab.eu/graph/bigramEntropy
https://golemlab.eu/graph/wordEntropy
https://golemlab.eu/graph/fleschGrade
https://golemlab.eu/graph/fleschEase
https://golemlab.eu/graph/SMOG
https://golemlab.eu/graph/ARI
https://golemlab.eu/graph/daleChallNew
https://golemlab.eu/graph/nominalRatio
https://golemlab.eu/graph/nominalRatioInverse

```

```

###
Examples:


    Question: How many chapters does Guardian of Hogwarts have?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT DISTINCT ?chapters WHERE {  ?
        story golem:title "Guardian of Hogwarts" . ?story golem:numberOfChapters ?
        chapters . }
    </SPARQL>

    ###


    Question: What is the average number of comments for stories from Glory of the
        Special Forces (TV)?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT AVG(?comments) AS ?
        avg_number_of_comments WHERE { ?story golem:fandom "Glory of the Special Forces
        (TV)" . ?story golem:numberOfComments ?comments . }
    </SPARQL>

    ###


    Question: Is the story Luna Lovegood and the Chamber of Innocence completed?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT DISTINCT ?status WHERE {  ?
        story golem:title "Luna Lovegood and the Chamber of Innocence" . ?story golem:
        publicationStatus ?status . }
    </SPARQL>

    ###


Question: How many 1984 - George Orwell stories are there?

Query:
```

Listing 5: English question NFS Prompt Example

SPANISH

```
Your task is to translate a question in natural language into a SPARQL query for the
    GOLEM knowledge graph.
The query must follow these guidelines:
1. SPARQL queries must include the following prefix:
   prefix golem: <https://golemlab.eu/graph/>
2. Enclose SPARQL queries within <SPARQL> </SPARQL> tags.
```

```
3. You must generate 1 query(ies).
4. It is very important that you use only the predicates provided below.
5. Examples are provided below for guidance.

###
Predicates:
https://golemlab.eu/graph/numberOfComments
https://golemlab.eu/graph/numberOfKudos
https://golemlab.eu/graph/publicationStatus
https://golemlab.eu/graph/dateModified
https://golemlab.eu/graph/characters
https://golemlab.eu/graph/collections
https://golemlab.eu/graph/fandom
https://golemlab.eu/graph/publisher
https://golemlab.eu/graph/rating
https://golemlab.eu/graph/series
https://golemlab.eu/graph/story_id
https://golemlab.eu/graph/summary
https://golemlab.eu/graph/numberOfChapters
https://golemlab.eu/graph/datePublished
https://golemlab.eu/graph/keyword
https://golemlab.eu/graph/contentWarning
https://golemlab.eu/graph/numberOfWords
https://golemlab.eu/graph/socialRelationships
https://golemlab.eu/graph/datePackaged
https://golemlab.eu/graph/romanticCategory
https://golemlab.eu/graph/noOfPairings
https://golemlab.eu/graph/topPartner
https://golemlab.eu/graph/topPartnerPairings
https://golemlab.eu/graph/averageWordLength
https://golemlab.eu/graph/MSTTR
https://golemlab.eu/graph/bigramEntropy
https://golemlab.eu/graph/wordEntropy
https://golemlab.eu/graph/fleschGrade
https://golemlab.eu/graph/fleschEase
https://golemlab.eu/graph/SMOG
https://golemlab.eu/graph/ARI
https://golemlab.eu/graph/daleChallNew
https://golemlab.eu/graph/nominalRatio
https://golemlab.eu/graph/nominalRatioInverse


###
Examples:


    Question: Â£CuÃąntos capÃŋtulos tiene this title is a wip of which i hate, like this
         fic and my life?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT DISTINCT ?chapters WHERE {  ?
        story golem:title "this title is a wip of which i hate, like this fic and my
        life" . ?story golem:numberOfChapters ?chapters . }
    </SPARQL>

    ###
```

```
Question: Â£CuÃąntos comentarios tienen de media las historias de éżŠèĮuãĄőãĆţãĆđãĆśãČĞãČłãĆń?

Query:
<SPARQL>
prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/> prefix gc: <https://ontology.golemlab.eu/>  SELECT AVG(?comments) AS ? avg_number_of_comments WHERE { ?story golem:fandom "éżŠèĮuãĄőãĆţãĆđãĆśãČĞãČłãĆń" . ?story golem:numberOfComments ?comments . }
</SPARQL>

###


Question: Â£EstÃą completada la historia A Small Steep Valley?

Query:
<SPARQL>
prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/> prefix gc: <https://ontology.golemlab.eu/>  SELECT DISTINCT ?status WHERE {  ? story golem:title "A Small Steep Valley" . ?story golem:publicationStatus ? status . }
</SPARQL>

###


Question: Â£CuÃąntas historias de Valley of Tears (TV) hay publicadas?

Query:
```

Listing 6: Spanish question NFS Prompt Example

A.1.4. a-DFSL Prompt Examples

Below are two examples of prompts for transforming natural language questions into SPARQL queries, one in English and one in Spanish.

ENGLISH

```
Your task is to translate a question in natural language into a SPARQL query for the GOLEM knowledge graph.
The query must follow specific guidelines to ensure accuracy and correctness:
1. SPARQL queries must include the following prefix:
   prefix golem: <https://golemlab.eu/graph/>
2. Enclose SPARQL queries within <SPARQL> </SPARQL> tags.
3. You must generate 1 query(ies).
4. It is very important that you use only the predicates provided below.
5. Examples are provided below for guidance.

###
Predicates:
https://golemlab.eu/graph/numberOfComments
https://golemlab.eu/graph/numberOfKudos
https://golemlab.eu/graph/publicationStatus
https://golemlab.eu/graph/dateModified
https://golemlab.eu/graph/characters
```

```
https://golemlab.eu/graph/collections
https://golemlab.eu/graph/fandom
https://golemlab.eu/graph/publisher
https://golemlab.eu/graph/rating
https://golemlab.eu/graph/series
https://golemlab.eu/graph/story_id
https://golemlab.eu/graph/summary
https://golemlab.eu/graph/numberOfChapters
https://golemlab.eu/graph/datePublished
https://golemlab.eu/graph/keyword
https://golemlab.eu/graph/contentWarning
https://golemlab.eu/graph/numberOfWords
https://golemlab.eu/graph/socialRelationships
https://golemlab.eu/graph/datePackaged
https://golemlab.eu/graph/romanticCategory
https://golemlab.eu/graph/noOfPairings
https://golemlab.eu/graph/topPartner
https://golemlab.eu/graph/topPartnerPairings
https://golemlab.eu/graph/averageWordLength
https://golemlab.eu/graph/MSTTR
https://golemlab.eu/graph/bigramEntropy
https://golemlab.eu/graph/wordEntropy
https://golemlab.eu/graph/fleschGrade
https://golemlab.eu/graph/fleschEase
https://golemlab.eu/graph/SMOG
https://golemlab.eu/graph/ARI
https://golemlab.eu/graph/daleChallNew
https://golemlab.eu/graph/nominalRatio
https://golemlab.eu/graph/nominalRatioInverse


###
Examples:


    Question: How many stories are there on Archive of Our Own?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT COUNT(DISTINCT ?story) as ?
        stories WHERE { ?story golem:story_id ?id }
    </SPARQL>

    ###


    Question: How many åijĂåŸŐ stories are there?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT (COUNT(DISTINCT ?story) as ?
        uploads) WHERE { ?story golem:fandom "åijĂåŸŐ" . }
    </SPARQL>

    ###

```

```
    Question: How many Kudos do Mr.Vampire (1985) stories get on average?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT AVG(?kudos) AS ?
        average_number_of_kudos WHERE { ?story golem:fandom "Mr.Vampire (1985)" . ?story
         golem:numberOfKudos ?kudos . }
    </SPARQL>

    ###


Question: How many 1984 - George Orwell stories are there?

Query:
```

Listing 7: English question a-DFSL Prompt Example

SPANISH

```
Your task is to translate a question in natural language into a SPARQL query for the
    GOLEM knowledge graph.
The query must follow specific guidelines to ensure accuracy and correctness:
1. SPARQL queries must include the following prefix:
   prefix golem: <https://golemlab.eu/graph/>
2. Enclose SPARQL queries within <SPARQL> </SPARQL> tags.
3. You must generate 1 query(ies).
4. It is very important that you use only the predicates provided below.
5. Examples are provided below for guidance.

###
Predicates:
https://golemlab.eu/graph/numberOfComments
https://golemlab.eu/graph/numberOfKudos
https://golemlab.eu/graph/publicationStatus
https://golemlab.eu/graph/dateModified
https://golemlab.eu/graph/characters
https://golemlab.eu/graph/collections
https://golemlab.eu/graph/fandom
https://golemlab.eu/graph/publisher
https://golemlab.eu/graph/rating
https://golemlab.eu/graph/series
https://golemlab.eu/graph/story_id
https://golemlab.eu/graph/summary
https://golemlab.eu/graph/numberOfChapters
https://golemlab.eu/graph/datePublished
https://golemlab.eu/graph/keyword
https://golemlab.eu/graph/contentWarning
https://golemlab.eu/graph/numberOfWords
https://golemlab.eu/graph/socialRelationships
https://golemlab.eu/graph/datePackaged
https://golemlab.eu/graph/romanticCategory
https://golemlab.eu/graph/noOfPairings
https://golemlab.eu/graph/topPartner
https://golemlab.eu/graph/topPartnerPairings
https://golemlab.eu/graph/averageWordLength
https://golemlab.eu/graph/MSTTR
```

```
https://golemlab.eu/graph/bigramEntropy
https://golemlab.eu/graph/wordEntropy
https://golemlab.eu/graph/fleschGrade
https://golemlab.eu/graph/fleschEase
https://golemlab.eu/graph/SMOG
https://golemlab.eu/graph/ARI
https://golemlab.eu/graph/daleChallNew
https://golemlab.eu/graph/nominalRatio
https://golemlab.eu/graph/nominalRatioInverse


###
Examples:


    Question: Â£CuÃąntas historias de [[fandom]] hay publicadas?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT (COUNT(DISTINCT ?story) as ?
        uploads) WHERE { ?story golem:fandom "[[fandom]]" . }
    </SPARQL>

    ###


    Question: Â£CuÃąntas historias se publican al aÃśo?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT (COUNT(DISTINCT ?story) as ?
        uploads) WHERE {  ?story golem:story_id ?id }
    </SPARQL>

    ###


    Question: Â£CuÃąntos autores han publicado alguna historia?

    Query:
    <SPARQL>
    prefix golem: <https://golemlab.eu/graph/>  prefix dc: <http://purl.org/dc/terms/>
        prefix gc: <https://ontology.golemlab.eu/>  SELECT (COUNT(DISTINCT ?author) as ?
        uploads) WHERE { ?story golem:author ?author . }
    </SPARQL>

    ###


Question: Â£CuÃąntas historias de Valley of Tears (TV) hay publicadas?

Query:
```

Listing 8: Spanish question a-DFSL Prompt Example