Extending KGQA Benchmarks to Low-Resource Languages: The Udmurt QALD-9-plus and Multilingual DynBench Snapshots

Aleksandr Gashkov[1], Maria Eltsova[1,2] and Andreas Both[1,*]

[1] Web & Software Engineering (WSE) Research Group, Leipzig University of Applied Sciences (HTWK Leipzig), Karl-Liebknecht-Straße 132, 04277 Leipzig, Germany
[2] CBZ München GmbH, Gartenstraße 47/1, Heilbronn, Germany

Abstract

Recent progress in Large Language Models (LLMs) is expected to significantly advance multilingual KGQA, including those for low-resource languages. However, hardly any of the existing KGQA benchmarks covers genuinely low-resource languages. Moreover, benchmarks are static, prone to obsolescence as KGs evolve, and increasingly unreliable for evaluating systems based on LLMs because a model produces the correct output as the requested information was already contained in the training data (the memorization effect). Hence, this paper extends multilingual KGQA benchmarking in two directions. First, we publish the first Udmurt version of QALD-9-plus, created by native-speaker translations for an endangered Uralic language. Second, we extend our *DynBench* framework by a multilinguality module and release two benchmark snapshot collections – *DynQALD* and *DynRuBQ* – generated from the well-known multilingual QALD-9-plus and bilingual RuBQ 2.0 datasets. Unlike prior approaches, DynBench preserves the NL question patterns and structural complexity of the original SPARQL queries while substituting type-compatible entities, yielding new comparable benchmark snapshots. The quality of the generated snapshots was evaluated by native-speaker experts and automatic metrics across six languages (English, German, French, Russian, Ukrainian, and Udmurt) in a Wikidata-based setting. While the error rates for high- and medium-resource languages remain in the low single-digit percentage range, our analysis reveals specific challenges for the agglutinative low-resource Udmurt (e.g., missing labels and morphological errors) – providing informative evidence for the KGQA community. All resources supporting this work, including the framework code and the datasets, are publicly available to facilitate use, reproducibility, and extension.

Keywords

Multilingual KGQA, Benchmark Dataset Generator, Large Language Models, Multilingual Approach, Low-Resource Languages

1. Introduction

Although more than 7,000 languages are spoken worldwide, only a few hundred are utilized on the World Wide Web, where the largest share of the content – 49.6% – is published exclusively

RAGE-KG 2026: The Third International Workshop on Retrieval-Augmented Generation Enabled by Knowledge Graphs, co-located with ISWC 2026, October 25–26, 2026, Bari, Italy

*Corresponding author.
alexander.gashkov@gmail.com (A. Gashkov); maria.eltsova@gmail.com (M. Eltsova); andreas.both@htwk-leipzig.de (A. Both)
0000-0001-6894-2094 (A. Gashkov); 0000-0003-3792-8518 (M. Eltsova); 0000-0002-9177-5463 (A. Both)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-4312/paper40.pdf

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-04

in English[1] while only around 18% of the world population speak English as a native or second language[2]. This disparity is mirrored in Natural Language Processing (NLP), information retrieval (IR), and question answering (QA), where the majority of research has focused on a small handful of high-resource languages (e.g., [1, 2, 3]) while low-resource languages often lack the extensive data available to dominant ones.

In recent years, LLMs have become the core engine for a wide spectrum of natural-language (NL) processing tasks, notably Question Answering over Knowledge Graphs (KGQA), where the general goal is to translate a user's NL question into a formal SPARQL query that retrieves the correct answer from a knowledge graph (KG); e.g., [4, 5, 6, 7, 8, 9] presented promising LLM-based approaches that require little hand-crafted logic. However, their application in multilingual scenarios is still limited, especially in extremely low-resource languages [10, 11, 12].

High-quality benchmark datasets are a cornerstone of empirical progress in KGQA. However, existing KGQA resources suffer from several systemic issues: multilingual (especially low-resource) and up-to-date benchmarks are scarce [13, 14, 15]; benchmark records become outdated while KGs evolve; and most benchmarks present a single, fixed distribution of SPARQL query shapes. Moreover, due to the fast integration of published datasets into the training data of LLMs, the trustworthiness of KGQA evaluations is doubtful as they often are heavily impacted by the memorization capabilities of the LLMs (cf. [16]). These challenges motivate a generalized solution that can generate fresh, controllable KGQA evaluation datasets on demand while preserving the linguistic naturalness of the questions and the structural integrity of the underlying SPARQL queries. As one possible solution, we presented DynBench in [17] and its improved version in [18]. In this paper, we evaluate its new version (extended by a multilinguality module) on six languages, including the low-resource Udmurt.

The following research questions are set in this work:

RQ1 Is it possible to generate a high-quality benchmark based on an existing one, such that the automatically produced question-query pairs retain the linguistic fluency and semantic correctness of the source data – not only for *high-resource* but also for *low-resource* languages?

RQ2 What additional challenges will the DynBench face when working with *low-resource* languages?

To fill the gaps mentioned above, we make the following *unique contributions*:

1. For the first time, we publish an extended full version of QALD-9-plus for another low-resource language – Udmurt [19], an endangered indigenous Uralic language.
2. We improve DynBench[3] [17, 18] – a largely automated pipeline that transforms an existing KGQA benchmark into a family of new benchmark snapshots with adjustable entity-selection difficulty – by adding multilinguality.
3. We present here two released benchmark dataset snapshots[4]: multilingual DynQALD and bilingual DynRuBQ, generated by DynBench as fixed, citable snapshots of a dynamic generation process.

[1] https://w3techs.com/technologies/overview/content_language
[2] https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/#statisticContainer
[3] https://github.com/WSE-research/DynBench-Backend
[4] https://zenodo.org/records/20073039

4. We provide validation evidence combining automatic filtering and human expert assessment.
5. We describe some challenges the KGQA community might face when developing digital solutions for low-resource languages.

All associated resources, including the multilingual benchmark dataset collections[4] and framework code[3], are made publicly available to facilitate future research.

The resource described in this paper, therefore, consists of: (i) the Udmurt extension of QALD-9-plus, (ii) the DynBench generation pipeline, (iii) the two released benchmark collections, (iv) metadata and validation results for the generated question-query pairs, and (v) Docker-based execution support. We call the resource dynamic because the generator can create new benchmark snapshots on demand; the datasets released with this paper are fixed, citable snapshots of this process.

Delta to our previous work. DynBench was introduced in [17], extended by an automated validation mechanism in [18], and complemented by a web-based demonstrator [20]. The DynBench pipeline is presented in Figure 1. After extracting the resources, a random entity from a query is selected, and a set of equivalent entities from a KG is identified. A unique replacement entity is then chosen from this equivalency class and its label is retrieved. Simultaneously, an LLM rewrites the original NL question according to a prompt template using the retrieved label, which completes the generation of a new question-query pair. After that, DynBench uses optimized text-based metrics to compare the original question with the generated one. If the transformation proves unsuitable, the method repeats the sampling process until the desired result is achieved or the maximum number of repetitions is reached. In comparison, this paper *newly* contributes (i) the multilingual module of DynBench, (ii) the released multilingual benchmark snapshots DynQALD and DynRuBQ, and (iii) a multilingual human evaluation including the low-resource Udmurt.

The paper is organized as follows: Section 2 surveys related work, Section 3 details the datasets and DynBench, Section 4 presents the experiments, Section 5 discusses limitations and results, and Section 6 concludes the paper.

2. Related Work

2.1. Low-resource Languages

In the research community, there is limited consensus on what exactly qualifies as a "low-resource language" [1, 2]: e.g., Turkish [21], Vietnamese [22], and Thai [23] are called low-resource by some researchers while others classify them as medium-resourced or higher [1, 12]. To avoid this controversy, we rely on the existing taxonomies: Joshi et al. [1] categorize languages into six groups by data availability (using Wikipedia pages as a proxy for unlabeled data), Nigatu et al. [2] define "low-resource" along four aspects (sociopolitical aspects, resources, artifacts, and agency), and Mbengue et al. [3] derive a formal definition of low-, medium-, and high-resource languages for the Semantic Web (and, hence, KGQA) from the language distribution across DBpedia, BabelNet, and Wikidata.

Table 1
Comparative characteristics of the languages (ISO 639 codes). Status and number of speakers were obtained from Ethnologue (https://www.ethnologue.com); "U-Data" – number of Wikipedia articles (unlabeled data); "L-Data" – number of Wikidata labels; "Usage" – content-language share of websites according to https://w3techs.com/technologies/overview/content_language.

Language	Status	Speakers	U-Data	L-Data	Usage, in %
en	International, Institutional	1.493B	7.214M	90.446M	49.6
de	National, Institutional	180M	3.139M	17.094M	5.9
fr	International, Institutional	333.5M	2.771M	17.972M	4.5
ru	International, Institutional	210.3M	2.111M	9.273M	3.4
uk	National, Institutional	35M	1.429M	6.338M	0.7
udm	Indigenous, Endangered	270K	5,915	0.195M	not ranked

Table 1 briefly characterizes the investigated languages describing their status, number of speakers, available data, and the usage statistics of content languages for websites.

Summing up the data from Table 1 and estimating them according to the taxonomies and classification above, we qualify English, German, and French as high-resource languages, Russian and Ukrainian as medium-resource ones, and Udmurt as a low-resource one.

2.2. Multilingual Benchmarks for KGQA

The training and fair evaluation of KGQA systems requires benchmarking datasets; however, most existing KGQA datasets lack the necessary depth of alignment, domain coverage, or multilingual support [14, 24, 13, 15, 25]. Table 2 demonstrates an overview of the multilingual benchmarks for Wikidata or DBpedia; in addition, EventQA [26] provides 1,000 event-centric questions in three languages over EventKG [27]. Recently presented datasets (e.g., CR-LT-KGQA [28], Spider4SPARQL [29], QueryBridge [30], CompoST [31], WQLD [32]) utilize only English.

Table 2
Overview of the multilingual KGQA benchmarks over Wikidata and DBpedia since 2018

Name	Year	Number of questions	SPARQL queries	Languages	Comment
QALD-9 [33]	2018	558	Yes	en, de, ru, pt, hi, fa, it, fr, ro, es, nl	Low translation quality
RuBQ 2.0 [34]	2021	2,910	Yes	ru, en	Machine-translated questions, unanswerable questions
MCWQ [35]	2022	124,187	Yes	en, he, kn, zh	Synthetic NL questions, translations obtained with MT
Mintaka [36]	2022	20,000	No	en, ar, de, ja, hi, pt, es, it, fr	Named entities are annotated in the English questions
QALD-9-plus [24]	2022	558 (507)	Yes	en, de, fr, ru, uk, lt, be, ba, hy	Not all questions are covered for Wikidata
QALD-10 [37]	2024	909	Yes	en, de, zh, ru	Manual transformation of SPARQL queries from DBpedia to Wikidata

Summing up, there is a deficit of the new benchmark datasets for multilingual KGQA (with SPARQL queries) including low-resource languages (only QALD-9, QALD-9-plus, and MCWQ have some of them). Another important issue with existing and newly emerging datasets is that they can become almost worthless or obsolete after a few exploitations due to the memorization effect and data contamination (e.g., [16, 38, 6]). Recently appeared frameworks for generating SPARQL queries by LLMs (e.g., [5, 39, 40, 7, 41]) or for creating new benchmarks [42, 43, 6, 41] may contribute to solving this problem.

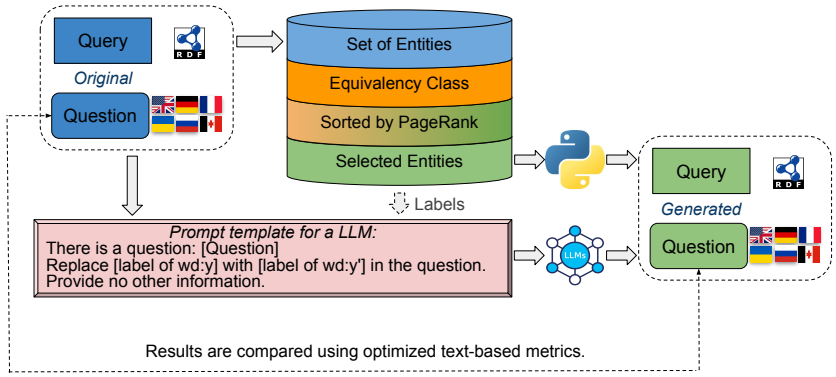


Figure 1: Overview of the DynBench pipeline with the multilingual prompt template; the dashed arrow denotes the comparison of the original and the generated question by text-based metrics for the automatic validation

However, to the best of our knowledge, most of these approaches are monolingual, and, generating new KGQA datasets directly from KGs or by LLMs, they typically produce one-off datasets whose structural and linguistic properties cannot be reproduced or controlled. Hence, two datasets generated by the same approach may differ substantially in difficulty or lexical complexity, which makes them non-comparable for systematic evaluation. Despite the similar name, DynBench is unrelated to Dynabench [44], a platform for human-in-the-loop *adversarial* data collection. We aim to fill this gap in the KGQA community by providing the first Udmurt dataset for KGQA as well as the unique DynBench framework and two *multilingual benchmark dataset collections* – DynQALD and DynRuBQ – generated by DynBench.

3. Material and Method

We introduce here the Udmurt extension of QALD-9-plus, the multilingual DynBench, and the two generated datasets DynQALD and DynRuBQ (their automatic validation and human evaluation follow in Section 4).

3.1. Source Dataset Requirements

For generating a benchmark dataset collection, an existing benchmark dataset (i.e., a source dataset) is required; our previous research showed that the quality of a generated dataset strongly depends on the quality of the source dataset [17]. Therefore, the source multilingual datasets should correspond to the following conditions: (1) share a KG because it is not always possible to interlink two or more KGs, (2) availability of ground truth SPARQL queries, (3) intersection in at least one language besides English, (4) different ways of creation, and (5) good quality of NL question and SPARQL query. Following these conditions, we preferred QALD-9-plus [24] to two other QALD benchmarks [33, 37] due to included medium-resource languages (e.g., Ukrainian) and also chose RuBQ 2.0 [34].

RuBQ 2.0 extends RuBQ 1.0 [45], the first Russian dataset for QA over Wikidata, and consists of 2,910 questions of varying complexity created in a semi-automatic way, their machine trans-

lations into English without any post-editing, and annotated SPARQL queries. Its unanswerable questions are not used because the corresponding records do not contain a SPARQL query needed to produce new questions.

QALD-9-plus is multilingual and contains 558 questions (507 of which provide a Wikidata SPARQL query; 93.9% of them are productive, i.e., executable with a non-empty result), providing for each question a textual representation in multiple languages, the corresponding SPARQL query, the answer entity URI, and the answer type. Its well-formed NL questions were human-translated from English to German, French, Russian, Armenian, Belarusian, Lithuanian, Bashkir, and Ukrainian.

3.2. Udmurt QALD-9-plus

To answer our RQ1 and RQ2, we extended the existing QALD-9-plus with another low-resource language – Udmurt – by human translations of NL questions done by native speakers following the same methodology as described in [24] and previously applied for Spanish in QALD-9-ES [46]. The Udmurt extension covers all of the 558 questions and corresponding SPARQL queries. The NL questions were translated by 3 native speakers and cross-checked by 2 professional linguists (also native speakers of Udmurt). Disagreements were resolved with the following procedure. While cross-checking, each annotator evaluated each translated NL question as follows: 3 – the best variant of translation, 2 – good variant, 1 – the worst variant. The variant with maximal score was chosen. In the case of equal scores, the variant marked by the annotator 1 (certified specialist in Udmurt philology) with "3" was selected.

Udmurt, which belongs to the Uralic language family, is an endangered indigenous language of the Russian Federation written using a modified version of the Russian Cyrillic alphabet; unlike most other languages from QALD-9-plus (with Bashkir, and partly Armenian, as exceptions), it is agglutinative, which could affect the LLMs' behavior and, therefore, the results.

3.3. Dynamic Dataset Generation with DynBench

We introduced the DynBench workflow in our previous studies [17, 18] and modified it here by adding a new feature – multilinguality (cf. Figure 1). DynBench can generate an arbitrary number of new multilingual benchmark snapshots (therefore, named *a dataset collection*) while preserving the NL question patterns and structural complexity of the original SPARQL queries of the given KGQA dataset.

Multilinguality module. The multilinguality module extends the monolingual pipeline [17, 18] in three places. First, the target language is a parameter: the label of the replacement entity is retrieved from Wikidata in this language, with a fallback to the related language, if available (Russian for Udmurt, cf. Section 5) and, finally to the label marked as "default for all languages". Second, the English prompt template (cf. Figure 1) receives the source question and both labels in the target language (or related/common label, see the previous step); it additionally names the target language. Third, the automatic validation (Section 4) runs with the LLM used in step two and Levenshtein threshold (maximum of four characters) as in our previous work. To produce a new benchmark record, we transform an existing record into a new one. Transformation

means the synchronous change of a question and a query (a question-query pair) in the way that preserves syntactic and semantic correctness of both. To produce semantically consistent question-query pairs, the substituted entities are drawn from a *type-compatible candidate set*, i.e., entities sharing `instance of` (`wdt:P31`[5]) and/or `subclass of` (`wdt:P279`) properties, and the generated queries are validated automatically without human intervention.

While transforming a question-query pair (cf. Figure 1), we assume that its difficulty consists of two main parts: the structural complexity of the SPARQL query – which DynBench keeps unchanged – and the individual difficulty of each resource. As a proxy for the latter, we use the entity's PageRank [47][6]: questions containing prominent, high-PageRank entities are typically easier w.r.t. entity recognition and linking than questions containing low-PageRank entities. There are four entity-selection modes in DynBench [20]: *Easy* (the selected new entity has the highest possible PageRank), *Similar* (a PageRank as close as possible to the one of the entity in the source question-query pair, i.e., the new entity has a similar importance in the given KG as the original resource), *Hard* (the lowest possible PageRank), and *Random* (entities are probed in random order). In this study, we restricted our investigation to the "Similar" mode because it is nearly impossible to vary the complexity for Udmurt due to the lack of labels.

For generating the NL representation of DynQALD and DynRuBQ from the source datasets QALD-9-plus and RuBQ 2.0, we employed two LLMs: Gemma-3-27b [48] (later referred to as Gemma 3) and Mistral-Small:24b[7] (later referred to as mistral-small). The models were selected because they are compact and can be run locally in most cases; however, DynBench allows the users to choose the LLM by themselves [20]. For all languages besides Udmurt, we used the same prompt template presented in Figure 1. For Udmurt, we had to add to the prompt additional information about the language because LLMs could not either identify the source language or generate a correct and fluent NL question in Udmurt during the transformation. In this case, LLMs struggled with the specific morphology of this agglutinative language, either replacing the entity with the Russian / Universal one or adding no suffixes (completely ignoring the grammar of the language).

3.4. Exemplar Generated Benchmark Datasets: DynQALD and DynRuBQ

The generated *DynQALD* dataset has 1791 unique questions in total, based on 356 unique new entities. The average length of the questions is 7.4 words in English, 5.9 words in Russian, 7.0 words in German, 7.6 words in French, and 5.8 words in Ukrainian. The lack of Udmurt labels resulted in the low number of question-query pairs in the Udmurt version of DynQALD. Gemma 3 generated only 27 Udmurt question-query pairs and mistral-small 28.

The generated *DynRuBQ* dataset has 4011 unique questions in total, based on 1773 unique new entities; the average question length is 8.8 words in English and 6.5 words in Russian, and, at the moment, DynRuBQ does not contain the unanswerable questions of its source.

As mentioned above, in both datasets, the "Similar" entity-selection mode preserves the PageRank-based entity-prominence level of the respective source dataset.

[5] http://www.wikidata.org/prop/direct/P31
[6] https://danker.s3.amazonaws.com/index.html
[7] https://mistral.ai/news/mistral-small-3

3.5. Availability, Sustainability and Reusability

The Python script to generate new question-query pairs and the datasets are available in our GitHub repository[8] under the MIT license; the released DynQALD and DynRuBQ datasets are fixed, citable snapshots of the dynamic generation process, kept in the JSON format to enable the reuse of existing evaluation systems (e.g., GERBIL [49]). The technical sustainability is ensured through the provided Docker container; the WSE Research Group maintains the resource, which remains fully open to community contributions. We plan to steadily improve DynBench (e.g., decreasing the procedure errors for more complex questions, building in cross-lingual transformation [20], and enhancing the automatic validation) and to extend the generated datasets with more verbalization examples and languages.

4. Experiments

To ensure dataset quality and to support reuse, we carried out two evaluation steps – an automatic validation of all generated question-query pairs and a human evaluation of representative samples.

4.1. Quality Evaluation of the Exemplar Benchmark Datasets

Here we apply the automatic validation mechanism of our previous work [18, 20], which supports semantic and syntactic integrity through back-transformation and metric-based validation, and complement it with a human evaluation.

Automatic evaluation. To conduct the evaluation precisely and accurately, we used the system of different metrics: number of sentences, Levenshtein distance, and error frequency due to their reliability, stability, and strong alignment with expert judgment regarding the transformation accuracy [18]. We first filtered the dataset using number of sentences metric to remove extreme errors (e.g., duplicate prompts or non-transformed queries), eliminating approximately 1% of the pairs.

In the next step, we perform a "back-transformation" process where the generated question-query pair is used as an input to our method and then compare the original NL question with the retranslated one, i.e.,:

$$(\text{question}, \text{query}) \simeq \textit{transform}(\textit{transform}((\text{question}, \text{query})))$$

The complete validation procedure, including the fuzzy replacement operation, the back-transformation process, the investigated text similarity metrics, and the derivation of the validation criterion, is described in detail in [20]. As demonstrated in [20], the Levenshtein distance provides the most accurate and stable criterion for distinguishing correct and erroneous transformations. Therefore, we use the Levenshtein distance as the automatic validation metric in this work. Applying this validation step, we identified approximately 30% of newly generated question–query pairs as erroneous and excluded them from the resulting datasets. Consistent with our previous study, transformations with a Levenshtein distance not exceeding 4 are considered valid.

[8] https://github.com/WSE-research/DynBench-Backend

Table 3
Human evaluation results: frequency of LLM errors (in % of the evaluated pairs); "–" denotes configurations that were not evaluated. Languages given as ISO 639 codes.

LLM	Source dataset	EN	RU	DE	FR	UK	UDM
Gemma 3	QALD-9-plus	0	1	1	3	4	32.1
Gemma 3	RuBQ 2.0	0	1	–	–	–	–
mistral-small	QALD-9-plus	0	0	2	3	4	37.5
mistral-small	RuBQ 2.0	0	0	–	–	–	–

Note that we did not evaluate how many of the removed pairs are false negatives or true negatives; all remaining question-query pairs were defined as correct and built the new dataset.

Human evaluation. The human evaluation protocol follows our previous work [20]. We selected 50 random question-query pairs from DynQALD and 50 random ones from DynRuBQ for each of the two LLMs, i.e., 200 in total for English and Russian (which are presented in both original benchmarks) and 100 for German, French, and Ukrainian and all available pairs (30) for Udmurt. Each record was independently assessed by at least two native speakers of the corresponding language. Experts verified the correctness of both the natural-language question and the corresponding SPARQL query, assigning errors to one of three categories: (EC1) errors in the source benchmark, (EC2) procedure errors, and (EC3) LLM errors [18]. Incorrect records from the source datasets (9 such records were identified for RuBQ 2.0 and 2 for QALD-9-plus) and their generated counterparts were excluded from the final analysis, and the frequency of correct question–query pairs and each error category was subsequently computed. Further improvement of automated evaluation and the approach itself resulted in minimization of procedure errors: 0 errors in RuBQ 2.0 and 1 error in QALD-9-plus for all languages besides Udmurt.

The frequency of the LLM errors discovered during the human evaluation is presented in Table 3. The detailed error analysis indicates that errors in the original benchmarks occurred for the same question-query pairs in all languages (QALD-9-plus – 2, RuBQ 2.0 – 9), and there were only 2 cases of procedure errors (QALD-9-plus) for all languages besides Udmurt, where the sample is small due to the missing labels. The LLM errors, however, vary from about 1% (English) up to 33–42% (Udmurt, depending on the LLM). For high- and medium-resource languages, these errors are minor grammatical inaccuracies that do not complicate the understanding (e.g., incorrect adjective declension or noun gender in German, Russian, and Ukrainian, or Russian-influenced prepositions in Ukrainian). The LLM errors for Udmurt are more severe: LLMs used Russian or English (universal) labels without adding the corresponding suffixes (ignoring the grammar), or the necessary transliteration was missing. While both small models were able to produce usable Udmurt questions for the majority of pairs, the error rate remains an order of magnitude above the one of the other evaluated languages – this gap is itself an informative answer to RQ2 and quantifies the additional challenges that LLM-based generation faces for agglutinative low-resource languages.

5. Limitations and Discussion

Some limitations call for further improvement. First, we did not use other open-source and commercial LLMs for generating new question-query pairs because the quality of both exploited LLMs is high enough to achieve our goal. However, we successfully tested possibilities to exploit other LLMs for this purpose [20]. Second, some question-query pairs could be transformed only by the simultaneous substitution of two (or more) entities to become semantically correct, which requires the further development of DynBench. Third, the approach was evaluated on the Wikidata KG only; since the current implementation relies on Wikidata labels, `instance_of`/`subclass_of` relations, and PageRank values, its portability to other KGs remains future work. Fourth, we included only one low-resource language in our experiments because the method still requires human efforts, especially in the case of low-resource languages. Moreover, the languages selected for the experiments represent only two language families; the three high-resource ones are related, the medium-resource Russian and Ukrainian are very closely related, and the low-resource Udmurt has been influenced by Russian. Fifth, the Levenshtein threshold of the automatic validation was calibrated on English data [18] and applied unchanged to all languages; for the agglutinative Udmurt, where case suffixes legitimately change the entity string, this transfer may let erroneous pairs pass or reject correct ones – a threat we could not quantify on the small Udmurt sample.

Compared to [17], we substantially reduced the rate of the procedure errors and LLMs' cross errors for high- and medium-resource languages, which indicates substantial improvements of DynBench. The remaining, in most cases minor, grammatical inaccuracies vary by language; possible reasons are that LLMs handle fusional languages (Ukrainian, Russian) worse than analytical English, and that Ukrainian and French have less available data on the Web.

Challenges with Udmurt. The statistics provided here are based on a non-representative sample (due to the small amount of data); however, they allow us to outline some trends that need further study.

The first (and obvious) challenge in KGQA the researchers face when processing *any* low-resource language is lack of labeled data. The possible solution here seems to be using labels marked as "default for all languages" or labels from a related language. In the case of Udmurt, both solutions are non-optimal since the related languages are also low-resource and the most universal labels use another script (e.g., Latin), so the transliteration or transcription is required. In our experiments, Gemma 3 struggled with this task (15.8% of errors in transformed questions) while mistral-small demonstrated no such cases. The more effective solution for Udmurt was to use the Russian labels (many words were loaned from Russian); however, both LLMs often used Russian label ignoring the Udmurt grammar (21.1% for Gemma 3 and 30% for mistral-small) despite the prompt instructions. The further investigation with exploiting other LLMs is required to recognize the reason of this behavior: are the models too small, or is there not enough training data for solving this task? Another common mistake for mistral-small (15% of errors) was generating the same question (sometimes with paraphrasing) by transformation process. The share of hallucinations is also high: Gemma 3 – 10.53%, mistral-small – 10%.

According to our results, the following challenges should be considered when working with low-resource languages: 1) lack of digital text resources that fit the mainstream way of training

models; 2) very special grammar and orthography of each language; 3) lack of human resources: native speakers, linguistic experts, NLP researchers etc. (comparable with [2]); 4) non-Latin script. Another risk can be the data contamination due to the low-quality and erroneous machine translation into a low-resource language.

Ethical considerations. Generating text in an endangered language raises specific responsibilities: erroneous synthetic Udmurt (e.g., entity labels with missing suffixes or transliterations) could pollute the small digital footprint of the language if it is reused, e.g., for training language models; we, therefore, recommend excluding the generated Udmurt questions (part of DynQALD) from training corpora.

6. Conclusion and Future Work

In this paper, we extended multilingual KGQA benchmarking in two directions: we published the first Udmurt version of QALD-9-plus, translated and cross-checked by native speakers (the first KGQA benchmark for an endangered Uralic language), and we presented multilingual DynBench, a pipeline for the dynamic creation of multilingual benchmark snapshots preserving the NL question patterns and SPARQL structure of a given KGQA benchmark, and used it to generate DynQALD and DynRuBQ from the well-known QALD-9-plus and RuBQ 2.0– motivated by the need for reproducible and trustworthy KGQA benchmarking in the era of LLM training-data contamination. Compared to our earlier work (the monolingual pipeline [17], its automatic validation [18], and a demonstrator [20]), this paper newly contributes the multilinguality module, the multilingual snapshots, and the first multilingual human evaluation including a low-resource language.

Answering RQ1, the human and automatic evaluations show that high-quality benchmark snapshots can be generated from an existing benchmark for high- and medium-resource languages (error rates of 1–6%), while for the low-resource Udmurt the quality is substantially lower. Answering RQ2, the main additional challenges for low-resource languages are missing KG labels, which limit the number of generatable pairs, and morphological errors of LLMs (e.g., ignored agglutinative suffixes and missing transliteration). Other challenges listed in this paper may outline not only the future work in the field of NLP, linguistics, and Semantic Web but also lead to trustworthy interdisciplinary research of low-resource languages.

The error analysis indicates that DynBench is repeatable and effective for the evaluated high- and medium-resource languages, while Udmurt reveals the remaining challenges; compared to our previous study [17], the procedure errors dropped from up to 20% there down to at most 2% here. Future work targets further multilingual expansion via cross-lingual LLM capabilities, refined multi-entity substitution, enhanced automatic validation, and a KG-agnostic pipeline. A promising direction is the extension of DynBench to other low-resource fusional languages related to the evaluated ones and to medium- or high-resource agglutinative languages in order to determine the nature of the LLM errors: Were these errors caused by a deficit of data, by interference with related language(s), by special grammar, or even by contaminated synthetically generated data in the LLMs' training corpora?

Published resources Code: https://github.com/WSE-research/DynBench-Backend; DynQALD/DynRuBQ snapshots: https://doi.org/10.5281/zenodo.20073039; Udmurt QALD-9-plus: https://github.com/WSE-research/QALD_9_plus_Udmurt; expert judgments: https://doi.org/10.6084/m9.figshare.33102470.

Acknowledgments We sincerely thank all who supported the human evaluation and translation work, in particular: Susanne Both, Olga Botalova, Mariia Dementieva, Ekaterina Zakharova, Iryna Zohal, Evelina Volska, and Renata Volska.

Declaration on Generative AI The authors used Grammarly, ChatGPT, Claude Opus 5, and Gemma 3/4 only for grammar and spelling checks and take full responsibility.

References

[1] P. Joshi, S. Santy, A. Budhiraja, K. Bali, M. Choudhury, The State and Fate of Linguistic Diversity and Inclusion in the NLP World, in: Proceedings of the 58th annual meeting of the association for computational linguistics, 2020, pp. 6282–6293.

[2] H. H. Nigatu, A. L. Tonja, B. Rosman, T. Solorio, M. Choudhury, The Zeno's Paradox of 'Low-Resource' Languages, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 17753–17774.

[3] N.-E. Mbengue, P. Monnin, M. Couceiro, F. Gandon, Which Are the Low-Resource Languages of the Semantic Web?, in: ESWC 2026-23rd European Semantic Web Conference, 2026.

[4] L.-P. Meyer, J. Frey, F. Brei, D. Heim, S. Gründer-Fahrer, S. Todorovikj, C. Stadler, M. Schröder, N. Arndt, M. Martin, Evaluating Large Language Models for RDF Knowledge Graph-Related Tasks—The LLM-KG-Bench-Framework 3, Semantic Web 17 (2026).

[5] L. Kovriguina, R. Teucher, D. Radyush, D. Mouromtsev, N. Keshan, S. Neumaier, A. Gentile, S. Vahdati, SPARQLGEN: One-Shot Prompt-based Approach for SPARQL Query Generation., in: SEMANTiCS (Posters & Demos), 2023.

[6] P. P. S. Dammu, H. Naidu, C. Shah, Dynamic-KGQA: A Scalable Framework for Generating Adaptive Question Answering Datasets, arXiv preprint arXiv:2503.05049 (2025).

[7] T. Schwabe, L. Siebel, P. Valach, M. Acosta, Q-NL Verifier: Leveraging Synthetic Data for Robust Knowledge Graph Question Answering, arXiv preprint arXiv:2503.01385 (2025).

[8] A. Perevalov, A. Gashkov, M. Eltsova, A. Both, SPARQL Query Candidate Filtering for Improving the Quality of Multilingual Question Answering over Knowledge Graphs using Language Models, Journal of Web Engineering (2025) 563–592.

[9] A. Gashkov, M. Eltsova, A. Perevalov, A. Both, Instruction-Tuned Language Models as Judges for SPARQL Query Correctness in Knowledge Graph Question Answering, in: 4TH International Workshop on Knowledge Graph Generation from Text (TEXT2KG). Co-located with the Extended Semantic Web Conference (ESWC 2025), Portorož, Slovenia, 2025.

[10] K. Huang, F. Mo, X. Zhang, H. Li, Y. Li, Y. Zhang, W. Yi, Y. Mao, J. Liu, Y. Xu, et al., A Survey on Large Language Models with Multilingualism: Recent Advances and New Frontiers, Artificial Intelligence Review (2026).

[11] V. D. Lai, N. Ngo, A. P. B. Veyseh, H. Mẫn, F. Dernoncourt, T. Bui, T. H. Nguyen, ChatGPT beyond English: Towards a Comprehensive Evaluation of Large Language Models in Multilingual Learning, in: Findings of the association for computational linguistics: EMNLP 2023, 2023, pp. 13171–13189.
[12] B. Pecher, J. Cegin, R. Belanec, I. Srba, J. Simko, M. Bielikova, Better as Generators Than Classifiers: Leveraging LLMs and Synthetic Data for Low-Resource Multilingual Classification, arXiv preprint arXiv:2601.16278 (2026).
[13] A. Perevalov, A. Both, A.-C. N. Ngomo, Multilingual Question Answering Systems for Knowledge Graphs—a Survey, Semantic Web 15 (2024) 2089–2124. URL: https://www.semantic-web-journal.net/system/files/swj3530.pdf.
[14] L. Jiang, R. Usbeck, Knowledge graph question answering datasets and their generalizability: are they enough for future research?, in: Proceedings of the 45th International ACM SIGIR conference on research and development in information retrieval, 2022, pp. 3209–3218.
[15] E. Loginova, S. Varanasi, G. Neumann, Towards end-to-end multilingual question answering, Inform. Systems Frontiers 23 (2021) 227–241.
[16] A. Gashkov, A. Perevalov, M. Eltsova, A. Both, SPARQL Query Generation with LLMs: Measuring the Impact of Training Data Memorization and Knowledge Injection, in: ICWE, 2025, pp. 177–192.
[17] A. Gashkov, A. Perevalov, M. Eltsova, A. Both, Towards Dynamically Generated KGQA Benchmark Datasets for Memorization-Resistant Evaluations, in: KGSWC, Springer Nature, 2026.
[18] A. Gashkov, A. Perevalov, M. Eltsova, A. Both, Automated Robust Dynamic Generation of KGQA Benchmark Datasets, in: 2026 International Conference on Semantic Computing (ICSC), 2026, pp. 24–31. doi:10.1109/ICSC67292.2026.00010.
[19] S. Edygarova, Udmurt, in: M. Bakró-Nagy, J. Laakso, E. Skribnik (Eds.), The Oxford Guide to the Uralic languages, Oxford University Press, United Kingdom, 2022, pp. 507–522.
[20] A. Gashkov, M. Eltsova, A. Both, DynBench Generator: A Web-Based Platform for On-Demand KGQA Benchmark Creation, in: International Conference on Web Engineering, Springer, 2026, pp. 291–295.
[21] E. Tosun, M. E. Buldur, Ö. Ezerceli, M. ElHussieni, A Hybrid Protocol for Large-Scale Semantic Dataset Generation in Low-Resource Languages: The Turkish Semantic Relations Corpus, arXiv preprint arXiv:2601.13253 (2026).
[22] L. Nguyen, H. Bui, ViTKGQA: A Vietnamese Temporal Knowledge Graph Question Answering Method, in: Asian Conference on Intelligent Information and Database Systems, Springer, 2026, pp. 401–416.
[23] X. Liang, Y.-M. J. Khaw, S.-Y. Liew, T.-P. Tan, D. Qin, Towards Low-Resource Languages Machine Translation: A Language-Specific Fine-Tuning with LoRA for Specialized Large Language Models, IEEE Access (2025).
[24] 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: ICSC, 2022, pp. 229–234.
[25] A. Perevalov, A. Gashkov, M. Eltsova, A. Both, Language Models as SPARQL Query Filtering for Improving the Quality of Multilingual Question Answering over Knowledge

Graphs, in: Web Engineering, 2024, pp. 3–18.
[26] T. Souza Costa, S. Gottschalk, E. Demidova, Event-QA: A dataset for event-centric question answering over knowledge graphs, in: Proceedings of the 29th ACM international conference on information & knowledge management, 2020, pp. 3157–3164.
[27] S. Gottschalk, E. Demidova, EventKG–the hub of event knowledge on the web–and biographical timeline generation, Semantic Web 10 (2019) 1039–1070.
[28] W. Guo, A. Toroghi, S. Sanner, CR-LT-KGQA: A Knowledge Graph Question Answering Dataset Requiring Commonsense Reasoning and Long-Tail Knowledge, arXiv preprint arXiv:2403.01395 (2024).
[29] C. Kosten, P. Cudré-Mauroux, K. Stockinger, Spider4SPARQL: A Complex Benchmark for Evaluating Knowledge Graph Question Answering Systems, in: 2023 IEEE international conference on big data (BigData), IEEE, 2023, pp. 5272–5281.
[30] A. Orogat, A. El-Roby, Querybridge: One Million Annotated Questions with SPARQL Queries-Dataset for Question Answering over Knowledge Graphs, in: Proceedings of the 34th ACM International Conference on Information and Knowledge Management, 2025, pp. 6503–6507.
[31] D. M. Schmidt, R. Schubert, P. Cimiano, CompoST: A Benchmark for Analyzing the Ability of LLMs to Compositionally Interpret Questions in a QALD Setting, in: International Semantic Web Conference, Springer, 2025, pp. 3–22.
[32] S. Walter, H. Bast, The Wikidata Query Logs Dataset, in: Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '26, Association for Computing Machinery, New York, NY, USA, 2026, p. 3442–3447. URL: https://doi.org/10.1145/3805712.3808612. doi:10.1145/3805712.3808612.
[33] R. Usbeck, R. Gusmita, M. Saleem, A.-C. Ngonga Ngomo, 9th Challenge on Question Answering over Linked Data (QALD-9), in: Semdeep/NLIWoD@ISWC, 2018.
[34] I. Rybin, V. Korablinov, P. Efimov, P. Braslavski, RuBQ 2.0: An innovated Russian question answering dataset, in: ESWC, Springer, 2021, pp. 532–547.
[35] R. Cui, R. Aralikatte, H. Lent, D. Hershcovich, Compositional Generalization in Multilingual Semantic Parsing over Wikidata, Transactions of the Association for Computational Linguistics 10 (2022) 937–955.
[36] P. Sen, A. F. Aji, A. Saffari, Mintaka: A Complex, Natural, and Multilingual Dataset for End-to-End Question Answering, in: COLING, 2022.
[37] R. Usbeck, X. Yan, A. Perevalov, L. Jiang, J. Schulz, A. Kraft, C. Möller, J. Huang, J. Reineke, A.-C. Ngonga Ngomo, et al., QALD-10–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.
[38] N. Carlini, D. Ippolito, M. Jagielski, K. Lee, F. Tramèr, C. Zhang, Quantifying Memorization Across Neural Language Models, in: ICLR, 2023.
[39] M. Lan, Y. Xia, G. Zhou, N. Huang, Z. Li, H. Wu, LLM4QA: Leveraging Large Language Model for Efficient Knowledge Graph Reasoning with SPARQL Query, JAIT 15 (2024).
[40] F. Brei, L. Bühmann, J. Frey, D. Gerber, L.-P. Meyer, C. Stadler, K. Bulert, ARUQULA - an LLM based Text2SPARQL approach using ReAct and knowledge graph exploration utilities, in: Proceedings of Text2SPARQL Workshop @ ESWC 2025, 2025.
[41] R. A. Jonker, B. M. R. d. A. Martins, S. Matos, BioGraphletQA: Knowledge-Anchored

Generation of Complex QA Datasets, in: European Conference on Information Retrieval, Springer, 2026, pp. 610–625.
[42] D. Bhandiwad, P. Gattogi, A. Kangen, M. Basaldella, S. Ferré, S. Vahdati, J. Lehmann, Bridging Language Models and Knowledge Graphs with Controlled Natural Languages, SSRN 5009450 (2024).
[43] A. Orogat, A. El-Roby, Maestro: Automatic generation of comprehensive benchmarks for question answering over knowledge graphs, Proceedings of the ACM on Management of Data 1 (2023) 1–24.
[44] D. Kiela, M. Bartolo, Y. Nie, D. Kaushik, A. Geiger, Z. Wu, B. Vidgen, G. Prasad, A. Singh, P. Ringshia, T. Thrush, S. Riedel, Z. Waseem, P. Stenetorp, R. Jia, M. Bansal, C. Potts, A. Williams, Dynabench: Rethinking Benchmarking in NLP, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 4110–4124.
[45] V. Korablinov, P. Braslavski, RuBQ: A Russian Dataset for Question Answering over Wikidata, in: The Semantic Web – ISWC 2020, Springer, 2020, pp. 97–110.
[46] J. Soruco, D. Collarana, A. Both, R. Usbeck, QALD-9-ES: A Spanish Dataset for Question Answering Systems, Studies on the Semantic Web, IOS Press BV, Netherlands, 2023, pp. 38–52.
[47] D. Diefenbach, A. Thalhammer, Pagerank and generic entity summarization for rdf knowledge bases, in: European Semantic Web Conference, Springer, 2018, pp. 145–160.
[48] G. Team, A. Kamath, J. Ferret, S. Pathak, N. Vieillard, R. Merhej, S. Perrin, T. Matejovicova, A. Ramé, M. Rivière, et al., Gemma 3 technical report, arXiv preprint arXiv:2503.19786 (2025).
[49] R. Usbeck, M. Röder, A.-C. Ngonga Ngomo, C. Baron, A. Both, M. Brümmer, D. Ceccarelli, M. Cornolti, D. Cherix, B. Eickmann, P. Ferragina, C. Lemke, A. Moro, R. Navigli, F. Piccinno, G. Rizzo, H. Sack, R. Speck, R. Troncy, J. Waitelonis, L. Wesemann, GERBIL – general entity annotation benchmark framework, in: 24th WWW conference, 2015.