PolyglotQL: A Pipeline for Multilingual Text-to-SPARQL Dataset Generation

Julio Perez[1,2], Fabio Barth[2], Georg Rehm[2,3]
[1]Technical University of Berlin, Germany
[2]Deutsches Forschungszentrum für Künstliche Intelligenz GmbH (DFKI), Berlin, Germany
[3]Humboldt-Universität zu Berlin, Germany
perezduranjulio@gmail.com, fabio.barth@dfki.de, georg.rehm@dfki.de

Abstract

We present PolyglotQL, an open-source ETL (Extract, Transform, Load) pipeline for systematically creating multilingual text-to-SPARQL datasets, along with an accompanying framework for evaluating text-to-SPARQL generation models. PolyglotQL provides an extensible and modular architecture that aggregates, normalizes, and augments heterogeneous question–SPARQL pairs from established text-to-SPARQL datasets. With this pipeline, we automatically construct a bilingual English–German dataset featuring contextualized entity and relationship mappings as well as automatically translated and aligned question pairs. We also conduct an empirical evaluation using two multilingual open large language models under two distinct contextualization settings. The results show consistent performance improvements when explicit grounding information is provided, highlighting the benefits of structured context in multilingual semantic parsing.

Keywords: SPARQL, Text2SPARQL, dataset, knowledge graph, multilingual, LLM, QALD

1. Introduction

Mapping natural language questions to SPARQL queries (text-to-SPARQL) remains a core problem at the intersection of natural language processing and the Semantic Web (Usbeck et al., 2024). In recent years, the multilingual capabilities of Large Language Models (LLMs) have improved massively across many general-domain tasks, raising the question of whether they have also improved at mapping multilingual inputs to SPARQL queries. Prior work provides multiple benchmarks and specialized datasets in various languages and from different knowledge graphs, e. g., the QALD challenges (Usbeck et al., 2024; Perevalov et al., 2022; Usbeck et al., 2018), but there is a lack of an integrated, reproducible framework that (i) aggregates these heterogeneous sources, (ii) supports multilingual augmentation, and (iii) provides tooling for preprocessing, contextual augmentation, and standardized evaluation.

In this paper, we address this gap by introducing PolyglotQL, a comprehensive dataset pipeline accompanied by an evaluation toolkit for text-to-SPARQL tasks. The pipeline is a fully open source toolkit for generating a homogeneous multilingual text-to-SPARQL corpus from established datasets supporting query generation of 15 knowledge graphs, prominently Wikidata and DBpedia. With this pipeline, we create a bilingual data split of English and German questions to fine-tune multilingual language models for the text-to-SPARQL task. We show that training language models on text-to-SPARQL tasks does not help the models achieve strong performances regardless of the language. However, when the input questions are enhanced with entity and relationship mappings, the models improve significantly in both languages.

Our contributions can be summarised as follows:

1. **PolyglotQL, a modular ETL pipeline** to aggregate, normalize, and augment question–SPARQL pairs from QA-over-KG datasets (scripts, documented and open-sourced). The tool is designed to be extensible to other Knowledge Graphs (KGs) and languages (Figure 1).
2. **A bilingual text-to-SPARQL dataset** containing a processed English–German split with JSON-formatted entity and relationship mappings (context column) produced using the pipeline. We provide counts and annotation statistics and release the split for benchmarking. The dataset serves as a showcase for the pipeline's ability to produce a much larger, consolidated, and multilingual corpus (roughly 895,954 examples), but this is not the focus of this paper's evaluation.
3. **An evaluation** of two multilingual open LLM families under two setups: v1, with no explicit contextual ID mappings in training or evaluation. And v2, with the context of entity/property IDs in training and evaluation. We demonstrate that explicit context yields significant gains and analyze the remaining failure modes.

The remainder of this paper describes related work, the dataset and pipeline, including preprocessing and augmentation steps, the evaluation methodology, experimental results, and an analysis of limitations and future work.

Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026), pages 6674–6684
11–16 May 2026.

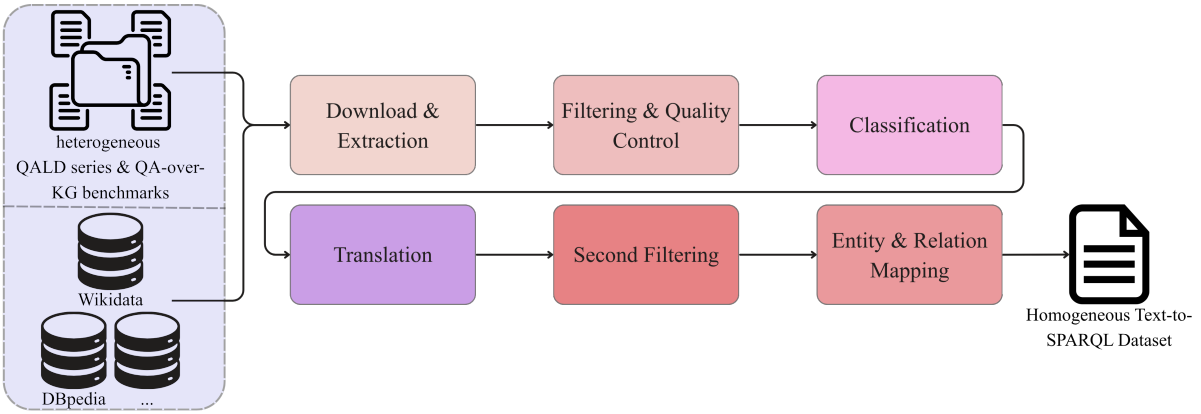

Figure 1: An overview of the pipeline. Datasets such as the QLAD series and public KGs are normalized and collected to form a single large text-to-SPARQL Dataset. The ETL pipeline can also translate questions from the QA-over-KG datasets into various languages to improve multilingual accessibility.

2. Related Work

The task of translating natural language questions into SPARQL queries has been addressed by various research groups, particularly in the context of challenges such as Question Answering over Linked Data (QALD) 10th version, with its corresponding benchmark (Usbeck et al., 2024). These efforts, however, present a fragmented landscape in terms of methodology, multilingual support, and resource availability. A recent contribution by Brei et al. (2024a) explores the use of small language models (< 1B parameters) for translating English text to SPARQL queries across different knowledge graphs, including Wikidata. A significant aspect of their work is the public release of their fine-tuning scripts, promoting reproducibility. The QALD-10 challenge itself showcases a variety of competing systems. While the official results primarily focus on English, several participating systems have multilingual capabilities. For instance, the WDAqua-core0 system (Diefenbach et al., 2017) supports French, German, and Italian as well. Similarly, the system by Shivashankar et al. (2022) handles both German and English. However, a common limitation is the lack of publicly available source code, hindering verification and extension by the research community. An exception is the work by Baramiia et al. (2022), which provides code for their English-based approach. Although they suggest their method can be extended to other languages through transformer fine-tuning, the implementation for this extension is not provided. Regarding evaluation, the community has largely relied on established frameworks. GERBIL (Usbeck et al., 2019), used for QALD-10, provides a comprehensive evaluation environment with a live endpoint. In contrast, other approaches, such as that of Brei et al. (2024a), offer more basic evaluation scripts that measure the number of correct answers, highlighting a variance in evaluation rigor across different works. Existing systems and benchmarks are valuable, but they typically focus on single challenge datasets, a single target KG (often Wikidata), and a single language (frequently English). Moreover, many high-performing systems do not release full pipelines or multilingual augmentations, which hampers reproducibility and cross-study comparison. Our work builds on these efforts and contributes an end-to-end pipeline and dataset targeted specifically at robust multilingual text-to-SPARQL modeling and reproducible evaluation.

3. PolyglotQL Overview

To generate datasets for future research use, we structure the pipeline around several key design goals. To ensure broad coverage, we combine multiple QA-over-KG datasets to capture a diverse range of query patterns. We also emphasize multilinguality, focusing on English and German first, by building the preprocessing pipeline to be able to construct data splits in up to 12 other languages. Moreover, we ensure reproducibility by publishing the pipeline on GitHub. We ground the dataset by including entity and relationship ID mappings in a machine-readable context field, reducing ambiguity in entity linking. Finally, for evaluation, we provide scripts that can execute both gold and predicted SPARQL queries against live endpoints.

3.1. Preprocessing and Data Augmentation

We implement a modular pipeline consisting of six sequential steps:

1. **Dataset Download and Extraction:** Retrieve and unpack archives, traverse repository trees, and collect raw artifacts.

2. **Filtering and Quality Control:** Remove trivially invalid entries and apply heuristics to detect malformed records.
3. **Classification:** Distinguish statements from valid questions using an LLM-based classifier (Gemma 27B 8-bit quantized).
4. **Translation:** Generate parallel target-language text (German in this work) using an open-source high-quality translation model (NousResearch/Hermes family in 4-bit mode) to enhance the original datasets.
5. **Second Filtering:** Filter out poorly translated samples and translation artifacts (repetitions, truncations) using length-based heuristics and manual spot checks.
6. **Entity and Relation Mapping:** Generate JSON-formatted entity (QID) and property (PID) mappings by leveraging gold SPARQL queries and few-shot LLM prompting to align textual mentions with identifiers.

The following sections provide a brief description of each step.

3.1.1. Dataset Download and Extraction

The first step automates the retrieval of source datasets (archives, Git repositories, release assets). The source formats of all datasets are heterogeneous, including XML, JSON, TSV, custom markup files, and files within nested folder systems and repositories. We therefore implement parsers for each major source, recursive scraping of repository trees (e. g., the QALD repository), and schema normalization to a unified format: `text_query`, `language`, `sparql_query`, `knowledge_graph`, and optional `context`. The pipeline saves the extracted samples in standardized intermediate JSON files, for filtering and quality control.

As our pipeline aggregates datasets from the QALD series (Usbeck et al., 2024; Perevalov et al., 2022; Usbeck et al., 2018) and various QA-over-KG benchmarks (Talcmor and Berant, 2018; Cui et al., 2021; Gu et al., 2021; Su et al., 2016; Trivedi et al., 2017; Dubey et al., 2019; Kaffee et al., 2019; Korablinov and Braslavski, 2020; Rybin et al., 2021; Yih et al., 2016), our initial dataset contains roughly 895,954 examples spanning 12 languages and 15 KGs. However, that larger collection is background material for the pipeline's capability and is not the primary target of the experiments reported here, as we focus our work on the German-English split.

3.1.2. Filtering and Quality Control

We apply an automatic filter by removing short (less than four characters) or empty text entries, as these commonly reflect dataset artifacts. We then perform manual spot-checking on a random sample of 200 English and German samples, respectively, and remove commonly occurring nonsensical queries (e. g., "not applicable", "I have no idea") that bypass the automatic filters. By identifying nonsensical queries, we generate a blocklist of queries that are deleted from the entire dataset for each language.

3.1.3. Classification

To detect and remove declarative statements or other non-question entries, we use a classifier implemented with the Gemma 27B model (quantized to 8-bit for memory efficiency) (Gemma et al., 2025). Each sample is classified as 'question' or 'statement'. Entries flagged as statements are removed from the dataset after being double-checked by a human annotator.

3.1.4. Translation

Translation expands the bilingual coverage by generating German parallel samples from the much larger English split of Wikidata-targeted examples. For transparency reasons, we use an open-source translation model rather than a commercial API for translating the evaluation samples. We use the Hermes-3 model published by NousResearch with 4-bit precision (Teknium et al., 2024). The pipeline is, however, agnostic to the specific translation backend. We limit large-scale translation to German to keep quality control manageable: approximately 200,000 additional German rows are generated before subsequent filtering.

3.1.5. Second Filtering

Because automated translation sometimes generates severe artifacts, a second filtering stage is necessary. We apply language-specific heuristics to remove malformed translations. These include detection of repeated tokens/phrases, extreme length deviations relative to source queries, and language-detection mismatches. These filtering steps remove problematic translations while preserving high-quality bilingual pairs. With the second filter, we end up with 157,246 German samples. We also spot-check 200 translated German samples by native-speaking annotators to validate the translation quality.

3.1.6. Entity and Relation Mapping

A key enhancement is the automatic creation of a JSON `context` column for KG-targeted information from Wikidata. The process leverages the gold SPARQL queries already present in many sources. A few-shot prompting pipeline (using

high-capability LLMs such as Hermes-3 (Teknium et al., 2024) for English and Llama-3.3 (Grattafiori, 2024) variants for German) extracts QIDs and PIDs from the SPARQL query and aligns them with their textual mentions in the natural language question. Not mentions are not added to the `context` field. These mappings enhance the data points crucially as they work like a lookup table for relevant entities inside the question. Each output is validated (using a syntactic JSON check) and stored in the `context` field. This produces **318,161** contextual annotation mappings (160,915 English and 157,246 German) as well as 784 annotation mappings for the QALD-10 test.

4. The Bilingual Datasplit

To showcase how PolyglotQL works, we generate an English-German dataset split containing 318,161 examples with explicit JSON-formatted contextual annotations (entity and relationship identifier mappings), which serve as the primary resource for our experiments. These contextual annotations are generated for Wikidata-targeted examples and are distributed roughly evenly across the two languages: 160,915 English samples and 157,246 German samples (see Table 1). Note that we do not generate a dataset of all possible languages, which would yield a dataset with more than 800,000 samples across 12 languages, for two reasons. First, the translation of parallel samples, as well as the classification, is both computationally and resource-expensive. Second, the pipeline is not fully automatic, and human annotators are needed to spot-check the translations as well as QIDs and PIDs mappings. However, to the best of our knowledge, this data split is the largest publicly available German text-to-SPARQL dataset.

Dataset	English	German
Bilingual Split	160,915	157,246
Train	128,733	125,796
Validation	16,091	15,725
Test	16,091	15,725
QALD-10	392	392

Table 1: Dataset statistics (all counts refer to examples that include contextual JSON mappings).

5. Experimental Setup

We evaluate our approach by executing both gold and generated SPARQL queries on the live Wikidata SPARQL endpoint and comparing the result sets using QALD-style set-based precision, recall, and macro-averaged F1. Non-executable or endpoint-failing queries are penalized (assigned zero for that instance). We also compare and categorize generated queries with the gold queries (Exact Match, Both Empty, Partial Match, No Overlap, Execution Failure) to better analyze failure modes. Early metrics include counts of executable queries and exact matches. We use formal F1 metrics for reporting results similar to previous work (Usbeck et al., 2024; Perevalov et al., 2022; Usbeck et al., 2018).

5.1. Benchmarks and Splits

For each language, we evaluate all models on a random test split of our bilingual dataset, containing 10% of the examples, which sums up to a 31,816 large bilingual test set. For comparison, we also evaluate the base and fine-tuned models on the test split of the QLAD-10 challenge (Usbeck et al., 2024), the latest challenge in the QLAD series. Like all test sets from the QLAD series, the QLAD-10 test set is not parsed into the processing pipeline and is not used for fine-tuning.

5.2. Fine-Tuning and Model selection

We apply two fine-tuning setups for the models:

- **v1 setup (no context):** In this setup, the models are trained without contextual ID mappings.
- **v2 setup (with context):** the content of the Wikidata-focused `context` field is parsed along with the initial question.

For comparability, we train all models in both setups with the same samples, seeds, and parameters.

We evaluate fine-tuned models using QLoRA (4-bit) with LoRA adapters for parameter-efficient fine-tuning. As a reference point, we also evaluate the off-the-shelf instruction-tuned variants of each model family – *Mistral-7B-Instruct-v0.1* and *Occiglot-7b-eu5-instruct* – without any task-specific fine-tuning. These instruction-tuned models are prompted with the v2 input format (i. e., the system prompt includes the entity and relationship ID mappings as structured context; see Appendix A for a concrete example). We refer to these non-fine-tuned evaluations as **base** in our results tables. We evaluate two models:

- **Mistral 7B v0.1** is a 7-billion-parameter open-source language model that was primarily trained on English text, but it also has some multilingual capabilities (Jiang, 2023).
- **Occiglot-7b-eu5** is a 7-billion-parameter open-source language model trained on the top five EU languages (English, Spanish, French, German, and Italian), providing strong multilingual capabilities (Avramidis, 2024).

We chose those two models because they are current, multilingual, open-source large language models with predominantly German and English pre-training data. All major training parameters (epochs, optimizer, mixed precision, gradient accumulation) are recorded in the repository. We train all models on a single A100 GPU with 80 GB of memory for efficiency.

5.3. Metrics and Error Categories

To evaluate the performance, we primarily report macro-averaged precision, recall, and F1 score, following the evaluation protocol established in the QALD benchmarks. These metrics treat all queries equally, regardless of their frequency or answer cardinality, and are widely used in semantic question answering.

In addition to these core metrics, we consider several secondary indicators that offer further insight into system behavior. First, we measure the percentage of executable queries, which reflects the robustness of generated outputs with respect to the underlying knowledge base interface. Second, we analyze the distribution of system outputs across predefined error categories:

- **Syntactic Failure:** Cases where the generated query cannot be parsed or executed.
- **Generated-Empty:** Cases where the query executes but returns an empty result set.
- **No-Overlap:** Cases where the predicted answer set does not overlap with the gold standard.
- **Partial-Overlap:** Cases where the predicted answer set overlaps with the gold standard but it is not the same.
- **Exact-Match:** Cases where the predicted and gold answer sets are identical.

Finally, we report per-model breakdowns on the QALD-10 benchmark. This facilitates direct comparison with existing systems and supports reproducibility within the community.

6. Results

6.1. Test Split Results

Table 2 summarizes representative F1 results on held-out test splits (v2 context setting) for all models. We evaluate the fine-tuned version for each setup and compare its performance with that of the base model. The results show that including explicit context (v2) massively improves performance for both models compared to v1. In particular, we see the largest improvement (a factor of 5 for the Mistral model or 10 for the Occiglot model) from the base model to the v1 models, demonstrating strong gains from fine-tuning, even without additional context about the mapping of entities/relationships to Wikidata IDs. The best performing model with 82.8 % F1 in English and 71.8 % in German is the fine-tuned Mistral model. However, both models achieve higher F1 scores in English than in German, except for the Occiglot model without context (v1).

Overall, we observe that models primarily struggle with generating KG IDs for specific entities. By enhancing the input prompt with entity mappings from the question entities, we can see a substantial improvement.

6.2. QALD-10 Results

On the QALD-10 test set, the Mistral v2 fine-tuned model achieves an F1 score around 28.5 % (see Table 3). Compared to the submitted systems of the QALD-10 challenge, the v2 models achieve results comparable to the worst-performing systems (See Table 3). The Mistral model achieves results that are even 10 percentage points better than the Singh & Gavrilev submission. However, the performance is worse than the top submissions of the chanotable that the systems submitted to QALD-10 are pipelines draines that are fully optimized for that task, which makes the comparison slightly biased, as we did not perform any pre- or post-processing on the model inputs or outputs. Moreover, all published results for the QALD-10 challenge are only for the English split and not for multilingual setups.

Like in the test set, the Mistral model improves its performance through fine-tuning in English more than in German. A notable observation is that the Occiglot model achieves a higher F1 score in German than in English, across both setups and fine-tuning. Compared to Mistral, Occiglot has been pre-trained on a data split of the same size for both English and German, which could explain why the model performs better in German than in English.

6.3. Error Analysis

For more insights on the failed generated queries, we categorize the errors into five categories (See Table 4). The analysis of error categories shows that both Mistral and Occiglot have a substantial reduction in syntactic failures after fine-tuning, regardless of the setup, confirming that fine-tuning effectively stabilizes query generation and improves syntactic correctness. While the baseline models frequently produce failing queries of nearly 300 in both languages, this number drops to close to zero in the v2 fine-tuned versions, showing that the models consistently learn to generate structurally valid SPARQL queries in any language.

Model	English			German		
	P	R	F1	P	R	F1
Mistral-7B-v0.1 (fine-tuned, v2)	0.910	**0.829**	**0.828**	0.836	0.720	0.718
Mistral-7B-v0.1 (fine-tuned, v1)	0.624	0.227	0.227	0.748	0.101	0.100
Mistral-7B-v0.1 (base)	0.389	0.048	0.046	0.369	0.044	0.042
occiglot-7b-eu5 (fine-tuned, v2)	0.891	0.806	0.805	0.851	**0.728**	**0.727**
occiglot-7b-eu5 (fine-tuned, v1)	**0.988**	0.301	0.300	**0.883**	0.306	0.302
occiglot-7b-eu5 (base)	0.204	0.022	0.021	0.189	0.022	0.022

Table 2: Model results on the test sets of macro Precision (P), Recall (R), and F1.

Model	English			German		
	P	R	F1	P	R	F1
Mistral-7B-v0.1 (fine-tuned, v2)	0.661	0.284	**0.285**	0.642	0.263	**0.259**
Mistral-7B-v0.1 (fine-tuned, v1)	**0.703**	0.069	0.069	0.673	0.056	0.056
Mistral-7B-v0.1 (base)	0.220	0.016	0.014	0.223	0.018	0.013
occiglot-7b-eu5 (fine-tuned, v2)	0.589	0.194	0.191	0.692	0.234	0.230
occiglot-7b-eu5 (fine-tuned, v1)	0.770	0.044	0.044	0.696	0.069	0.069
occiglot-7b-eu5 (base)	0.254	0.023	0.022	0.151	0.008	0.005
SPARQL-QA (Borroto et al., 2023)	–	–	0.595	–	–	–
Baramiia (Baramiia et al., 2022)	–	–	0.428	–	–	–
Singh & Gavrilev (Usbeck et al., 2024)	–	–	0.195	–	–	–

Table 3: Model results on the QALD-10 test set of macro Precision (P), Recall (R), and F1. The last three results by Borroto et al. (2023), Baramiia et al. (2022), and Singh & Gavrilev (Usbeck et al., 2024) are selected submission results from the official QALD-10 challenge.

However, the reduction in syntactic errors results in a higher proportion of empty result queries, with Mistral and Occiglot generating 245 and 264 queries, respectively, that yield empty results without `context` (averaged across English and German from Table 4). This suggests that while the models now generate executable queries more reliably, they sometimes fail to correctly ground entities or relations, resulting in valid but semantically empty outputs. The introduction of entity linking in the v2 setup shifts the error categories from structural to semantic: rather than producing invalid syntax, the models generate executable but mismatched queries. This trade-off is consistent across both languages and models, although it is slightly higher for Occiglot. The no-overlap and partial-overlap categories also increase after fine-tuning, reflecting that the models occasionally retrieve partially relevant results even when the overall answer set diverges from the gold standard. Notably, the fine-tuned v2 models exhibit a slight increase in partially correct outputs, suggesting that fine-tuning improves semantic alignment even when complete accuracy is not achieved.

When comparing the two models, Mistral demonstrates more consistent improvements across both English and German, with fewer syntactic and semantic errors after fine-tuning. Occiglot, in contrast, maintains a slightly higher rate of empty results in German, which may be related to its more balanced multilingual pre-training, which occasionally leads to less precise entity selection in specific contexts.

Finally, it is worth noting that most generated queries across all fine-tuned systems are executable. This high execution rate highlights the robustness of the fine-tuning procedure and its ability to produce structurally coherent queries in both English and German. The remaining differences between Mistral and Occiglot are therefore not due to query validity, but rather to semantic precision.

Reproducibility Notes

The entire pipeline, including translation and context-generation code, model training scripts, and evaluation tooling, is published in the project repository and Hugging Face dataset card. Exact training arguments (QLoRA/LoRA settings, LoRA rank, learning rate, batch configuration) and seeds used for sampling are recorded in the repository per-run logs to enable reproduction. See the appendices and repository for the full details.[1]

[1] https://github.com/julioc-p/polyglotql

Setting	Language	Model	Syntactic failure	Empty results	No-overlap	Partial-overlap
Base	English	Mistral	298	80	6	2
		Occiglot	278	91	13	1
	German	Mistral	295	82	7	2
		Occiglot	318	57	12	1
Fine-tuned (v1)	English	Mistral	6	248	110	0
		Occiglot	18	283	71	1
	German	Mistral	20	241	107	0
		Occiglot	8	245	109	0
Fine-tuned (v2)	English	Mistral	1	148	129	5
		Occiglot	3	157	154	5
	German	Mistral	1	151	133	5
		Occiglot	1	182	116	5

Table 4: Error Category Distribution for QALD-10 Test Set. Values are absolute counts.

7. Analysis and Discussion

7.1. Entity/relation Linking as the Primary Bottleneck

A central finding across all experiments is that entity and predicate linking remains the dominant source of error. In the v1 setup (without explicit identifier context), most queries fail due to incorrect or missing QIDs/PIDs, resulting in empty or non-overlapping answers. These errors account for the majority of non-syntactic failures. Providing explicit identifier mappings (v2) drastically reduces this ambiguity, allowing the model to focus on the compositional structure of the query, which leads to substantial improvements in F1 scores across both languages.

7.2. Reduction of Syntactic Failures and Shifting Error Profiles

Table 4 shows that syntactic failures, which dominate the baseline models (nearly 300 instances per language), are almost completely eliminated after fine-tuning. This reduction indicates that the models successfully internalize the grammar and surface structure of SPARQL through parameter-efficient training. Recent work (e. g., Brei et al., 2024b) has shown that smaller LLMs (<8B parameters) often struggle to generate syntactically valid and executable SPARQL queries, with a large proportion of outputs failing to parse or execute correctly. In contrast, our experiments demonstrate that both Mistral-7B and Occiglot-7B achieve a high proportion of executable queries across languages. This finding highlights the impact of targeted fine-tuning with structured supervision and entity grounding, which effectively mitigates the structural instability commonly observed in smaller or similarly sized models.

The high number of Partial-Overlap and No-Overlap cases in the fine-tuned v2 setup for both models suggests that fine-tuning also enhances partial semantic alignment, allowing the models to approximate correct query intent even when full correctness is not achieved. Compared to the v1 setting, these categories indicate a gradual progression from invalid outputs toward semantically relevant but incomplete retrievals.

7.3. Cross-lingual Behavior and Model-specific Tendencies

Both models achieve high execution rates in English and German, but subtle differences emerge in their error distributions. Mistral-7B, trained primarily on English data, benefits more strongly from fine-tuning in English but transfers these gains relatively well to German. Occiglot-7B, which has balanced multilingual pre-training, performs more consistently across both languages yet shows a slightly higher proportion of empty queries in German, indicating more conservative or uncertain grounding.

The overall high proportion of executable queries across all fine-tuned setups confirms that both architectures learn robust SPARQL generation patterns. Most remaining errors are not due to invalid syntax but to semantic grounding mismatches.

7.4. Structural Complexity Remains a Challenge

Despite the strong improvements achieved through contextual entity grounding (v2), structurally complex queries such as those involving multi-hop joins or nested filters remain challenging. The QALD-10 dataset, which contains a higher proportion of such queries, still exposes limitations in the ability to generalize beyond simple retrieval or single-hop structures. The persistence of these structural errors highlights that fine-tuning enhances stability and correctness in surface-level syntax but does not yet equip models with a comprehensive logical understanding or reasoning depth.

8. Conclusions

We present PolyglotQL, an open ETL pipeline for text-to-SPARQL generation together with a bilingual English–German split of text-to-SPARQL samples, and report empirical findings that explicit contextual grounding (using entity/property IDs) drastically reduces the entity-linking bottleneck and substantially improves model performance. Moreover, the fine-tuned Mistral model shows comparable results with SOTA solutions on established benchmarks. PolyglotQL and the dataset (and sampled model checkpoints) are released to facilitate reproducible research and further experimentation with multilingual and multi-KG text-to-SPARQL tasks. To the best of our knowledge, this is the largest publicly available German text-to-SPARQL dataset.

Limitations

Reliance on LLMs for preprocessing and augmentation Several preprocessing stages (translation, statement classification, and context generation) use other LLMs. While this enabled scaling, it introduces biases and occasional errors originating from those LLMs (e. g., mistranslations or misaligned identifiers). We propose deterministic Wikidata API lookups as a future alternative for context generation to reduce reliance on LLM outputs. Until such deterministic pipelines are integrated, downstream models will reflect the artifacts of upstream LLMs.

Limited manual verification at dataset scale Exhaustive manual inspection of nearly 900k aggregated examples is infeasible. We applied automated heuristics and spot-checks, but residual noise and labeling errors remain possible. This constrains the strength of claims about absolute model performance and suggests human-in-the-loop validation for critical downstream use-cases or for extending to lower-resource languages.

Language and KG coverage in experiments Although the pipeline can produce splits for many languages and KGs, our experimental evaluation concentrates on English and German and primarily on Wikidata-targeted examples. Generalization to typologically distant languages or other KGs (DBpedia, Freebase variants) remains to be validated. Future work should extend evaluation to additional languages and KGs, balancing automated augmentation with careful quality control.

Statistical testing and computational constraints The experiments were constrained by computational and time limitations; some configurations were not exhaustively repeated with multiple seeds, and comprehensive statistical testing (confidence intervals/hypothesis tests) remains future work. We documented the seeds and the reasons for the pragmatic experimental design; however, broader replication with more seeds would strengthen claims of minor effects.

Ethical Considerations and Broader Impact

The dataset and pipeline assemble many open resources; licensing compliance and attributions are retained in the released artifacts. Large-scale automated translation and LLM-based augmentation may propagate biases (e. g., gendered label shifts) or introduce artifacts – users should apply downstream fairness and robustness analyses. We encourage future work to integrate human-in-the-loop validation and deterministic lookups.

Acknowledgements

This work was supported by the consortium NFDI for Data Science and Artificial Intelligence (NFDI4DS)[2] as part of the non-profit association National Research Data Infrastructure (NFDI e. V.), funded by the Federal Republic of Germany and its states through the German Research Foundation (DFG) project NFDI4DS (no. 460234259). Further support was provided from the European Union's Horizon Europe research and innovation programme under grant agreement no. 101189745 (HIVEMIND).[3]

Bibliographical References

Eleftherios et al. Avramidis. 2024. Occiglot at WMT24: European open-source large language models evaluated on translation. In *Proceedings of the Ninth Conference on Machine Translation*, pages 292–298, Miami, Florida, USA. Association for Computational Linguistics.

Nikita Baramiia, Alina Rogulina, Sergey Petrakov, Valerii Kornilov, and Anton Razzhigaev. 2022. Ranking approach to monolingual question answering over knowledge graphs. In *NLIWoD@ ESWC*, pages 32–37.

Manuel Borroto, Francesco Ricca, Bernardo Cuteri, and Vito Barbara. 2023. Sparql-qa enters the qald challenge. In *QALD-10 – The 10th challenge on question answering over linked data: Shifting from DBpedia to Wikidata as a KG for KGQA*.

[2] https://www.nfdi4datascience.de
[3] https://hivemind-project.eu

Felix Brei, Johannes Frey, and Lars-Peter Meyer. 2024a. Leveraging small language models for text2sparql tasks to improve the resilience of ai assistance. *arXiv preprint arXiv:2405.17076*.

Felix Brei, Johannes Frey, and Lars-Peter Meyer. 2024b. Leveraging small language models for text2sparql tasks to improve the resilience of ai assistance.

Ruixiang Cui, Rahul Aralikatte, Heather Lent, and Daniel Hershcovich. 2021. Multilingual Compositional Wikidata Questions. *arXiv e-prints*, page arXiv:2108.03509.

Dennis Diefenbach, Kamal Singh, and Pierre Maret. 2017. Wdaqua-core0: A question answering component for the research community. In *Semantic Web Challenges*, pages 84–89, Cham. Springer International Publishing.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *Proceedings of the 18th International Semantic Web Conference (ISWC)*. Springer.

Team Gemma, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. 2025. Gemma 3 technical report.

Aaron et al. Grattafiori. 2024. The llama 3 herd of models.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the Web Conference 2021*, pages 3477–3488. ACM.

Albert Q. et al. Jiang. 2023. Mistral 7b.

Lucie-Aimée Kaffee, Kemele M. Endris, Elena Simperl, and Maria-Esther Vidal. 2019. Ranking knowledge graphs by capturing knowledge about languages and labels. In *Proceedings of the 10th International Conference on Knowledge Capture,*

K-CAP 2019, Marina Del Rey, CA, USA, November 19-21, 2019. ACM.

Vladislav Korablinov and Pavel Braslavski. 2020. RuBQ: A Russian Dataset for Question Answering over Wikidata. *arXiv e-prints*, page arXiv:2005.10659.

Aleksandr Perevalov, Dennis Diefenbach, Ricardo Usbeck, and Andreas Both. 2022. Qald-9-plus: A multilingual dataset for question answering over dbpedia and wikidata translated by native speakers.

Ivan Rybin, Vladislav Korablinov, Pavel Efimov, and Pavel Braslavski. 2021. Ru{bq} 2.0: An innovated russian question answering dataset. In *Eighteenth Extended Semantic Web Conference - Resources Track*.

Kanchan Shivashankar, Khaoula Benmaarouf, and Nadine Steinmetz. 2022. From graph to graph: Amr to sparql. In *Proceedings of the 7th Natural Language Interfaces for the Web of Data (NLIWoD) co-located with the 19th European Semantic Web Conference (ESWC 2022), Hersonissos, Greece, 29th May*.

Yu Su, Huan Sun, Brian Sadler, Mudhakar Srivatsa, Izzeddin Gür, Zenghui Yan, and Xifeng Yan. 2016. On generating characteristic-rich question sets for QA evaluation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 562–572, Austin, Texas. Association for Computational Linguistics.

Alon Talcmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *NAACL*.

Ryan Teknium, Jeffrey Quesnelle, and Chen Guang. 2024. Hermes 3 technical report.

Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. 2017. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Felix Conrads, Michael Röder, and Giulio Napolitano. 2018. 8th challenge on question answering over linked data (qald-8) (invited paper). In *Semdeep-/NLIWoD@ISWC*.

Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrads, Jonathan Huthmann, Axel-Cyrille Ngonga-Ngomo, Christian Demmler, and Christina Unger. 2019. Benchmarking question answering systems. *Semantic Web*, 10(2):293–304.

Ricardo Usbeck, Xi Yan, Aleksandr Perevalov, Longquan Jiang, Julius Schulz, Angelie Kraft, Cedric Möller, Junbo Huang, Jan Reineke, and Axel-Cyrille et al. Ngonga Ngomo. 2024. Qald-10–the 10th challenge on question answering over linked data: Shifting from dbpedia to wikidata as a kg for kgqa. *Semantic Web*, 15(6):2193–2207.

Wen-tau Yih, Matthew Richardson, Chris Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206. Association for Computational Linguistics.

Appendix: Example Context and Prompts

Listing 1: Example of generated English context.

```
{
  "entities": {
    "United States Army": "Q9212",
    "character": "Q95074"
  },
  "relationships": {
    "spouse": "P26",
    "instance of": "P31",
    "employer": "P108"
  }
}
```

Listing 2: System prompt for English context generation.

```
You are a helpful assistant that extracts
    Wikidata entities and properties from
    SPARQL queries.
Output a valid JSON dictionary with no
    trailing characters. Format:
{
  "entities": {"ENTITY_LABEL_IN_QUESTION":
      "QID"},
  "relationships": {"
      RELATIONSHIP_LABEL_IN_QUESTION": "PID
      "}
}
Only use what you see in the SPARQL, no
    inferred knowledge. The labels in the
    JSON should correspond to how they are
    expressed or implied in the question.
```

The code and logs are available at: https://github.com/julioc-p/polyglotql

A. Prompt Structure

Both the base (non-fine-tuned instruction-tuned) and fine-tuned v2 models receive the same input format: a system prompt containing the task instruction and the structured entity/relationship context, followed by the user's natural language question. The v2 fine-tuned models are trained on this format; the base models receive it at inference time without prior task-specific training.

Listing 3: Example v2 input prompt (English). The system message embeds the entity and relationship ID mappings as structured JSON context. The user message contains only the natural language question.

```
[
  {
    "role": "system",
    "content": "You are an expert text to
      SparQL query translator. Users will
      ask you questions in English and you
      will generate a SparQL query based on
      the provided context.
      CONTEXT:
      {\"entities\":
        {\"Barnard College\": \"Q167733\",
         \"American\": \"Q30\",
         \"female\": \"Q6581072\"},
       \"relationships\":
        {\"instance of\": \"P31\",
         \"employer\": \"P108\",
         \"gender\": \"P21\",
         \"country of citizenship\": \"P27
             \"}}"
  },
  {
    "role": "user",
    "content": "Who was Barnard College's
      American female employee?"
  }
]
```

In the v1 setup (no context), the input contains only the question and knowledge graph name, without entity/relationship mappings:

Listing 4: Example v1 input prompt (English). No entity or relationship ID mappings are provided.

```
[
  {
    "role": "user",
    "content": "Write a SparQL query that
      answers this request: 'Who was
          Barnard
      College's American female employee?'
      from the knowledge graph Wikidata."
  }
]
```