Out-of-distribution robustness in multilingual instruction-tuned models: the case of Aya101

Anonymous GitHub Repository

Abstract

Multilingual large language models(LLMs) like `Aya101` aim to support diverse languages, yet often perform poorly in low- and mid-resource settings. This study investigates the out-of-distribution robustness of `Aya101` in Polish and Slovak, focusing on its performance on human-written SK-QuAD and Po-QuAD datasets. As a baseline, we include in-distribution evaluation using machine-translated MultiQ. We apply four controlled input perturbations—synonym substitution, typos, rewording, and code-switching—to test the model's resilience to realistic variation. We evaluate model responses using BERTScore and manual annotation across grammatical correctness, factual accuracy, and language fidelity. The results show that `Aya101` gives fluent answers but often fails to provide correct facts in our out-of-distribution settings. In particular, factual accuracy drops sharply when we use inputs different from the training data. While the model handles surface-level noise reasonably well, it struggles with diverse human-written data. Our work underscores the need for more robust evaluation methods and higher-quality training data to improve generalization in multilingual LLMs.

1 Introduction

Large language models (LLMs) have achieved remarkable performance across a range of natural language processing tasks. However, their capabilities are often concentrated in high-resource languages, particularly English, leading to systematic disparities in performance for speakers of less-resourced languages (Joshi et al., 2020; Bender et al., 2021). This linguistic bias limits the accessibility and reliability of LLMs globally.

To address this imbalance, multilingual instruction-tuned models such as `Aya101` (Üstün et al., 2024) aim to provide broader language coverage. `Aya101`, trained with a focus on a wide set of typologically diverse languages, represents a new generation of multilingual models designed to improve inclusiveness in NLP. However, we still don't know much about how well they work in languages with fewer resources, especially when the input is changed in small but realistic ways.

Prior research has highlighted that multilingual models often struggle to generalize beyond their training distribution, especially when exposed to morphologically complex or mid-resource languages (Hung et al., 2022). Evaluation efforts such as MultiQ (Holtermann et al., 2024) have introduced multilingual QA benchmarks through machine translation, but translation artifacts often limit their realism. To improve the quality of evaluation in languages like Polish and Slovak, manually curated datasets such as Po-QuAD (Tuora et al., 2023) and SK-QuAD (Hládek et al., 2023) provide more linguistically authentic QA contexts.

Another key challenge in LLM deployment is robustness to input noise and variation. Recent frameworks such as CHECKLIST (Ribeiro et al., 2020) and contrastive evaluations (Gardner et al., 2020) have demonstrated that even state-of-the-art models can be vulnerable to minor perturbations such as synonym substitutions or typos. While these evaluations are well-established for English, they are rarely applied to multilingual contexts. Also, code-switching—when people mix two languages in one sentence—is common in multilingual communities. This makes things more complicated, and earlier studies have shown that even multilingual models can get confused or make mistakes when dealing with such mixed-language input (Hung et al., 2022).

In this paper, we evaluate the **out-of-distribution (OOD) robustness** of multilingual LLMs in Polish and Slovak. We focus on systematic prompt perturbations in a question answering (QA) setting (Ribeiro et al., 2020). Using both in-distribution (ID) data (machine-translated MultiQ) and out-of-distribution (OOD) data

(human-written SK-QuAD and Po-QuAD), we measure model resilience to four types of input variations: synonym substitution, typographical errors, rewording, and code-switching.

Our evaluation combines human annotation and an automatic metric, BERTScore (Zhang et al., 2020), allowing us to assess robustness across grammaticality, factual correctness, and language fidelity. We examine the strengths and limitations of `Aya101` and highlight the challenges of ensuring equitable performance across languages (Hendrycks and Dietterich, 2019; Hung et al., 2022).

2 Datasets

We evaluate the multilingual robustness of the `Aya101` language model, a 13-billion-parameter decoder-only transformer developed by Cohere For AI. `Aya101` is fine-tuned from the mT5 model (Xue et al., 2021), which was pretrained on a large-scale, multilingual corpus spanning 101 languages. The model underwent instruction tuning using several datasets, including xP3x, the Aya Collection, and ShareGPT-Command, to enhance its ability to follow task instructions across diverse linguistic settings.

The Aya Collection (Singh et al., 2024), in particular, was designed to support instruction-tuned large language models with multilingual coverage. While extensive, this dataset exhibits significant imbalance. English has the largest amount of data, with approximately 17.8 million rows, while languages with fewer resources, such as Polish and Slovak, are underrepresented, with only 4.45 million and 4.15 million rows, respectively. Moreover, a substantial portion of the Polish and Slovak examples are machine-translated rather than organically sourced. This may reduce the quality and naturalness of the language, which could negatively affect the model's performance in these languages.

To evaluate `Aya101`'s robustness under both expected and challenging conditions, we sample two evaluation datasets, targeting both in-distribution (ID) and out-of-distribution (OOD) settings for Polish and Slovak. These languages represent mid-resource, morphologically rich European languages with limited high-quality QA benchmarks. Each question is subjected to perturbations including synonym substitutions, typographical errors, rewordings, and code-switching, to assess performance under varied real-world conditions.

2.1 Dataset Construction

In-Distribution (ID): MULTIQ. We consider the MULTIQ dataset (Holtermann et al., 2024) as our ID benchmark. MultiQ is a multilingual QA dataset automatically translated from English that consists of simple and factual questions. Since most of the Polish and Slovak data on which `Aya101` was trained was also machine translated, we decided to use MultiQ as our ID dataset. We sample 50 Polish and 50 Slovak examples for evaluation to ensure a manageable annotation workload while preserving topical diversity. As a result of the translations being machine-generated, many questions exhibit minor grammatical errors or stylistic unnaturalness that is consistent with prior findings on MT-based multilingual corpora (Hung et al., 2022; Joshi et al., 2020).

Out-of-Distribution (OOD): SK-QuAD and Po-QuAD. Our OOD datasets include SK-QuAD (Hládek et al., 2023) and Po-QuAD (Tuora et al., 2023), manually curated from Slovak and Polish Wikipedia passages. These datasets consist of human-written question-answer pairs and contain no known overlap with training data of mainstream multilingual QA models; hence, we consider these datasets as out-of-distribution for `Aya101`. We selected a subset of 50 examples from the full QuAD datasets randomly while preserving topic diversity. This was designed to test the model's ability to handle real-world language and complex meaning in question-answering scenarios.

2.2 Dataset Validation

To ensure evaluation quality, we—as native speakers—manually assessed all evaluation subsets using the following criteria:

- **Fluency:** Text should read naturally without awkward phrasing or literal translation artifacts. This aligns with fluency checks described in Ribeiro et al. (2020).
- **Grammatical Correctness:** Evaluates syntax, morphology, and punctuation correctness, similar to the syntactic evaluation dimensions used in prior robustness work (Ribeiro et al., 2020).
- **Correct Translation:** Ensures Slovak and Polish MultiQ questions have the same meaning as the English MultiQ questions. This follows common accuracy dimensions in trans-

lation quality frameworks like MQM (Freitag et al., 2021).

- **Answer Correctness:** For SK-QuAD and Po-QuAD, we check whether the answer is factually correct and appropriately grounded in the context. This criterion draws on factual consistency evaluation approaches used in QA settings such as Honovich et al. (2021).

Annotations were conducted on a sample of 50 questions per dataset and language (200 total), following shared guidelines to ensure consistency. While the MultiQ datasets exhibited minor machine translation artifacts, they generally passed fluency and grammaticality checks. The human-written OOD datasets demonstrated high scores in all metrics.

Dataset	Evaluation Criteria	Score
Slovak MultiQ	Fluency	90%
	Grammatical Correctness	98%
	Correct Translation	94%
Polish MultiQ	Fluency	95%
	Grammatical Correctness	98%
	Correct Translation	96%
SK-QuAD	Fluency	96%
	Grammatical Correctness	98%
	Correct Answer	100%
Po-QuAD	Fluency	98%
	Grammatical Correctness	98%
	Correct Answer	98%

Table 1: Human evaluation of Slovak and Polish datasets based on fluency, grammatical correctness, and correctness of answers or translations. Scores represent the percentage of questions which we annotated as acceptable for each criterion.

3 Experimental Setup

The goal of this experimental setup is to evaluate the multilingual robustness of the instruction-tuned `Aya101` model using in-distribution (ID) and out-of-distribution (OOD) subsets in Slovak and Polish. Each subset contains 50 questions.

3.1 Datasets

Our evaluation focuses on two types of datasets in **Polish** and **Slovak**, chosen to represent both in-distribution and out-of-distribution conditions for multilingual QA.

The in-distribution *MultiQ* subsets reflect the type of data instruction-tuned models are commonly exposed to during training. Since the original MultiQ dataset does not have answer references, we created them manually.

The out-of-distribution *SK-QuAD* (Slovak) and *Po-QuAD* (Polish) subsets are more linguistically authentic and are not expected to overlap with the training distribution of `Aya101`.

Our subsets were created from the original datasets by picking 50 questions with four perturbations for each question, totaling 250 questions per subset to ensure a manageable annotation workload while preserving topical diversity. This means that having two subsets per language, we evaluated **1000** questions in total.

3.2 Prompt Perturbations

To simulate realistic input variation and test model robustness, we apply four types of controlled prompt perturbations to each question. These transformations are inspired by behavioral testing frameworks such as CHECKLIST (Ribeiro et al., 2020) and are motivated by prior work showing that these perturbations are common forms of input noise and user behavior in real-world multilingual applications (Belinkov and Bisk, 2018; Solorio et al., 2014).

The four perturbation types include: *Synonym Substitution*, *Typographical Errors*, *Rewording*, and *Code-Switching*. Each one is made to test how well the model can handle different challenges, while keeping the main meaning of the original prompt the same. Descriptions and examples of each type are provided in Table 2.

All perturbations were created by us manually to make sure they kept the same meaning and sounded natural. For each original question, we made four perturbed variants (one per type), resulting in five total versions per item. This setup enables a controlled comparison of `Aya101`'s response quality under varied input conditions.

3.3 Annotation Process

To evaluate the model's answers, we annotated them based on three main criteria inspired by prior evaluation frameworks for NLP robustness and multilingual QA:

- **Grammatical Correctness** — whether the answer is syntactically well-formed and conforms to standard language conventions (Ribeiro et al., 2020; Freitag et al., 2021).

- **Factual Correctness** — whether the answer is factually accurate and grounded in the given context, as studied in prior research (Honovich et al., 2021; Gardner et al., 2020).

Perturbation Type	Description and Examples
Synonym Substitutions	Replace words with synonyms to test lexical robustness. **EN**: "Who is the **leader** of Poland?" → "Who is the **chief** of Poland?" **PL**: "Kim jest **przywódca** Polski?" → "Kim jest **lider** Polski?" **SK**: "Kto je **vodca** Pol'ska?" → "Kto je **líder** Pol'ska?"
Typographical Errors	Introduce spelling mistakes to simulate input noise. **EN**: "What is the **capittal** of Slovakia?" **PL**: "Jaka jest **stolcia** Słowacji?" **SK**: "Aké je **hlavvné** mesto Slovenska?"
Rewording Questions	Restructure while preserving meaning. **EN**: "When was Napoleon born?" → "What is Napoleon's birth year?" **PL**: "Kiedy urodził się Napoleon?" → "W którym roku urodził się Napoleon?" **SK**: "Kedy sa narodil Napoleon?" → "Aký je rok narodenia Napoleona?"
Code-Switching	Mix English words in Polish/Slovak prompts. **PL**: "Jaki jest **capital** Polski?" **SK**: "Aké je **main city** Slovenska?"

Table 2: Types of prompt perturbations used to evaluate model robustness, with examples in English (EN), Polish (PL), and Slovak (SK).

- **Language Fidelity** — whether the output remains in the correct language without inappropriate code-switching, as discussed in multilingual evaluation research (Solorio et al., 2014; Hung et al., 2022).

We rated each answer on a simple binary scale (0 or 1) for each of these criteria.

Since we are native speakers of Polish and Slovak, we did all the annotations ourselves. To keep the process consistent, we followed shared guidelines and used example cases for reference.

This manual annotation helped us check things that automatic tools might miss.

3.4 Automatic Evaluation

Alongside manual annotation, we also used an automatic metric to measure how close the model's answers were to the correct ones. We chose **BERTScore**, which compares the meaning of two texts using word embeddings from a pre-trained BERT model.

We considered several other automatic evaluation metrics, including **BLEU** (Papineni et al., 2002), **ROUGE** (Lin, 2004), and **METEOR** (Banerjee and Lavie, 2005), which are commonly used in natural language generation tasks. Recent surveys have also questioned the continued reliance on surface-level metrics like BLEU and ROUGE in modern NLG evaluation, emphasizing the need for metrics that capture deeper semantic equivalence (Schmidtová et al., 2024). However, these metrics primarily rely on *surface-level overlap*, such as n-gram matching, and often fail to capture *semantic equivalence* when the wording differs but the meaning remains the same. This limitation is particularly problematic for our robustness analysis, where we expect model outputs to use varied phrasing in response to perturbed prompts.

We chose **BERTScore** because it evaluates *semantic similarity* by leveraging contextual embeddings from a pre-trained BERT model (Zhang et al., 2020). This allows it to recognize when two answers are meaningfully similar even if their surface forms diverge. This property aligns closely with our human annotation criteria (factual correctness and language fidelity), making BERTScore a more appropriate fit for our multilingual robustness evaluation.

We calculated the average BERTScore for each type of prompt (original and all four perturbations). We also measured how much the score changed between the original and the perturbed versions. This helped us understand how robust the model was to small changes in the input.

3.5 Implementation

We used the publicly available `Aya101` model from HuggingFace's Transformers library for all experiments. Model inference was conducted in a **zero-shot** setting, meaning that we provided only the question as input, without including any additional examples or context. We did not perform any parameter tuning or prompt tuning; the model was used in its default configuration without modification.

All inference was executed on our university's *high-performance computing (HPC) network*, using batch processing to efficiently generate responses across multiple datasets. All outputs were exported to spreadsheets for manual annotation. The entire process—from prompt perturbation cre-

ation to annotation—was carried out manually and kept consistent across both languages and datasets. For examples, see Appendices A and B.

In preparation for evaluation, we structured our analysis around both automatic and manual metrics. For each perturbation type, we computed average BERTScore and manual ratings. This allowed us to systematically assess the impact of perturbations on model performance and identify frequent failure modes such as grammatically correct but factually incorrect answers.

4 Experiment Results

To assess `Aya101`'s performance in low-resource languages, we evaluated its robustness using both in-distribution (MultiQ) and out-of-distribution (SK-QuAD and Po-QuAD) datasets in Slovak and Polish. Each dataset consisted of 50 QA pairs, evaluated under five prompt variants to probe robustness to surface-level changes. We used BERTScore F1 (Zhang et al., 2020) as an automatic metric to measure semantic similarity between generated and reference answers. Additionally, manual evaluation was conducted for three criteria: grammatical correctness, factual correctness, and language fidelity.

4.1 Slovak Evaluation

As shown in Table 3, `Aya101` achieved higher semantic similarity (Avg. BERTScore: **0.791**) on the in-distribution MultiQ dataset compared to (Avg. BERTScore: **0.717**) on the out-of-distribution SK-QuAD dataset. The scores were close across perturbations, ranging from **0.698** to **0.728** in SK-QuAD and **0.754** to **0.820** in Slovak MultiQ, indicating high robustness of the model. The lower average score for the OOD dataset shows the model struggles to match the correct meaning when processing new, unfamiliar data. However, it is important to note that BERTScore captures semantic similarity, not factual accuracy or correctness. In some cases, the model produced fluent answers that were semantically close but factually incorrect. Manual evaluation confirms this distinction as factual correctness dropped significantly from 50.8% in MultiQ to only 13.2% in SK-QuAD. Grammatical correctness remained high across both datasets, indicating that `Aya101` maintains linguistic fluency even when its factual content degrades. Language fidelity was consistent, showing that the model keeps the answers in Slovak, even on OOD prompts. The results of the manual evaluation for Grammatical Correctness, Factual Correctness, and Language Fidelity across all perturbation types for the Slovak datasets can be seen in Figures: 1, 2 and 3

4.2 Polish Evaluation

A similar pattern was observed in the Polish evaluation (Table 4). `Aya101` obtained an average BERTScore of **0.775** on MultiQ and **0.720** on Po-QuAD, both scores very similar to the Slovak ones. Similarly, the F1 scores were roughly consistent for different perturbations, ranging from **0.712** to **0.730** in Po-QuAD and **0.763** to **0.787** in Polish MultiQ. Manual scores showed factual correctness dropped sharply from 41.6% in ID to 7.2% in OOD. Grammatical Correctness was also higher in MultiQ (87.6%) than in Po-QuAD (71.6%) suggesting that the model's language quality degrades more noticeably in Polish than in Slovak under OOD conditions. The results of the manual evaluation for Grammatical Correctness, Factual Correctness, and Language Fidelity across all perturbation types for the Polish datasets can be seen in Figures: 1, 2 and 3

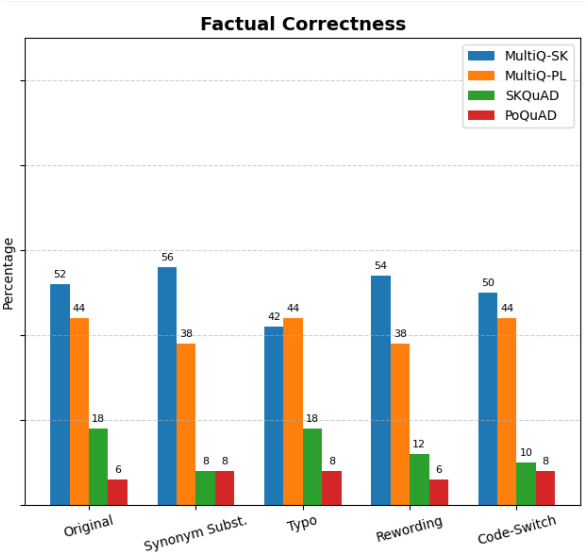

Figure 1: Manual evaluation results for Factual Correctness across different prompt perturbations and datasets.

Dataset	Perturbation Type	BERTScore F1	Grammar (%)	Factual (%)	Language Fidelity (%)
SK-QuAD	Original Question	0.698	92	18	98
	Synonym Substitution	0.721	90	8	98
	Typographical Errors	0.724	90	18	98
	Rewording	0.716	88	12	96
	Code-Switching	0.728	84	10	92
	Average	**0.717**	**88.8**	**13.2**	**96.4**
MultiQ	Original Question	0.777	94	52	98
	Synonym Substitution	0.793	90	56	98
	Typographical Errors	0.810	92	42	100
	Rewording	0.820	92	54	100
	Code-Switching	0.754	90	50	78
	Average	**0.791**	**91.6**	**50.8**	**94.8**

Table 3: Slovak evaluation: BERTScore F1 and manual annotation scores (grammatical correctness, factual correctness, and language fidelity) across prompt perturbation types.

Dataset	Perturbation Type	BERTScore F1	Grammar (%)	Factual (%)	Language Fidelity (%)
Po-QuAD	Original Question	0.712	70	6	100
	Synonym Substitution	0.718	72	8	100
	Typographical Errors	0.722	78	8	100
	Rewording	0.730	78	6	100
	Code-Switching	0.716	60	8	92
	Average	**0.720**	**71.6**	**7.2**	**98.4**
MultiQ	Original Question	0.787	90	44	100
	Synonym Substitution	0.763	94	38	100
	Typographical Errors	0.783	84	44	98
	Rewording	0.766	82	38	98
	Code-Switching	0.777	88	44	78
	Average	**0.775**	**87.6**	**41.6**	**94.8**

Table 4: Polish evaluation: BERTScore F1 and manual annotation scores (grammatical correctness, factual correctness, and language fidelity) across prompt perturbation types.

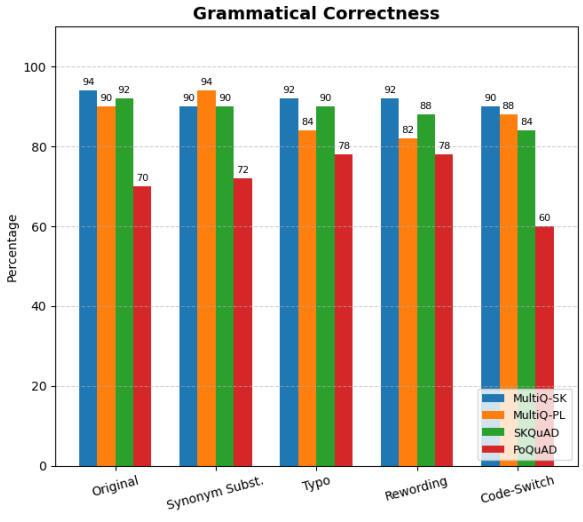

Figure 2: Manual evaluation results for Grammatical Correctness across different prompt perturbations and datasets.

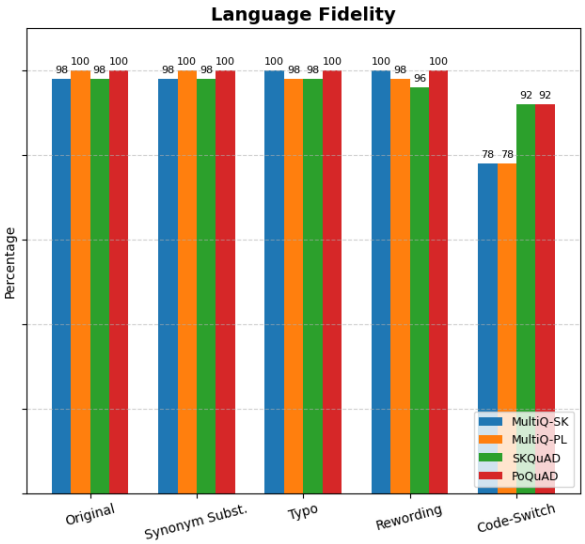

Figure 3: Manual evaluation results for Language Fidelity across different prompt perturbations and datasets.

5 Discussion

5.1 Key Findings and the Limits of OOD Generalization

Our evaluation of the multilingual instruction-tuned model `Aya101` shows a clear gap between performance on in-distribution (ID) and out-of-distribution (OOD) datasets in both Polish and Slovak. On the Slovak MultiQ dataset, the model achieved 50.8% factual correctness, while this dropped to just 13.2% on SK-QuAD. A similar pattern was observed in Polish, with factual accuracy decreasing from 41.6% on MultiQ to 7.2% on Po-QuAD. While the model maintains strong fluency and grammatical correctness, its factual accuracy drops sharply in OOD contexts. These findings align with prior work highlighting the difficulty multilingual models face when generalizing to more complex, less predictable linguistic input (Hung et al., 2022; Joshi et al., 2020; Figueras et al., 2025).

This large performance gap might suggest that `Aya101` struggles to generalize factual knowledge when the test data is more complex and human-written. A likely contributing factor is the nature of `Aya101`'s training data, which relies heavily on machine-translated text for both Polish and Slovak. While the dataset is large, it may include various translation artifacts and be of poorer quality than human-written text (Artetxe et al., 2020). Models may overfit to consistent syntactic and lexical patterns often found in machine-translated text (Toral and Sánchez-Cartagena, 2017; Wei et al., 2024), which may inflate performance on evaluation datasets containing such features. In contrast, our QuAD subsets consist of human-written, varied inputs that better reflect real-world language use, thereby revealing the model's struggles with factual correctness and generalization.

Despite these limitations, `Aya101` showed resilience to several types of prompt perturbations. BERTScore values remained relatively stable across variations like typographical errors and code-switching, suggesting that the model can handle surface-level noise. However, semantic similarity metrics like BERTScore focus on meaning overlap and can miss factual mistakes. In our evaluation, we saw several cases where the answers were fluent but still wrong—showing why human evaluation is important when testing multilingual QA models.

These findings underscore a core limitation in multilingual LLMs: linguistic fluency is orthogonal to factual reliability. Strong performance on in-distribution datasets like MultiQ does not necessarily translate to robust generalization on more natural and diverse out-of-distribution data. While machine translation may be a contributing factor, other elements such as task format, content complexity, or annotation differences may also play a role. This highlights the broader challenge of achieving reliable multilingual generalization, especially in settings with fewer resources and for tasks that demand factual precision.

5.2 Perturbation Robustness Analysis

We tested how `Aya101` responds to small changes in the input, such as synonym substitution, typos, rewording, and code-switching. While BERTScore stayed stable across these changes, our manual evaluation showed that factual accuracy and grammar often got worse, especially in our out-of-distribution samples.

In SK-QuAD, factual correctness was lowest for synonym substitution (8%) and code-switching (10%), while original and typo inputs were higher (18%). Rewording caused a moderate drop (12%). Grammatical correctness stayed high (84–92%), and language fidelity was above 90% for all prompts.

On SK-MultiQ, factual accuracy was much better (42–56%) and more consistent across all prompts. This shows that the model handles variations better when the data is similar to its training set. Code-switching lowered language fidelity to 78%, the biggest drop.

In Po-QuAD, factual accuracy was very low (6–8%) across all inputs, and grammar dropped sharply under code-switching (60%). PL-MultiQ again performed better, with factual correctness reaching 44% and only small drops under synonym substitution and rewording.

Across all samples, code-switching reduced language fidelity the most. Typos had the least impact. These results show that `Aya101` is fairly good at handling noisy input. This also shows that automatic scores like BERTScore might miss important errors, and human evaluation is needed to get the proper insights.

5.3 Language-Specific Differences

Although `Aya101` was evaluated on both Polish and Slovak, we observed small but meaningful differences in its performance across the two languages. For example, grammatical correctness was slightly

higher in Slovak (91.6% on MultiQ) than in Polish (87.6%), and the model also produced more fluent answers in Slovak on the out-of-distribution SK-QuAD dataset.

On the other hand, Polish showed better language fidelity in the OOD setting (98.4% on PoQuAD vs. 96.4% on SK-QuAD), meaning the model was slightly more consistent in keeping the output language correct. These differences may stem from subtle variations in the amount and quality of training data available for each language.

As noted in Section 2, the Aya Collection includes approximately 4.45 million Polish and 4.15 million Slovak examples, predominantly generated through machine translation. These examples were used during the instruction tuning phase. The collection covers a variety of tasks, including question answering (QA), the exact proportion of QA-specific data for Polish and Slovak being 15.1% and 16.1% respectively. Both language datasets were translated from English sources using the NLLB 3.3B model (Team et al., 2022), providing consistency in the translation pipeline. However, small differences in translation quality or topical coverage may still have contributed to the observed performance variation. Additionally, `Aya101` is fine-tuned from the mT5 model, which was pretrained on the multilingual C4 (mC4) dataset (Xue et al., 2021).

Still, the main trend remains the same: our experiment setup suggests that `Aya101` performs better on in-distribution data and struggles with factual correctness in human-written, out-of-distribution examples across both languages. This suggests that the challenge of robust generalization is not language-specific, but rather a broader issue in multilingual settings with fewer resources.

5.4 Future Work

There are several directions for future research that could build on this study. First, we would extend the evaluation to more languages, including ones with even fewer data available, to better understand how instruction-tuned models like `Aya101` perform across a broader linguistic spectrum. This would also help determine whether the trends we observed in Polish and Slovak generalize to other language families and writing systems.

Second, evaluating additional models would allow for comparisons between different multilingual architectures and training strategies. This could include closed-source models or newer instruction-tuned alternatives to `Aya101`.

Another key area is automation. In this study, all prompt perturbations were created manually to ensure quality. While effective, this approach is time-consuming and limits scalability. In the future, we aim to develop tools for automatically generating controlled perturbations across languages, which would make robustness testing more consistent and efficient.

Finally, future work could focus on improving factuality under input variation. This could involve methods such as contrastive instruction tuning (Honovich et al., 2021), multilingual retrieval-augmented generation, or fine-tuning models on adversarial examples. Improving the reliability of factual answers remains an important challenge for using multilingual LLMs in real-world settings.

6 Conclusion

This study investigated the multilingual capabilities of the open-source large language model `Aya101`, with a focus on its performance in Polish and Slovak. By evaluating the model on both in-distribution (MultiQ) and out-of-distribution (PoQuAD and SK-QuAD) subsets, we observed clear differences in output quality, particularly in factual accuracy and grammaticality.

Our results have shown that `Aya101` gives factually correct outputs significantly more frequently on our machine-translated subsets in comparison to our more diverse human-written subsets. We think this discrepancy occurs not only due to translation artifacts in training data but also reflects structural and linguistic differences between the datasets themselves. The out-of-distribution datasets (PoQuAD and SK-QuAD) contain more complex, naturally phrased questions and a wider variety of topics, and `Aya101`'s worse performance on them could expose limitations in the model's ability to generalize beyond simplified, synthetic patterns. These findings might suggest that `Aya101`'s apparent fluency in low-resource languages may overstate its actual robustness in real-world scenarios.

This study highlights the importance of critically assessing language models in multilingual contexts, especially where the underlying training data is sparse or machine-generated. Future work should explore improved training datasets, better evaluation metrics for factuality, and fine-tuning strategies for underrepresented languages.

Limitations

In our study, we used both automatic tools and manual checks to evaluate the model's answers. Instead of evaluating every output, we selected a carefully designed sample of 1,000 responses. This size allowed us to include multiple languages (Polish and Slovak), both in-distribution and out-of-distribution samples, and four types of prompt perturbations—giving us broad coverage while keeping the task manageable. As native speakers of both languages, we followed shared guidelines and example cases to stay consistent. Still, some personal interpretation may remain—for example, we occasionally differed on whether an answer was completely correct or sounded fully natural.

Another limitation concerns the scope of the model itself. In our evaluation, we used the base instruction-tuned version of `Aya101` without any additional fine-tuning or task-specific adaptation. This reflects a realistic usage scenario, where an end user relies on the model as-is for general factoid question answering—the type of task it was originally trained and optimized for. While this setup makes our results applicable to typical use cases, they may not extend to situations where the model is further customized for specialized domains.

Finally, our evaluation focused only on two languages—Polish and Slovak—which means the findings may not fully represent `Aya101`'s behavior in other languages with fewer resources. For instance, languages with different scripts (e.g., Devanagari or Arabic), or those with more complex morphology (e.g., Finnish or Hungarian), may pose different challenges for the model due to limited and potentially less representative training data.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2020. Translation artifacts in cross-lingual transfer learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7674–7684, Online. Association for Computational Linguistics.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? . In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Blanca Calvo Figueras, Eneko Sagarzazu, Julen Etxaniz, Jeremy Barnes, Pablo Gamallo, Iria De Dios Flores, and Rodrigo Agerri. 2025. Truth knows no language: Evaluating truthfulness beyond english.

Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, Nitish Gupta, Hannaneh Hajishirzi, Gabriel Ilharco, Daniel Khashabi, Kevin Lin, Jiangming Liu, Nelson F. Liu, Phoebe Mulcaire, Qiang Ning, Sameer Singh, Noah A. Smith, Sanjay Subramanian, Reut Tsarfaty, Eric Wallace, Ally Zhang, and Ben Zhou. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323, Online. Association for Computational Linguistics.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations.

Daniel Hládek, Ján Staš, Jozef Juhár, and Tomáš Koctúr. 2023. Slovak dataset for multilingual question answering. *IEEE Access*, 11:32869–32881.

Carolin Holtermann, Paul Röttger, Timm Dill, and Anne Lauscher. 2024. Evaluating the elementary multilingual capabilities of large language models with multiq.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. q^2: Evaluating factual consistency in knowledge-grounded dialogues via question generation and question answering.

Chia-Chien Hung, Anne Lauscher, Ivan Vulić, Simone Paolo Ponzetto, and Goran Glavaš. 2022. Multi2woz: A robust multilingual dataset and conversational pretraining for task-oriented dialog.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. 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, pages 6282–6293, Online. Association for Computational Linguistics.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Patrícia Schmidtová, Saad Mahamood, Simone Balloccu, Ondřej Dušek, Albert Gatt, Dimitra Gkatzia, David M. Howcroft, Ondřej Plátek, and Adarsa Sivaprasad. 2024. Automatic metrics in natural language generation: A survey of current evaluation practices. In *Proceedings of the 17th International Natural Language Generation Conference*, pages 557–583, Tokyo, Japan. Association for Computational Linguistics.

Shivalika Singh, Freddie Vargus, Daniel Dsouza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura OMahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Minh Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. 2024. Aya dataset: An open-access collection for multilingual instruction tuning.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Antonio Toral and Víctor M. Sánchez-Cartagena. 2017. A multifaceted evaluation of neural versus phrase-based machine translation for 9 language directions.

Ryszard Tuora, Aleksandra Zwierzchowska, Natalia Zawadzka-Paluektau, Cezary Klamra, and Łukasz Kobyliński. 2023. Poquad - the polish question answering dataset - description and analysis. pages 105–113.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model.

A Appendix A: Spreadsheet Data

title	context	question	fluen	comp	com	references	predictions	FL-FC-LF	Synonym Su
Jozef Filus	Gymnázi	V ktorom roku emigroval rímskokatolícky misionár Jozef Filus do Nemecka ?	1	1	1	1951	Roku 1831	1-0-1	V ktorom rok
Jozef Filus	Gymnázi	Kde dokončil rímskokatolícky misionár Jozef Filus svoje teologické štúdiá ?	1	1	1	v St. Augustine	Jozef Filus dokončil svoje teolo	1-0-1	Kde ukončil
Jozef Filus	Gymnázi	Kde študoval rímskokatolícky misionár Jozef Filus filozofiu ?	1	1	1	v Nitre	Štrasburg	1-0-1	Kde sa vzde
Jozef Filus	Gymnázi	Kam emigroval rímskokatolícky misionár Jozef Filus v roku 1951 ?	1	1	1	do Nemecka	V roku 1951 emigroval rímskok	1-0-1	Do akej kraji
Jozef Filus	Gymnázi	V ktorom roku bol rímskokatolícky misionár Jozef Filus vysvätený za kňaza ?	1	1	1	1952	Rok 1914	1-0-1	Kedy bol rím
Holub plužik	Na hniez	Ako dlho trvá výchova mláďat na hniezde u Holuba plužika ?	1	1	1	Výchova mláďat	Od mája do septembra.	1-0-1	Aký čas trvá
Holub plužik	Na hniez	Aké podmienky vyžaduje Holub plužik na hniezdenie ?	1	1	1	Na hniezdenie pa	Hniezdi v suchom a lenivom pr	1-1-1	Aké požiada
Holub plužik	Na hniez	Čo predchádza páreniu u Holuba plužika ?	1	1	1	zobáčikovanie, h	Holuby plužiky sa pária v obdol	1-0-1	Aké správan
Holub plužik	Holub plu	Ako vyzerá Holub plužik ?	1	1	1	Hlavu a temeno	Holub plužik je vták s krátkym c	1-0-1	Aký je vzhľa
Holub plužik	Holub plu	Aké rozpätie krídel má Holub plužik ?	1	1	1	63-69 cm	Rozpätie krídel holuba plužika j	1-0-1	Aké veľké kr
Abbey Road	Legenda:	Kedy vznikla legenda o smrti Paula McCartneyho ?	1	1	1	12. októbra, 196	Rok 1980	1-0-1	V ktorom rok
Abbey Road	Legenda:	Kvôli čomu sa vznikla legenda o smrti Paula McCartneyho ?	1	1	1	neznámy hlas oz	Legenda o smrti Paula McCartr	1-0-1	Z akého dôv
Abbey Road	Legenda:	Na čo upozornil neznámy hlas vo vysielaní ohľadom Paula McCartneyho ?	1	1	1	Legenda o Paulc	Neznámy hlas upozornil na to,	1-0-1	Na čo pouká
Abbey Road	Legenda:	Kto mal byť vyfotený na obale platne Abbey Road ?	1	1	1	Paulov dvojník	Beatles	1-1-1	Kto mal byť
Abbey Road	V začiatk	Koľko času mal fotograf na zhotovenie snímky pre album Abbey Road ?	1	1	1	iba okolo 10 min	Päťdesiat minút.	1-0-1	Koľko času r

Figure 4: Example of spreadsheet data for SK-QuAD original QAs

Synonym Substi	predictions	FL-FC-LF	Typo	predictions	FL-FC-LF	Rewording	predictions	FL-FC-LF	Code-Switching	predictions	FL-FC
V ktorom roku sı	Rok 1786	1-0-1	V ktorom roku er	Rok 1831	1-0-1	Kedy odišiel rím	Roku 1831	1-0-1	In which roku err	1740	1-0-1
Kde ukončil rím	V roku 1854 aba	1-0-1	Kde dokončil rím	V roku 1854 aba	1-0-1	V ktorom meste	V Ríme.	1-0-1	Where dokončil	Jozef Filus doko	1-0-1
Kde sa vzdeláva	V Ríme	1-0-1	Kde študoval rím	V Ríme	1-0-1	V ktorom meste	Mesto Bologna	1-0-1	Kde studied rím	Sorbonne	1-0-0
Do akej krajiny o	Južná Afrika	1-0-1	Kam emigroval r	V roku 1951 emi	1-0-1	Do ktorej krajiny	Slovensko	1-0-1	Kam emigrated r	V roku 1951 emi	1-0-1
Kedy bol rímskol	Rok 1914	1-0-1	V ktorom roku bo	Rok 1914	1-0-1	Kedy bol Jozef F	Rok ordinatory 1	0-0-0	In which year bo	1846	1-0-1
Aký čas trvá stad	Holub plužik sa t	1-1-1	Ako dlho trva vyc	Hoci je to závisl	1-1-1	Ako dlho trvá, ký	Odhaduje sa, že	1-1-1	Ako dlho lasts th	Holuby plúžkové	1-0-1
Aké požiadavky	Holuby plužiky p	1-0-1	Aké podmienky	Holub plúžik je v	1-0-1	Aké faktory sú dı	Prvá. Teplota voc	1-0-1	Aké conditions v	Holub plužik je s	1-0-1
Aké správanie pi	Holuby plužiky s	1-0-1	Čo predchadza p	Holuby plužiky s	1-0-1	Aké správanie vj	Pred párením sa	1-1-1	What happens pı	Pred párením sa	1-0-1
Aký je vzhľad Hc	Pštrosi majú král	1-0-1	Ako vzyera Holu	Holuby jedia listy	1-0-1	Čo charakterizuj	Plodný orol je mi	1-0-1	How does Holub	Holub plužik je r	1-0-1
Aké veľké krídla	Holub plužik má	1-0-1	Ake rozpatie kríd	Holub plužik má	1-0-1	Koľko meria rozp	Rozpätie krídel h	1-0-1	Aký wingspan m	Psolná dĺžka krí	1-0-1
V ktorom roku sı	Rok 1980	1-0-1	Kedy vznik lege	Rok 2003	1-0-1	Kedy sa začalo r	Rok 2009	1-0-1	Kedy vznika lege	V roku 1963.	1-0-1
Z akého dôvodu	Povery o úmrtí P	1-0-1	Kvôli čomu sa vz	Legenda o smrti	0-0-1	Čo spôsobilo vzr	V roku 2009 sa c	1-0-1	Why vznikla lege	McCartney měl v	0-0-0
Na čo poukázal	McCartneyho srt	1-0-1	Na čo upozornil i	Na jeho homose	1-0-1	Čo spomenul ne	Chcem byť tvoj r	1-0-1	Na čo upozornil	McCartneyho srt	1-0-1
Kto mal byť zast	John Lennon	1-1-1	Kto mal bť vyfot	Ringo Starr.	1-1-1	Koho plánovali z	John Lennon	0-1-1	Kto mal byť vyfot	John Lennon	0-0-1
Koľko času mal	Päť dní.	1-0-1	Kolko času mal f	Fotograf mal na z	1-0-1	Aký časový limit	Fotograf mal ten	1-1-1	Kolko času maj	Päť minút.	1-1-1

Figure 5: Example of spreadsheet data for SK-QuAD perturbed QAs

B Appendix B: Examples of scripts executed through ITU's HPC network

```
(/opt/itu/condaenvs/pytorch) [jkoz@slurmhead project]$ cat aya101_script.py
from transformers import AutoModelForSeq2SeqLM, AutoTokenizer

checkpoint = "CohereForAI/aya-101"

tokenizer = AutoTokenizer.from_pretrained(checkpoint)
aya_model = AutoModelForSeq2SeqLM.from_pretrained(checkpoint)

skinputs0 = tokenizer.encode('Koľko farieb je v dúhe?', return_tensors='pt')
skoutputs0 = aya_model.generate(skinputs0, max_new_tokens=128)
print('Koľko farieb je v dúhe?', tokenizer.decode(skoutputs0[0]))

skinputs1 = tokenizer.encode('Koľko odtieňov obsahuje dúha?', return_tensors='pt')
skoutputs1 = aya_model.generate(skinputs1, max_new_tokens=128)
print('Koľko odtieňov obsahuje dúha?', tokenizer.decode(skoutputs1[0]))

skinputs2 = tokenizer.encode('Koíko farieb je v dúhhe?', return_tensors='pt')
skoutputs2 = aya_model.generate(skinputs2, max_new_tokens=128)
print('Koíko farieb je v dúhhe?', tokenizer.decode(skoutputs2[0]))

skinputs3 = tokenizer.encode('Aký je počet farieb, ktoré tvorí dúha?', return_tensors='pt')
skoutputs3 = aya_model.generate(skinputs3, max_new_tokens=128)
print('Aký je počet farieb, ktoré tvorí dúha?', tokenizer.decode(skoutputs3[0]))

skinputs4 = tokenizer.encode('Koľko colors je v dúhe?', return_tensors='pt')
skoutputs4 = aya_model.generate(skinputs4, max_new_tokens=128)
print('Koľko colors je v dúhe?', tokenizer.decode(skoutputs4[0]))

skinputs5 = tokenizer.encode('Aká je sila, ktorá priťahuje predmety k Zemi?', return_tensors='pt')
skoutputs5 = aya_model.generate(skinputs5, max_new_tokens=128)
print('Aká je sila, ktorá priťahuje predmety k Zemi?', tokenizer.decode(skoutputs5[0]))
```

Figure 6: Example script for running questions with Aya101 through HPC.

```
(/opt/itu/condaenvs/pytorch) [jkoz@slurmhead project]$ ls
aya101_script.py  job.28197.out  job.42950.out  job.43124.out  job.46317.out  job.46332.out
aya_multiq.py     job.28447.out  job.42967.out  job.43135.out  job.46318.out  job.46679.out
euro_script.py    job.28453.out  job.42975.out  job.45342.out  job.46329.out
gpu.job           job.28454.out  job.43096.out  job.45925.out  job.46330.out
(/opt/itu/condaenvs/pytorch) [jkoz@slurmhead project]$ cat gpu.job
#!/bin/bash

#SBATCH --job-name=simple-gpu    # Job name
#SBATCH --output=job.%j.out      # Name of output file (%j expands to jobId)
#SBATCH --cpus-per-task=64       # Schedule four cores
#SBATCH --gres=gpu               # Schedule a GPU
#SBATCH --time=02:00:00          # Run time (hh:mm:ss) - run for one hour max

python3 euro_script.py

echo "Running on $(hostname):"
nvidia-smi
```

Figure 7: Example for running the questions script through HPC.