The Translation Barrier Hypothesis:
Multilingual Generation with Large Language Models Suffers from Implicit Translation Failure

Niyati Bafna[1], Tianjian Li[1], Kenton Murray[1],
David R. Mortensen[2], David Yarowsky[1], Hale Sirin[1], and Daniel Khashabi[1]
[1]Johns Hopkins University, Center for Language and Speech Processing;
[2]Language Technologies Institute, Carnegie Mellon University
{nbafna1,tli104}@jhu.edu

Abstract

Multilingual generation with large language models (LLMs) is often of poor quality for mid- to low-resource languages, but the causes for this are not well-understood. We first demonstrate the existence of an implicit *task-solving→translation* pipeline for generation, whereby the model first solves the required task in a largely target-language-agnostic manner, and subsequently translates answer concepts into the intended target language. We hypothesize that the failure of the translation stage, despite task-solving success, is an important culprit for the observed low quality of final outputs, and formalize this as the *translation barrier hypothesis*. We quantify the extent to which either stage in the pipeline is responsible for final failure for a word translation task across 108 language pairs, and find that the translation barrier explains a dominant portion of error for a majority of language pairs, and is especially severe for low-resource target languages. Our results highlight an important bottleneck for end-to-end multilingual generation, relevant for future work seeking to improve multilinguality in LLMs. [1]

1 Introduction

Large language model (LLM) generation in mid-to-low-resource languages (LRLs) is of notoriously worse quality than that in high-resource languages (HRLs) (Robinson et al., 2023; Hendy et al., 2023; Cahyawijaya et al., 2024; Jiao et al., 2023; Shen et al., 2024), and further suffers the problem of off-targetness, where an LM fails to output text in the intended language of generation (Zhang et al., 2020; Li and Murray, 2023; Marchisio et al., 2024). While ongoing work attempts to use explicit cascaded approaches, i.e. LLM generation into HRLs, followed by translation into the intended target, for multilingual generation (Shi et al., 2022; Liu et al.,
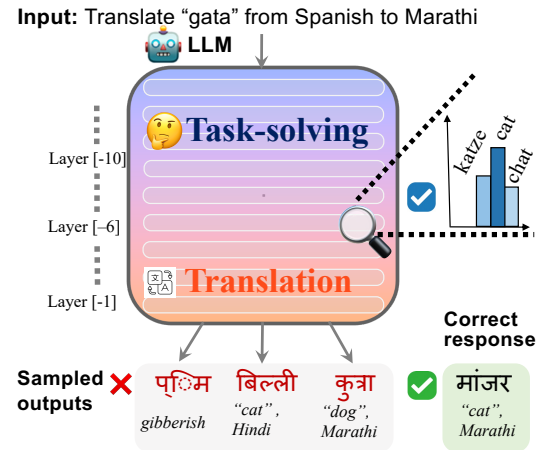

Figure 1: *Task-solving* succeeds with the correct answer concept (i.e. `cat`) discovered in intermediate layers, expressed in various HRLs, but the model fails to realize or *translate* the concept into the target LRL.

Translation Barrier Hypothesis: *Translation failure in LLMs accounts for a large proportion of poor quality final outputs for multilingual generation, where translation refers to model-internal transfer of answer concepts into the intended output language, given an implicit* task-solving→translation *cascade.*

2024), or attempt to include support for more languages in LLMs for end-to-end generation, we still lack a systematic understanding of the reasons for and mechanisms of the above failures.

Work from mechanistic interpretability shows that intermediate layer representations of multilingual LLMs are close to English representations even when prompted to generate in other languages, and that the models demonstrate language-specific behavior only in late layers of the model (Wendler et al., 2024; Foroutan et al., 2022; Tang et al., 2024; Kojima et al., 2024). Building upon this intuition, we posit a *model-internal task-solving→translation* cascade, with the middle lay-

---

[1]github.com/JHU-CLSP/translation-barrier

1541

ers of the LLM responsible for *task-solving*, or discovering the required output concepts, in a target-language-agnostic manner, and the last few layers responsible for realizing those concepts in the target language, through *translation*.

We present the **translation barrier hypothesis**, illustrated and stated in Figure 1, regarding the poor quality of multilingual generation. This hypothesis has important consequences for end-to-end multilingual generation. If translation failure, as opposed to task-solving failure, is the dominant culprit for low multilingual performance, then we can modularize or investigate architectural alternatives for the final LLM layers, or seek to induce better transfer in other ways, while still relying on powerful LLM task processing for end-to-end generation in LRLs. On the other hand, if task-solving capabilities themselves fail for LRL targets, explicit cascading methods become more viable options, calling for more investment in external specialized HRL-LRL MT systems.

Our work quantifies the translation barrier on a word translation task, for two mainstream multilingual models, in order to demonstrate the extent of this issue. We choose word translation as a minimal generation task: in this simple case, *task-solving* consists simply of realizing the semantics or the content of the input term, whereas *translation* refers to generating the target language term for that concept. We use logit lens to retrieve model-internal hypotheses across different layers, assessing whether the model has arrived at the appropriate semantics for a particular input regardless of language; we also assess the on-target accuracy of the final outputs. This allows us to demonstrate the existence of the posited pipeline. We then test our hypothesis, by quantifying the extent to which either stage in the pipeline is responsible for final failure.

We make the following contributions:

- We formalize the **translation barrier hypothesis**, highlighting a key mechanism of failure for multilingual generation with LLMs.

- Working with a word translation task, we provide evidence for the existence of an implicit *task-solving→translation* pipeline for LLM generation over 36 (target) × 3 (source) language pairs, and develop a framework within which we can measure and compare the contribution of *task-solving* and *translation* failure to total final failure.

- We characterize the languages in which task-solving occurs, going beyond English-centric previous work to show that intermediate layer representations are in fact multilingual to some extent and dominated by model-supported languages, with English foremost.

- We find that *translation* failure dominates ($> 50\%$) final failure for 65% and 78% of all language pairs for `aya-23` and `llama-3.1` respectively, and that it especially forms the bottleneck for generation in low-resource or unsupported *target* languages. On the other hand, we find that *task-solving* is the bottleneck in the case of a low-resource *source* language, and is also affected to some degree by the intended target language.

- We provide case studies with a larger sized model and on a different task. These yield consistent findings with the above, inviting future work in understanding the translation barrier in different settings.

2 Experimental Setup

Word translation dataset Given a word translation task between a source and target language, we are interested in judging the semantics of intermediate layer outputs regardless of language. We create a multiparallel dataset consisting of translations of words aligned across all languages of interest. We take 400 source words in English, split equally among nouns, verbs, adjectives, and adverbs, and use the Google Translate API[2] to collect a set of translations per word, including synonyms, for all languages and for English itself. This allows us to construct a lexicon between any source and target language for the same concepts, as well as assess off-target accuracy in all other languages per concept. We manually inspected the source words and lexicons for correctness and coverage (see Appendix B for more details).

Models We choose two performant mainstream models with a differing extents of multilinguality to conduct our experiments on: Aya-23-8B (`aya-23`) (Aryabumi et al., 2024) and Llama-3.1-8B-Instruct (`llama-3.1`) (Grattafiori et al., 2024). These are both 8-billion-parameter, decoder-only Transformer models, fine-tuned for instruction-following in 23 and 8 languages respectively. See

---

[2]translate.google.com

Appendix C for generation hyperparameters and our prompting setup.

Languages By virtue of their data-hungry nature, mainstream LLMs only support a limited set of usually high- and mid-resource languages, but may be used and are claimed to demonstrate incident multilingualism in other languages (Blevins and Zettlemoyer, 2022). We study 36 target languages, including all languages supported by either model,[3] and a mix of 11 additional mid- and low-resource languages unsupported by both models. See Appendix A for the full list of supported and unsupported languages and their resource level. We use three different source languages against all the above targets: a high-resource supported language (Spanish), a mid-resource supported language (Hindi), and a mid-to-low resource unsupported language (Telugu). Note that "source" language refers to that of the translation task input; the task instruction is always provided in English. We consider all the above 36 languages as candidates for task-solving languages in intermediate model layers.

3 Quantifying Task-solving and Translation failure

The translation barrier hypothesis presupposes the existence of a model-internal *task-solving→translation* pipeline (demonstrated and discussed in § 4), with model success predicated on task-solving success followed by translation success. We can study this hypothesis in the context of a given task by quantifying the extent to which translation failure is responsible for poor final outputs, as opposed to task-solving failure.

Obtaining intermediate layer outputs We use logit lens (Nostalgebraist, 2020) to obtain text outputs from intermediate layers. Let x be an input prompt and $h_i(x)$ be the output of layer i. Logit lens applies the unembedding matrix W_u and softmax to $h_i(x)$ to produce a probability distribution over the vocabulary at layer i:

$$p_i(\cdot \mid x) = \mathrm{softmax}\left(W_u \cdot \mathrm{norm}\left(h_i(x)\right)\right)$$

where norm refers to RMSNorm (Zhang and Sennrich, 2019) for `llama-3.1` and Layer Normalization (Ba et al., 2016) for `aya-23`. We apply greedy

---

[3]We count simplified and traditional Chinese as separate; thus we say that `aya-23` supports 24 languages. Thai is the only language supported by `llama-3.1` but not `aya-23`.

decoding with this distribution to produce text output $O(l_i)$ from the intermediate layer l_i.

Logit lens only operates at the token level; however, terms in our dataset may be multi-token. We iteratively append the most probable token of the final layer output (standard greedy decoding) to x and apply a logit lens at every decoding step, enabling us to decode multi-token outputs from intermediate layers.

Defining translation loss Given a metric M which measures the quality of final model outputs for our task, suppose that we can measure the quality of task-solving at intermediate layers with a comparable metric M'. We then define *translation loss* TL for a given source word x and reference $\mathbf{y}$,

$$TL(x) := \max_{i<L}[M'(O(l_i), \mathbf{y})] - M(O(l_L), \mathbf{y}), \tag{1}$$

where $O(l_i)$ is the text output at model layer l_i, and L is the total number of layers. Aggregating instance level TL over the dataset $\mathcal{D}$ yields the dataset level translation loss:

$$TL(\mathcal{D}) = \sum_{x \in \mathcal{D}} TL(x)$$

The first term in Equation 1 measures task-solving success: we consider the task-solving performance at the best intermediate (non-final) layer for a given input. The second term measures final success at output layer l_L, i.e. the success of the *task-solving→translation* pipeline. Thus $TL(\mathcal{D})$, the difference between these, measures performance loss due to failed (internal) translation.

Defining TLP for word translation Our word translation dataset contains a set of equivalents $\{y^t\}$ per concept (source word x) per target language t. The metric $M(\hat{y}, \mathbf{y} = \{y^t\}) \in \{0, 1\}$ for measuring final accuracy is a binary exact match metric with any of the target language reference translations. We consider the model to have task-solved successfully at a given layer if the layer output expresses the semantics of the input in *any* non-source language. Thus,

$$M'(O(l_i), \mathbf{y}) = \max_{t \in T, t \neq s} M(O(l_i), \mathbf{y} = \{y^t\})$$

where T is the set of all languages and s is the source language. Given that the binary accuracy metric straightforwardly allows us to compute total failure as the number of source words with failed
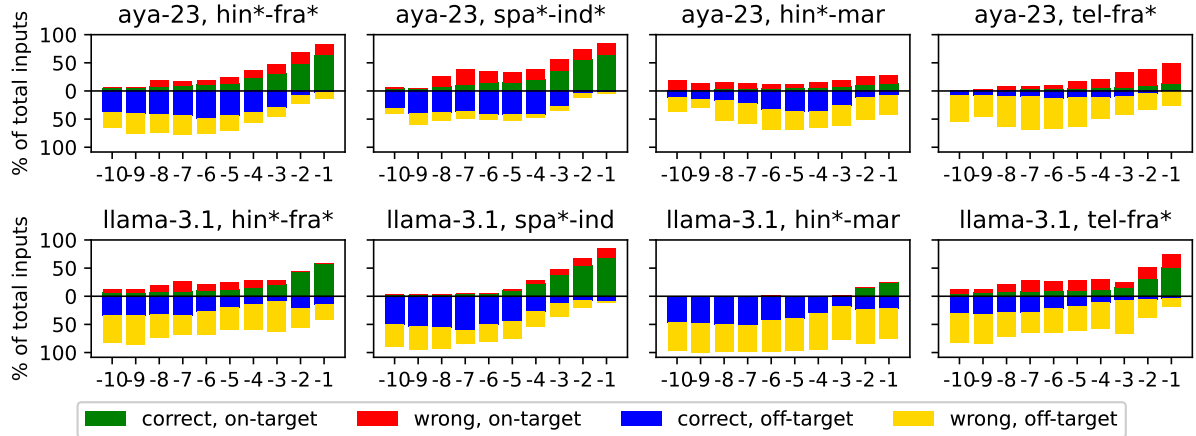
Figure 2: Each plot shows the percentage of on-target correct (◫) and incorrect (■) outputs in the *top half*, and off-target correct (◪) and incorrect (▨) outputs in the *bottom half*. We show this for the last 10 layers of `aya-23` and `llama-3.1`, for all outputs with a reliable LID tag. * supported language. We observe the *task-solving* stage with initially high accurate but off-target outputs (◪), followed by the *translation* stage, where the models transitions to on-target outputs (◫+■). We see that the translation stage is successful for French and Indonesian (HRLs) with high final on-target accuracy (◫), but fails for Marathi (LRL). For a low-resource source language like Telugu, *task-solving* itself may fail, as with `aya-23`, with low off-target accuracy (◪).

final translations d_F, we can now compute translation loss proportion (TLP):

$$TLP = TL(\mathcal{D})/d_F \qquad (2)$$

Note that in this setup, the complement $1 - TLP$ gives us task-solving failure as a fraction of total failure. Thus, computing TLP allows us to compare the proportion of blame that either stage of the pipeline bears for total final failure.

Layerwise analysis For each layer output, we are interested in its task-solving accuracy, measured by the exact match metric M', and its language. We use our dataset language tag for the layer output in case of a match against our listed language equivalents. For inaccurate outputs, we perform language ID on the layer output with the NLLB language ID model ([NLLB Team et al., 2022](#)).[4] We only retain the tag if it belongs to our set of target languages, which includes all languages supported by the model i.e. plausible intermediate languages, and discard it if not. This is because intermediate layer output text may be gibberish or malformed, in which case we expect the language ID model to return a noisy random tag. See [Appendix E](#) for a manual evaluation of the reliability of the resulting tags. We track accuracy and language of output for the last ten layers of each model.[5]

4 Results and Discussion

4.1 Visualizing the pipeline

In [Figure 2](#), we plot on-target and off-target accuracy for four language pairs over all outputs for which we found a reliable language tag. As targets, we include French and Indonesian as representative HRL targets, and Marathi as an example of an unsupported LRL target. We also include all three source languages. We find the patterns discussed below hold generally for all language pairs; see [Appendix F](#) for more examples.

We first observe the effect of the source language. While both models show high off-target accuracy with Hindi and Spanish as sources, `aya-23` behaves differently with respect to Telugu: regardless of target, it yields low intermediate or final accuracy. This indicates `aya-23` simply fails to comprehend Telugu task inputs, resulting in failed task-solving even for HRL targets such as French. This is less true of `llama-3.1`, which shows similar patterns with Telugu as source as with Hindi and Spanish.

We then look at the effect of the target language. We consistently observe high off-targetness in middle layers, including for Spanish and other supported or high-resource languages, with considerable off-target accuracy, which subsequently decreases as the model pivots to the desired language. For supported or HRL targets, there is a

---

[4][huggingface.co/facebook/fasttext-language-identification/](#)

[5]Preliminary experiments show that including earlier intermediate layers does not affect measurements of task-solving

accuracy. Logit lens is also unreliable for early layers ([Belrose et al., 2023](#)).

corresponding increase in on-target outputs and accuracy in later layers, indicating successful translation. Note that while HRLs like Indonesian may not be supported by the LLM (`llama-3.1`), we still see similar patterns for these languages as for supported HRLs, indicating that language resourcedness and presumably its presence in pre-training corpora play an important role in this regard. On the other hand, for Marathi and other unsupported or low-resource languages, we simply observe a total drop in correct (on-target or off-target) intermediate outputs as the model moves away from off-target equivalents but also fails to translate correctly, resulting in poor final outputs. *This provides intuition for the translation barrier for LRL generation with LLMs.* We note that languages pairs with English as a target behave differently from the above, with on-targetness and accuracy emerging early and continuing until the final layer. This is not surprising, given the dominance of English in model training data.

The consistent nature of these plots, with off-target accuracy in middle layers (successfully or unsuccessfully) converted to on-target accuracy, can be interpreted as the model first "task-solving", or realizing correct answer concepts in some language(s), and then converting or translating those to the intended language. We find that 91.6% and 92.6% of all on-target final correct answers for all language pairs are also identified as correct at some intermediate layer in a *non-target* language for `llama-3.1` and `aya-23` respectively (see Appendix F for language-wise breakdowns). *This, along with the above visualizations, shows evidence of a task-solving→translation pipeline in the model.* This also validates our technique of observing intermediate accuracy.

4.2 Characterizing task-solving languages

The above observations raise the question of what language(s) task-solving occurs in, referred to *task-solving languages*, and whether these differ by language pair. In order to investigate this, we first identify the task-solving layers of the model by locating the switch between the *task-solving* and *translation* stages of the generation pipeline, and then study the distribution of languages in the task-solving layers.

Layer of switch We use the proportion of the target language among all accurate layer outputs to track the switch between the two stages, since this
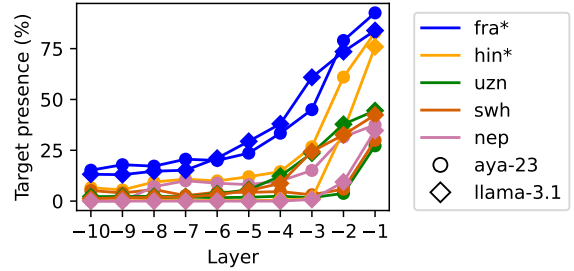

Figure 3: % of target language presence among *accurate* layer outputs for 5 target languages, averaged over source language. * supported language. This stays low for middle layers, indicating that accurate answers are largely off-target, and increases in final layers, indicating *translation* to the target language.

will be high in the *translation* stage, and plot this for five languages in Figure 3 for both models.[6]

In general, both models show a significance increase in target language presence around the last four layers. We imagine that this is model- and task-dependent. We also compute the specific layer with the maximum increase in target language presence from the previous layer, and find that this ranges -1 to -4 over different language pairs (see Appendix G for details). We found no correlation between the layer of switch and accuracy at the final layer over all language pairs.

Distribution over task-solving languages As per the above, we consider all layers up to the fourth-last layer as *task-solving* layers,[7] and look at the distribution over languages for all identified intermediate correct (off-target) answers in these layers. We show the aggregate distribution over all language pairs in Figure 4 for both models.

We see that English dominates the intermediate representations for both models, although considerably more so for `llama-3.1`. This is expected, given that `aya-23` is much more multilingual than `llama-3.1`. However, while previous work such as (Wendler et al., 2024; Etxaniz et al., 2024; Schut et al., 2025) focus exclusively on English as the intermediate representation language, we find that it only accounts for 30.7% and 48.5% of correct intermediate layer outputs for `aya-23` and `llama-3.1` respectively, with the rest of the distri-

---

[6]See Appendix H for complete plots. We also look at the proportion of target language outputs regardless of accuracy by layer; this results in similar trends.

[7]This is a conservative heuristic approximation: as shown in Figure 3, the switch between the stages is fairly fluid around the last few layers. Results look similar for surrounding choices of layer.
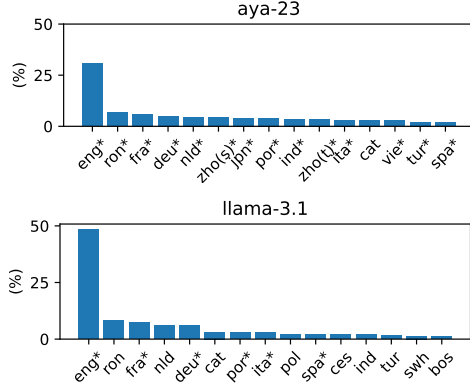
Figure 4: Distribution over *task-solving languages*, i.e. the languages of correct intermediate layer outputs, aggregated over all source-target language pairs for `aya-23` (top) and `llama-3.1` (bottom). We show the top 15 task-solving languages, covering 85% and 97% of probability mass of the distribution for `aya-23` and `llama-3.1` respectively. * supported language.

bution spread over other high-resource languages, with a tail of LRLs.[8] *While this shows that multilingual LLMs are undeniably English-centric, it also demonstrates some extent of multilinguality, and raises questions about what information might be represented in other languages.*

We therefore also look at this distribution conditioned on target language (averaged over source languages), and find that it looks similar at middle layers, showing an increase of target language presence near final layers. (See Appendix H for details.) *Thus, we find that the task-solving stage of our demonstrated pipeline occurs in an English-dominant mix of supported and high-resource languages, and that this mix of languages is largely agnostic of the target language in middle layers.*

4.3 Quantifying the translation barrier

Figure 2 suggests that the model is often able to solve the task, but fails to generate / translate correctly into LRLs. We quantify this effect by computing: (a) task-solving success or intermediate accuracy (the first term in Equation 1): the percentage of task outputs that were correct at *some* intermediate layer in *some* language (b) final success (the second term in Equation 1): on-target accuracy at the final layer, and (c) TLP, as per

---

[8]Note that we only compute this distribution over our considered 36 languages. These include all model-supported languages: we observe that these constitute the bulk of the probability mass. Our experiments indicate that including other unsupported languages will largely only modify the long tail of the distribution.

Equation 2: the proportion of total failure that can be attributed to translation failure.

First, we look at these three quantities averaged over all target languages for each source language in Table 1. We see that intermediate accuracy is much higher than final accuracy in all cases, with much less variance, indicating that the former is stabler and less dependent on the target language than the latter. This ties in with our understanding of the relatively target-language agnostic task-solving step, followed by a target-language dependent translation step. *Overall, translation loss accounts for a high percentage of total failure (more than 50% in all cases except one). this lends evidence to the translation barrier hypothesis.*

	Src.	Final Acc.	Int. Acc.	TLP
aya -23	spa*	42.0 ± 28.1	84.6 ± 7.2	71.6 ± 7.3
	hin*	37.4 ± 24.3	76.1 ± 7.2	59.6 ± 8.4
	tel	9.5 ± 13.0	25.3 ± 11.3	17.5 ± 4.0
	Avg	**29.6**	**62.0**	**49.6**
llama -3.1	spa*	39.8 ± 29.4	84.2 ± 9.8	68.0 ± 18.3
	hin*	33.7 ± 23.8	75.2 ± 7.8	58.2 ± 15.3
	tel	31.6 ± 21.1	69.4 ± 7.8	52.1 ± 12.6
	Avg.	**35.0**	**76.3**	**59.4**

Table 1: Mean and std. dev. of final accuracy ("final acc."), intermediate accuracy ("int. acc."), and TLP (all in %), over all target languages for each source language, for both models. * supported language. We see that intermediate accuracy is higher than final accuracy, with much less variance, indicating that the former is stabler and less dependent on the target language than the latter.

We also see that final and intermediate accuracy are roughly similar for both models, with the exception of for Telugu, for which `aya-23` is much worse. This is perhaps surprising given than `aya-23` is more multilingual than `llama-3.1`, supporting 24 as opposed to just 8 of our 36 target languages.

Role of translation failure Next, we look at intermediate accuracy, final accuracy, and TLP by language pair and model. We show these results for 18 target languages against each source language for both models in Figure 5, with the complete results for all 36 target languages in Appendix J. We find that translation failure dominates total failure ($> 50\%$) for 65% and 78% of all language pairs for and `aya-23` and `llama-3.1` respectively. This provides direct evidence for the translation barrier hypothesis for individual language pairs.

We further find distinct patterns for high-resource versus low-resource target languages. We observe that intermediate accuracy stays high even
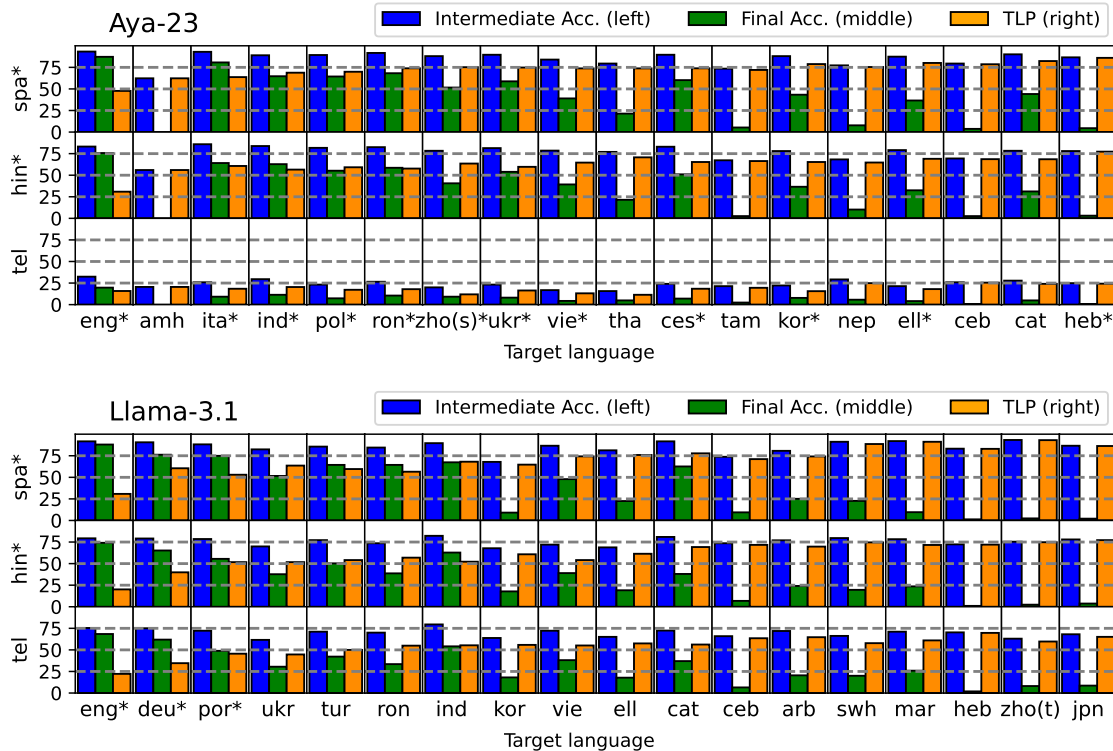
Figure 5: Intermediate accuracy (■), final accuracy (■), and TLP (■, Equation 2) for `aya-23` and `llama-3.1`, sorted in ascending order of mean TLP over source language for 18 target languages, selected to cover the range of mean TLP. *: supported language. While intermediate accuracy (■) is high even for LRLs (Cebuano-ceb, Nepali-nep), final accuracy (■) is high for supported HRLs (Portuguese-por, German-deu), but drops considerably for LRLs. TLP (■) is high for most target languages, and especially for low-resource languages. See expanded figure including all target languages in Appendix J.

for low-resource unsupported target languages (like Nepali and Hebrew) for Hindi and Spanish as sources: in line with our previous observations, this indicates that the model is able to correctly task-solve even for low-resource targets. In contrast, and as expected, we observe that final accuracy is high for supported target HRLs (like German, Italian, and Portuguese, for both models) but drops drastically for low-resource target languages (like Tamil and Swahili), some of which are supported, such as for Greek with `aya-23`. Script and language family may naturally also contribute to this performance drop (e.g. `llama-3.1` performs worse for Japanese and Arabic than for Romanian). High intermediate and final accuracy as in the first case (HRLs) means that TLP is relatively low: a larger part of total failure comes from task-solving errors, and the model is more capable of generating successfully in the case of task-solving success. However, high intermediate accuracy and low final accuracy as in the second case (LRLs) indicates that translation loss accounts for a much bigger percentage of total error, going up to 82.3% for `spa-cat` with

`aya-23`, and 91.1% for `spa-mar` with `llama-3.1`.

Leaving aside language pairs with English as a target, or with Telugu as a source, translation failure dominates total failure ($TLP > 50\%$) for 100% and 90% of language pairs for `aya-23` and `llama-3.1`. English is a special case: given that it dominates the internal representations of these models, generation into English may have a minimal or absent translation stage, meaning that task-solving success largely determines final success. We discuss the case of Telugu as a source below.

The above observations indicate that translation loss plays an outsized role in final failure even for supported target languages, and an even bigger role for low-resource or unsupported languages. These results validate the translation barrier hypothesis for the word translation task.

Evidence for entanglement in the task-solving stage The above discussion generally supports our understanding of target-language-agnostic *task-solving* stage followed by a target-language-specific *translation* stage. However, we note the

caveat that there is a clear disparity between intermediate task-solving accuracy across target languages, with the largest difference of 32.9 percentage points between spa-eng and spa-amh with aya-23, as well as differences between supported languages (12 percentage point difference between hin-eng and hin-jpn with aya-23). *This indicates some level of entanglement between task-solving success and language of generation, meaning that the intended language of generation affects the model's ability to task-solve to some extent.* This hints that even in the hypothetical case of no translation loss, end-to-end processing for low-resource or unsupported languages would still fall behind that for HRLs due to imperfect transfer of task-solving abilities to LRLs. This finding calls for an evaluation of the future of end-to-end multilingual generation with LLMs on a larger scale of languages, with one alternative being to invest in better explicit output-side LLM+MT cascades.

Impact of source language We see that for both models, mean intermediate and final accuracy as well as mean TLP are highest for Spanish, then Hindi, then Telugu (Table 1). It is unsurprising that intermediate accuracy and therefore final accuracy depend on the source language: task-solving success is naturally conditioned on comprehension of the task inputs. *These findings justify exploring an input-side MT+LLM cascade for different language task inputs even for supported languages.*

This also explains why TLP is lowest for Telugu: with low intermediate accuracy, translation loss can only occur on very few inputs as a percentage of the total failure, which is dominated by failed task-solving cases. For example, since task-solving fails for 74% of total inputs for Telugu-Cebuano with aya-23, task-solving failure dominates total failure even though the translation stage also fails on nearly all correct intermediate outputs.

4.4 Case study: Effect of scale

We repeat our experiments for 4 language pairs on Llama-3.1-70B-Instruct and show the effect of using a larger model on TLP in Figure 6. We find that the $70B$ model shows similar intermediate accuracy on average as the $8B$ model, but consistently higher final accuracy, or capability to produce the target language, resulting in lower TLP. However, absolute TLP values remain high. These preliminary results indicate that the translation barrier may continue to be a problem at scale.
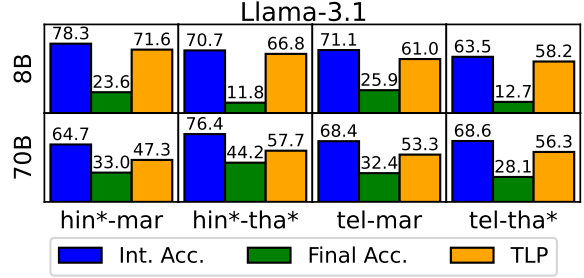

Figure 6: Intermediate accuracy (■), final accuracy (■), and TLP (■, Equation 2) for llama-3.1-8B (top) and llama-3.1-70B (bottom). *: supported language. Final accuracy (■) is higher for the latter. TLP (■) is lower but remains high in absolute terms.
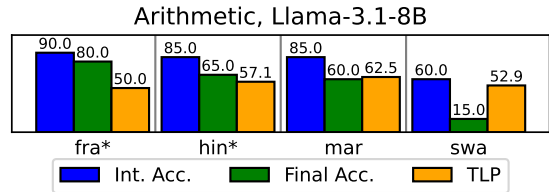

Figure 7: Intermediate accuracy (■), final accuracy (■), and TLP (■, Equation 2) for llama-3.1-8B for an arithmetic task. Intermediate acc. stays high, but final accuracy is lower for LRLs, leading to high TLP.

4.5 Case study: Arithmetic Task

We conduct a small case study to investigate the translation barrier on an arithmetic task. We used 20 word arithmetic questions (such as "Eight times two equals?"), with numeral word answers, and prompted the model to respond in a target language, for 4 variously resourced languages. We explore setting the prompt language to the intended target language as well as to English, and find similar behaviour for both settings. We manually examine the outputs and observe the task-solving→translation pipeline, with the correct solved answers in HRLs in intermediate layers, with a switch to the target language in late layers. (See Appendix K for examples.) See Figure 7 for intermediate and final accuracy, and TLP. We observe high TLP especially for lower-resourced languages; these results are consistent with our findings on word translation. We invite work in better understanding the translation barrier for other tasks and model sizes.

5 Related Work

While multilingual LLMs are trained to perform well for several languages (Lin et al., 2022; Workshop et al., 2023; Aryabumi et al., 2024), their pre-

training data is dominated by English (Xue et al., 2021; Chung et al., 2023; Li et al., 2025a). Etxaniz et al. (2024) and Zhao et al. (2024) show that explicit cascading with MT often outperforms end-to-end multilingual inference. Further, works such as Wendler et al. (2024) and (Schut et al., 2025) apply logit lens to intermediate layers and observe that multilingual LLMs "think" in English, or that intermediate representation spaces of these models lie close to English. This is reinforced by works studying neuron activations for multilingual processing (Foroutan et al., 2022; Tang et al., 2024; Kojima et al., 2024), which find language-specific neurons only in later model layers. Concurrent works to ours such as Lu et al. (2025) and Wang et al. (2025) demonstrate that late layer translation failure affects cross-lingual consistency and accuracy for factuality recall. Our work contributes to this discourse by formalizing and quantifying the translation barrier for a core generation task over a range of languages, with a focus on its impact on low-resource languages.

6   Conclusion

We formalize the **translation barrier hypothesis** for poor quality multilingual generation with LLMs. Working with a minimal generation task for 108 language pairs, we visualize a *task-solving→translation* pipeline for multilingual generation, demonstrating its failure for low-resource languages. We then quantify the role of the translation barrier as an important culprit for low final performance, discussing its relevance for differently resourced source and target languages. Our findings contribute insights for future strategies in multilingual generation with LLMs.

Limitations

Scope   This work only explores the translation barrier hypothesis for a word translation task. It may be more difficult to assess task-solving success for more complex sequence-level tasks such as sentence-level machine translation, summarization, or open-ended instruction following in a principled manner using artifacts such as intermediate layer representations, since the nature of correct intermediate answer concepts could be less clear to ascertain. There may also be additional complexities to take into account when extending our setup to these tasks: for example, whether task-solving occurs "globally" over many parts or subtasks of

a single input or "locally" for different fragments of the input, or the manner in which the target language syntax interacts with the ordering of the concepts in the task-solving stage. Thus, our hypothesis may be difficult to verify for other tasks.

We clarify that our findings regarding the role played by the translation barrier may be dependent on the nature and complexity of the task, as well as the usage of strategies such as task- or language-specific finetuning or inference-time techniques such as in-context learning to boost multilingual performance.

While we do not claim that the statement of the hypothesis holds for all tasks and settings, we believe that the line of investigation in our work has important implications for future directions to improve the multilingual capabilities in LLMs, especially regarding the potential of end-to-end versus cascaded or modular approaches for the same. Our goal is to invite discussion and further investigation of the translation barrier, as well as the most viable strategies for its mitigation.

Models and languages   We conduct our experiments on two mainstream multilingual models; our findings should be verified on a range of different models. Similar to previous work whose insights we build upon, we also only work with decoder-only models. Our posited internal pipeline may look different for encoder-decoder models such as Aya-101 (Üstün et al., 2024).

Further, our findings are restricted to the 36 target languages and 3 source languages that we test on, and may differ for other languages and language families. In this work, we focus on mid- to low-resource range languages, including national languages like Swahili and Thai. Future work may look at extending our setup and exploring the translation barrier for lower-resourced languages.

Finally, we use English as our prompt language regardless of the source language (i.e. the language of the term to be translated); this is as per work that says that this performs better (Dey et al., 2024). Testing the effect of different prompt languages may add further nuance to our findings.

Ethics Statement

We do not anticipate any negative consequences of this work, which is intended to diagnose poor multilingual capabilities in large language models. We acknowledge that different language communities may have various and diverse needs from language

technologies that may diverge from the points of focus of mainstream NLP, and not necessarily include LLM technologies (Bird, 2020). All the models we used are publicly available for research purposes: Aya-23-8B has a CC-BY-NC license, and the Llama family of models have a Llama 3.1 Community License Agreement, permitting research use. We will release all the artifacts created with the non-anonymized version of this paper. We used AI agents only for coding assistance (e.g. GitHub Copilot).

Acknowledgments

References

Sotiris Anagnostidis and Jannis Bulian. 2024. How susceptible are llms to influence in prompts? *Preprint*, arXiv:2408.11865.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Kelly Marchisio, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *Preprint*, arXiv:1607.06450.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Steven Bird. 2020. Decolonising speech and language technology. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3504–3519, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2022. Language contamination helps explains the cross-lingual capabilities of English pretrained models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3563–3574, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, and Pascale Fung. 2024. LLMs are few-shot in-context low-resource language learners. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 405–433, Mexico City, Mexico. Association for Computational Linguistics.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. 2023. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. *Preprint*, arXiv:2304.09151.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Krishno Dey, Prerona Tarannum, Md Arid Hasan, Imran Razzak, and Usman Naseem. 2024. Better to ask in english: Evaluation of large language models on english, low-resource and cross-lingual settings. *arXiv preprint arXiv:2410.13153*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. Do multilingual language models think better in English? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564, Mexico City, Mexico. Association for Computational Linguistics.

Negar Foroutan, Mohammadreza Banaei, Rémi Lebret, Antoine Bosselut, and Karl Aberer. 2022. Discovering language-neutral sub-networks in multilingual language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7560–7575, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *arXiv preprint arXiv:2302.09210*.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is ChatGPT A Good Translator? Yes With GPT-4 As The Engine. *Preprint*, arxiv:2301.08745.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6919–6971, Mexico City, Mexico. Association for Computational Linguistics.

Chong Li, Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2024. Improving in-context learning of multilingual generative language models with cross-lingual alignment. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8058–8076, Mexico City, Mexico. Association for Computational Linguistics.

Tianjian Li and Kenton Murray. 2023. Why does zero-shot cross-lingual generation fail? an explanation and a solution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12461–12476, Toronto, Canada. Association for Computational Linguistics.

Tianjian Li, Haoran Xu, Weiting Tan, Kenton Murray, and Daniel Khashabi. 2025a. Upsample or upweight? balanced training on heavily imbalanced datasets. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3325–3343, Albuquerque, New Mexico. Association for Computational Linguistics.

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2025b. Language ranker: A metric for quantifying llm performance across high and low-resource languages. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27):28186–28194.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, and 2 others. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chaoqun Liu, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu, and Lidong Bing. 2024. Is translation all you need? A study on solving multilingual tasks with large language models. *arXiv preprint arXiv:2403.10258*.

Danni Liu and Jan Niehues. 2025. Middle-layer representation alignment for cross-lingual transfer in fine-tuned llms. *Preprint*, arXiv:2502.14830.

Meng Lu, Ruochen Zhang, Ellie Pavlick, and Carsten Eickhoff. 2025. Paths not taken: Understanding and mending the multilingual factual recall pipeline. *arXiv preprint arXiv:2505.20546*.

Kelly Marchisio, Wei-Yin Ko, Alexandre Bérard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6653–6677.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2214–2231, Online. 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, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

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, and 20 others. 2022. No language left behind: Scaling human-centered machine translation.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. lesswrong.com/posts/AcKRB8wDpdaN6v6ru. Accessed: 2025-03-29.

Nathaniel Robinson, Perez Ogayo, David R. Mortensen, and Graham Neubig. 2023. ChatGPT MT: Competitive for high- (but not low-) resource languages. In *Proceedings of the Eighth Conference on Machine Translation*, pages 392–418, Singapore. Association for Computational Linguistics.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. 2025. Do multilingual llms think in english? *arXiv preprint arXiv:2502.15603*.

Lingfeng Shen, Weiting Tan, Sihao Chen, Yunmo Chen, Jingyu Zhang, Haoran Xu, Boyuan Zheng, Philipp Koehn, and Daniel Khashabi. 2024. The language barrier: Dissecting safety challenges of LLMs in multilingual contexts. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2668–2680, Bangkok, Thailand. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Hale Sirin, Sabrina Li, and Thomas Lippincott. 2024. Detecting structured language alternations in historical documents by combining language identification with Fourier analysis. In *Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024)*, pages 46–50, St. Julians, Malta. Association for Computational Linguistics.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Mingyang Wang, Heike Adel, Lukas Lange, Yihong Liu, Ercong Nie, Jannik Strötgen, and Hinrich Schütze. 2025. Lost in multilinguality: Dissecting cross-lingual factual inconsistency in transformer language models. *arXiv preprint arXiv:2504.04264*.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, and 375 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

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. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Biao Zhang and Rico Sennrich. 2019. Root Mean Square Layer Normalization. In *Advances in Neural Information Processing Systems 32*, Vancouver, Canada.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Shaolei Zhang, Qingkai Fang, Zhuocheng Zhang, Zhengrui Ma, Yan Zhou, Langlin Huang, Mengyu Bu, Shangtong Gui, Yunji Chen, Xilin Chen, and Yang Feng. 2023. Bayling: Bridging cross-lingual alignment and instruction following through interactive translation for large language models. *arXiv preprint arXiv:2306.10968*.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. *arXiv preprint arXiv:2402.07827*.

Language	Language (Sub)Family	FLORES+ code	Supported by Aya	Supported by Llama	Resource level
English	Germanic	eng_Latn	✓	✓	∼7.0M
Cebuano	Austronesian	ceb_Latn	-	-	∼6.1M
German	Germanic	deu_Latn	✓	✓	∼3M
French	Romance	fra_Latn	✓	✓	∼2.7M
Dutch	Germanic	nld_Latn	✓	-	∼2.2M
Russian	East Slavic	rus_Cyrl	✓	-	∼2.1M
Spanish	Romance	spa_Latn	✓	✓	∼2M
Italian	Romance	ita_Latn	✓	✓	∼1.9M
Polish	West Slavic	pol_Latn	✓	-	∼1.7M
Chinese (Simplified)	Sinitic	zho_Hans	✓	-	∼1.5M
Chinese (Traditional)	Sinitic	zho_Hant	✓	-	(see above)
Japanese	Japonic	jpn_Jpan	✓	-	∼1.5M
Ukranian	East Slavic	ukr_Cyrl	✓	-	∼1.4M
Vietnamese	Austroasiatic	vie_Latn	✓	-	∼1.3M
Arabic	Semitic	arb_Arab	✓	-	∼1.3M
Portuguese	Romance	por_Latn	✓	✓	∼1.2M
Persian	Indo-Iranian	pes_Arab	✓	-	∼1M
Catalan	Romance	cat_Latn	-	-	∼800k
Indonesian	Austronesian	ind_Latn	✓	-	∼700k
Korean	Koreanic	kor_Hang	✓	-	∼700k
Turkish	Turkic	tur_Latn	✓	-	∼600k
Czech	West Slavic	ces_Latn	✓	-	∼600k
Romanian	Romance	ron_Latn	✓	-	∼500k
Hebrew	Semitic	heb_Hebr	✓	-	∼400k
Uzbek	Turkic	uzn_Latn	-	-	∼300k
Greek	Hellenic	ell_Grek	✓	-	∼300k
Tamil	Southern Dravidian	tam_Taml	-	-	∼200k
Thai	Kra-Dai	tha_Thai	-	✓	∼200k
Hindi	Indic	hin_Deva	✓	✓	∼200k
Telugu	Southern Dravidian	tel_Telu	-	-	∼100k
Swahili	Niger-Congo	swh_Latn	-	-	∼100k
Marathi	Indic	mar_Deva	-	-	∼100k
Bosnian	South Slavic	bos_Latn	-	-	∼100k
Yoruba	Niger-Congo	yor_Latn	-	-	∼30k
Nepali	Indo-Aryan	nep_Deva	-	-	∼30k
Amharic	Semitic	amh_Ethi	-	-	∼15k

Table 2: All studied languages listed with model support details, with FLORES+ codes from NLLB Team et al. (2024). We report the number of Wikipedia articles for a language as a proxy for its resource level, with the anomaly of Cebuano Wikipedia being disproportionately large due to automatic creation.

A Languages

See Table 2 for the list of languages that we studied.

Language	Average Number of Equivalents in Lexica
English	7.5
Spanish	9.5
French	7.7
German	9.8
Russian	12.9
Japanese	6.0
Portuguese	8.6
Italian	7.4
Turkish	9.7
Korean	8.6
Arabic	6.8
Polish	10.3
Hindi	7.0
Hebrew	3.9
Ukranian	7.4
Chinese (Simplified)	1
Chinese (Traditional)	1
Vietnamese	6.8
Indonesian	9.0
Romanian	12.9
Dutch	8.2
Czech	6.5
Greek	2.7
Marathi	2.3
Swahili	1.7
Nepali	1.1
Tamil	1.9
Telugu	1
Uzbek	4.4
Catalan	1.4
Bosnian	5.5
Cebuano	1.3
Persian	5.4
Thai	9.4
Amharic	1
Yoruba	1.4

Table 3: All studied languages listed with average number of equivalents per source word per language.

B   Dataset Details

In order to curate the source concepts that we translate to every language, we prompted ChatGPT to give us 200 instances each of English nouns, verbs, adjectives and adverbs (totalling 800 words). We then filtered out any words that did not appear in the English NLTK WordNet (Bird and Loper, 2004; Miller, 1994), and manually vetted the remaining words to retain "general" concepts (i.e. excluding any proper nouns as well as highly specific cultural terms). We also checked for offensive words and did not encounter any. Finally, we sample 100 words per part-of-speech of the filtered words, and conduct all our experiments on the resulting 400 concepts.

We collect translations, including synonyms and alternate word senses, for these 400 concepts for each of the other 35 languages studied using Google Translate as mentioned in § 2. For average number of equivalent words per source word per language, see Table 3. We would like to note that the above method yielded only 1 equivalent per source word in some languages, like Chinese and Amharic and that this may have affected our results for these languages.

Manual evaluation   To evaluate the expanded alternative translation outputs, we sampled 100 words from two languages (Turkish and Polish) and manually evaluated the alternative translation groups for each source word for plausibility (on a scale of 1 to 4, 1 = "none of the words are plausible" and 4 = "all of the words are plausible") and for coverage (on a scale of 1 to 4, 1 = "none of the word senses are covered" and 4 = "all of the word senses are covered"). We report 3.9 and 3.4 for coverage, and 3.8 and 3.7 for plausibility, for Polish and Turkish respectively.

We further asked proficient speakers of Hindi, Marathi, Uzbek, Telugu, and Tamil, to go through 100 entries of our dataset, and report the number of entries with incorrect listed translations. Note that the last four languages are among the lowest-resourced in study. We report the resulting error rates in Table 4.

The consent of all annotators was obtained regarding the usage of the collected data for accuracy and coverage reporting of the lexica. We did not provide any monetary compensation for this annotation.

Language	Error %
Hindi	2%
Marathi	7%
Uzbek	0%
Tamil	1%
Telugu	2%

Table 4: Error rates in our dataset for 5 languages.

C   Model hyperparameters and prompting setup

Hyperparameters and compute   We use greedy decoding with a temperature of 0 and the official chat template of `aya-23` and `llama-3.1`, respectively. We run inference with logit lens on NVIDIA H100 NVL Tensor Core GPUs. This takes about 30 minutes to process the test set of size 400 with batch size 8 on a single GPU, for each of our 108 language pairs for each of `aya-23` and `llama-3.1`. We downloaded `aya-23` and `llama-3.1` from Huggingface Transformers (Wolf et al., 2020).

We additionally run experiments on 4 language pairs with `llama-3.1-70B`. This takes 45 minutes per language pair on 4 GPUs with a batch size of 8.

This does not include exploratory experiments and debugging.

Prompting   Given that LLM performance is sensitive to the wording of the prompt (Anagnostidis and Bulian, 2024), we try a few different prompts for the word translation task. We find that final performance differs slightly with different wordings of the instruction as well as the manner in which the target language is specified.

Here are examples of prompts we explored:

1. ```
   Translate the following word into English:

   gata
   ```

2. ```
   Translate the following word from spa_Latn to mar_Deva. Respond with a single word.

   Word: gata

   Translation:
   ```

3. ```
   Give a one-word translation of the following word from Spanish to Marathi.

   Word: gata

   Translation:
   ```

4. ```
   Translate the following word from Spanish to Marathi. Respond with a single word.

   Word: gata

   Translation:
   ```

We use prompt (4) for all our experiments, as performing well for both our models.

D   Examples of inputs and outputs

See Table 5 for examples of intermediate layer outputs for some inputs.

	Source word	Concept	Target Word	Int. Layer No.	Int. Layer Output	Language	Correct Concept?
Hindi → Turkish	जानवर	animal	hayvan	-5	동물lar	Korean	✓
Spanish → Uzbek	piso	floor	qavat	-6	زمین	Persian	✓
Telugu → Italian	క్లిఫ్	cliff	scogliera precipizio	-2	clic	English	✘
Spanish → Uzbek	lento	slow	sekin	-3	chậm	Vietnamese	✓

Table 5: Intermediate layer output examples that are in a language other than the input or the target language. The prompt with the source word is formatted as mentioned in Appendix C.

E   Reliability of the LangID

Recent work shows that neural language identification models can be performant even for mixed-script and noisy multilingual data (Sirin et al., 2024). To evaluate the language ID performance of the NLLB model (NLLB Team et al., 2022) over intermediate outputs, we sampled 100 source-target word pairs from two different language pairs, Hindi - Turkish and Spanish - German. Note that we discard labels that do not correspond to one of our 36 target languages as per § 3, and only report on-targetness or off-targetness in Figure 2 for inputs for which we can find a reliable label.

We evaluated the resulting language identification labels for two different layer outputs for each word pair. We found that all labeled samples had outputs that were recognizably in some language. We marked the LID as correct if it accurately identified the language of any of the words in the output, given that the outputs could be language-mixed. We found that the model was 80% accurate over intermediate layer outputs.
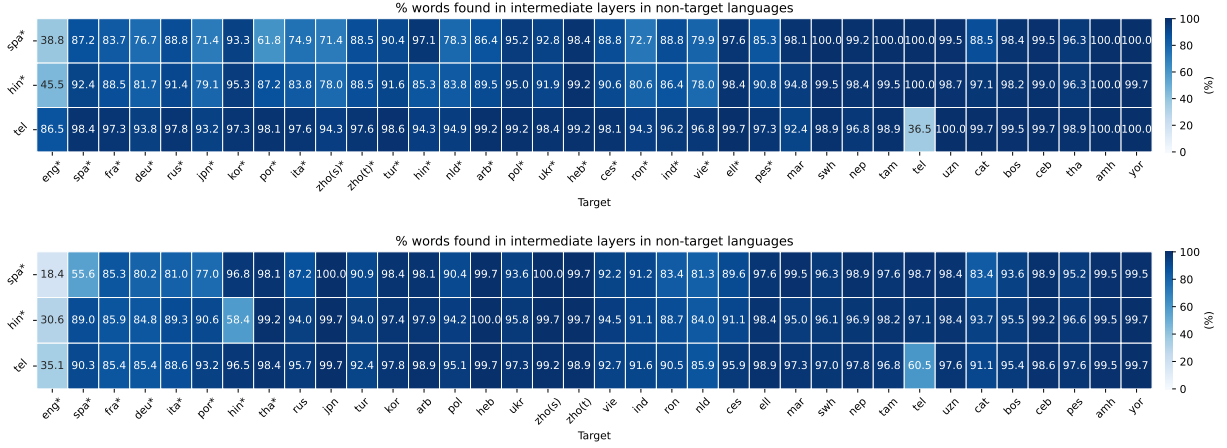
Figure 8: This shows the percentage of task inputs that had final correct answers for which we also correct intermediate layer outputs in a different (non-target) language for `aya-23` (top) and `llama-3.1` (bottom). This demonstrates that intermediate layers perform task-solving at some layer of abstraction.

F Evolution through layers for all languages

Demonstrating abstract task-solving We show the percentage of final answers that were identified as correct in some intermediate layer, in a *different language from the target*, in Figure 8 for both models. In general, we see that this percentage is high, although less so for HRL supported languages. This, in conjunction with the shape of the distribution over intermediate languages that are largely target-language agnostic as discussed in § 4 and Appendix H, demonstrates that intermediate layers perform task-solving at some layer of abstraction.

Layer-wise accuracy for other language pairs We choose five representative language pairs and show the evolution of on-target and off-target accuracy through various layers for each of them for `aya-23` and `llama-3.1` in Figure 9. These cover all three source languages, supported HRL and MRL languages, as well as unsupported languages.
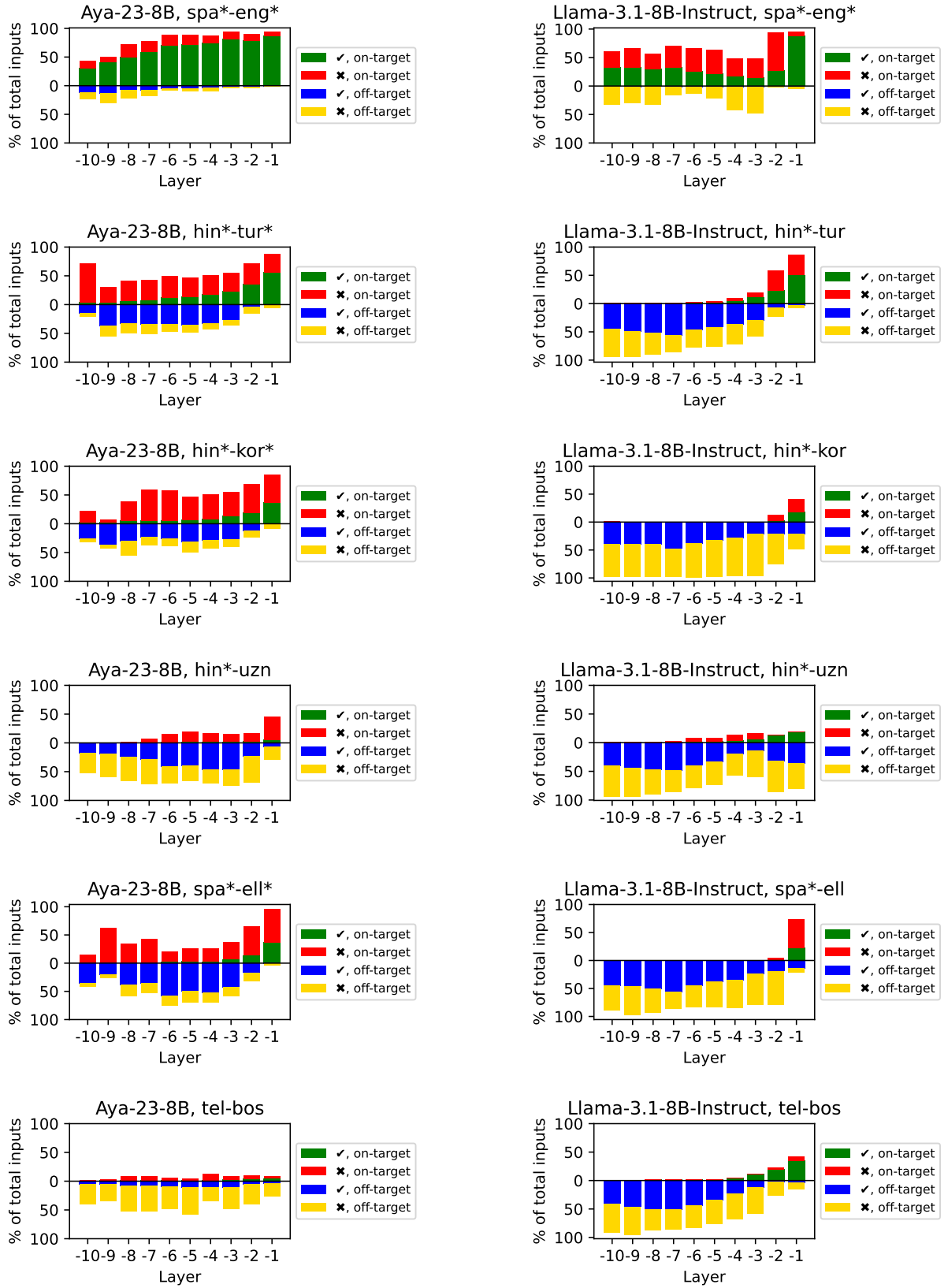
Figure 9: Layerwise analysis plots for five language pairs: `aya-23` (left) vs. `llama-3.1` (right) (extension of Figure 2
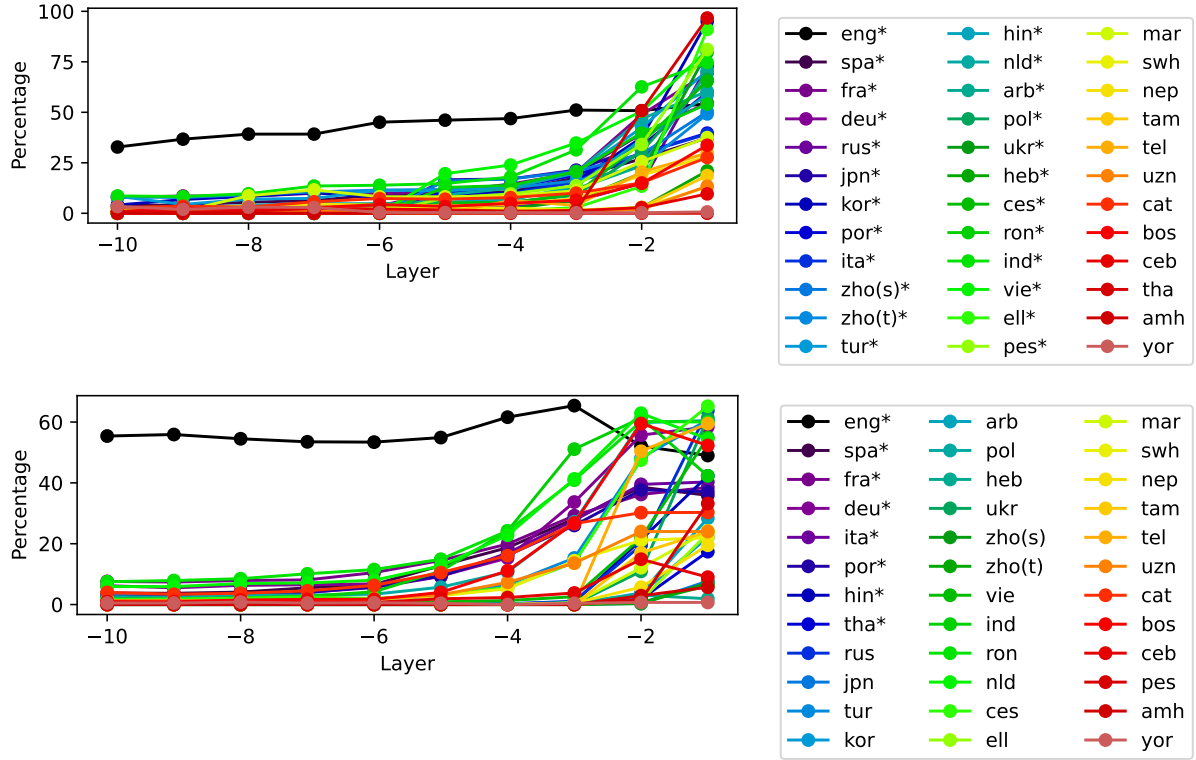
Figure 10: Target language presence by proportion of all languages for correct intermediate answers for `aya-23` (top) and `llama-3.1` (bottom) (complete version of Figure 3 for all target languages).

G Layer of switch for individual language pairs

See Figure 10 for a complete version of Figure 3 for all target languages.

In Figure 11, we show the layer with the maximum increase in target language presence for accurate outputs, in order to identify a particular layer of switch between the *task-solving* and *translation* stages. We exclude language pairs that have negligible final accuracy ($< 5\%$), since these effectively have no translation stage. We see the layer of switch is on or after the fourth-last layer for `aya-23` and on or after the third-last layer for `llama-3.1`, with `aya-23` often showing earlier switches than `llama-3.1`.
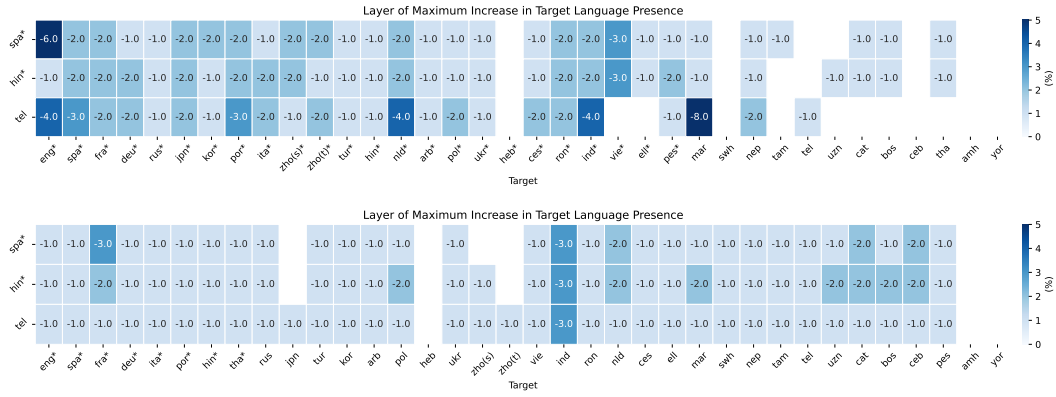
Figure 11: The layer with the maximum increase in target language presence for accurate outputs for `aya-23` (top) and `llama-3.1` (bottom). Only shown for language pairs that had $> 5\%$ final accuracy.

H Distribution of "task-solving" languages by target language

Given that there is a prior model-internal reasoning step before generation, we are interested in the language(s) in which this reasoning occurs. As we visualize in Figure 2 and Figure 9, there is a somewhat fluid shift to the target language at later layers; we can consider these the layers responsible for the translation / generation part of the pipeline. We see that this shift emerges around the fourth-last layer; we therefore consider the task-solving layers as all the layers up to the fourth-last layer. We show the distribution of correct intermediate equivalents over intermediate languages for each target language (averaged over source language) in Figure 12 for `llama-3.1` for these middle layers. In general, we see that this distribution looks similar over target languages, supporting the idea of model interlingua that is to a large degree agnostic of the target language. The plot looks similar for `aya-23` and also largely similar when using different layer cutoffs for what we consider as task-solving layers, with increasing target language presence as we move closer to the final layer.

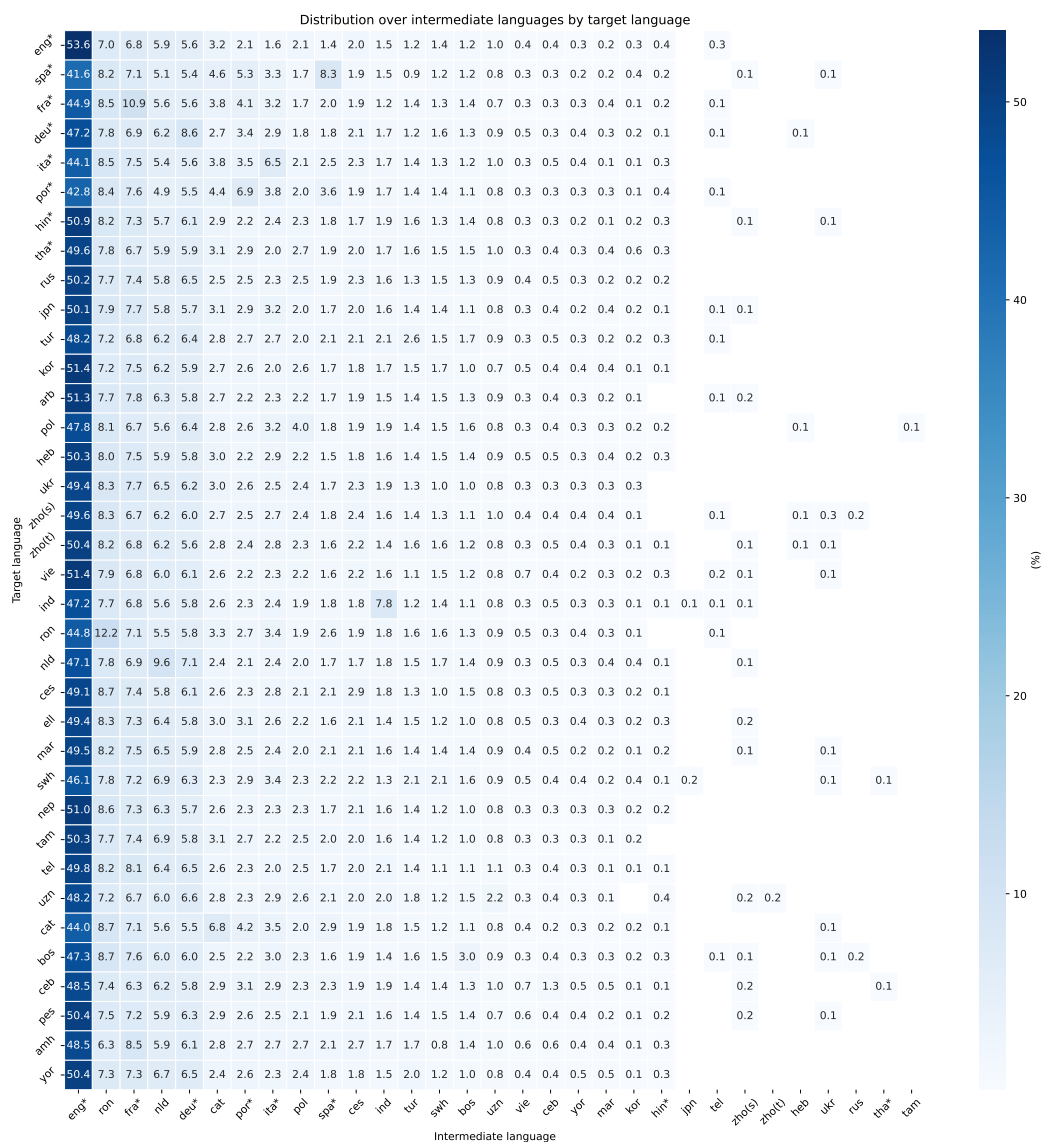Distribution over intermediate languages by target language

Figure 12: Distribution over intermediate languages for all correct (off-target) intermediate layer equivalents. This is an expanded version of Figure 4 by target language.
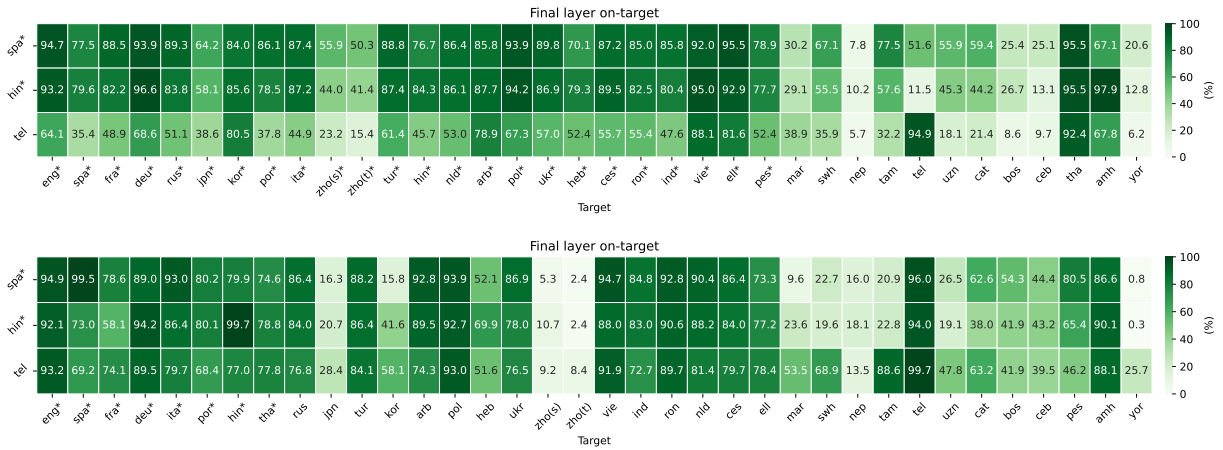
Figure 13: Percentage of outputs that were on-target for `aya-23` (top) and `llama-3.1` (bottom). * supported language

I On-targetness

We show the percentage of final layer on-target outputs in Figure 13 for both models. Predictably, this is higher for supported HRLs than for unsupported or low-resource languages. Note that these results are dependent on the accuracy of the language ID model, which may be easily confused on short spans of text.

J Results for all languages

We show the expanded version of Figure 5 in Figure 14 for aya-23 and Figure 15 for llama-3.1.
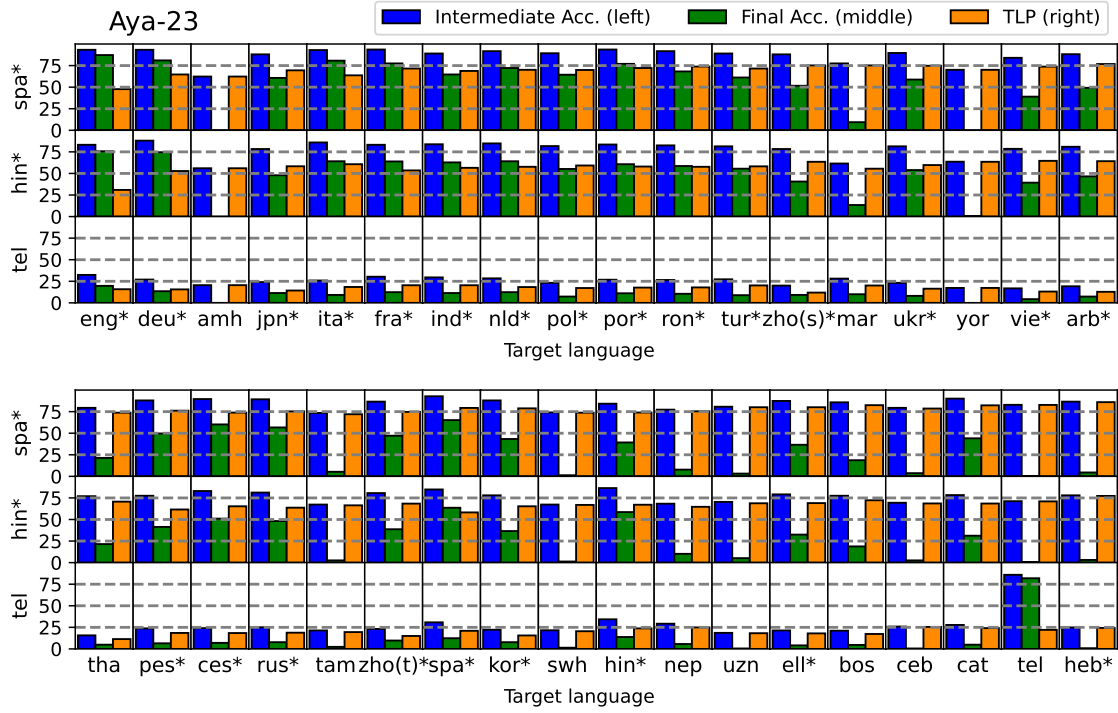

Figure 14: Intermediate and final accuracy, and TLP (Equation 2) for aya-23, sorted in ascending order of mean TLP (complete version of Figure 5, for all languages).
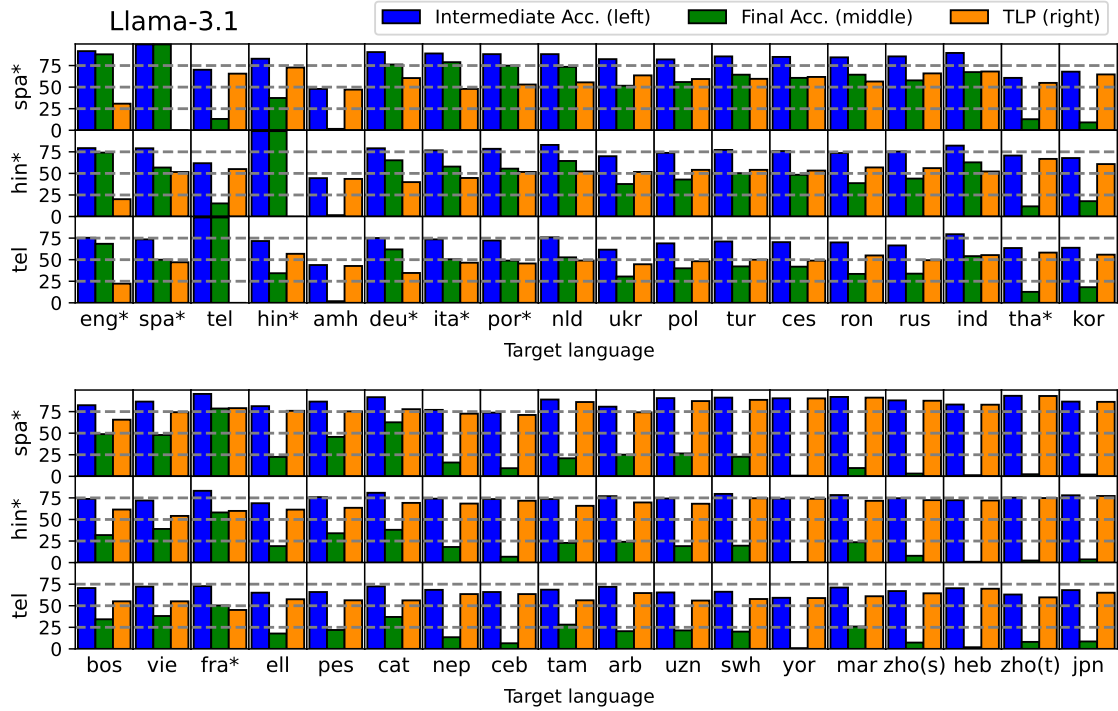

Figure 15: Intermediate and final accuracy, and TLP (Equation 2) for llama-3.1, sorted in ascending order of mean TLP (complete version of Figure 5, for all languages).

Example 1	
Input	Solve the following arithmetic and provide an answer in words in Marathi. Respond with only the answer in Marathi. Example: Input: Two plus three equals? Target language: Marathi Answer: पाच Now solve the following question: Input: Eight divided by two equals? Target language: Marathi Answer:
Correct Answer	चार
Layer 22	four, four, -four, six, isko, five, bốn, 四, fours, half
Layer 23	four, four, eight, 4, -four, six, five, fours, bốn, FOUR
Layer 24	four, eight, four, six, five, 四, eight, -four, FOUR, Four
Layer 25	four, four, six, five, eight, FOUR, seven, 四, -four, Four
Layer 26	four, four, 四, -four, Four, FOUR, _ four, five, cuatro, fours
Layer 27	four, four, 四, cuatro, tre, -four, FOUR, 4, tree, five
Layer 28	four, four, 四, Four, cuatro, 4, -four, 四, _ four, FOUR
Layer 29	quadr, qu, quat, नेट, 四, Four, fou, -qu, Qu, ने
Layer 30	四, Four, Four, नेट, च, four, quat, fou, cat, four
Layer 31	च, च, अ, प, Four, чет, quatre, नेट, quat, आ
Generated Text	चार

Table 6: Input and intermediate and final response for the word arithmetic task with Marathi. Intermediate correct responses are highlighted in blue.

K Arithmetic task

See examples of our input, intermediate responses, and the final generated responses in Table 6 and Table 7. We tried various prompting settings, including a zero-shot prompt. We found that both aya-23 and llama-3.1 tended to give sentence-long responses in those settings. The above format, i.e. 1-shot prompting with the example in the target language, worked best to constrain the answer.

We also tried a variant of this task where the input itself in provided in the target language. Our main observations were that even in this case, the intermediate layers exhibit answers in HRLs before switching to the target language. Further, this degrades performance for all languages, especially in Marathi and Swahili. This ties back to our discussion in § 4 about source language comprehensibility.

Example 2	
Input	Solve the following arithmetic and provide an answer in words in Swahili. Respond with only the answer in Swahili. Example: Input: Two plus two equals? Target language: Swahili Answer: nne Now solve the following question: Input: Four plus six equals? Target language: Swahili Answer:
Correct Answer	kumi
Layer 22	ĩa, eight, Banc, six, four, LAY, leDb, |, forth, ecies
Layer 23	eight, nine, ĩa, Banc, ogany, alus, 8, |, eight, six
Layer 24	enda, 254, mash, dere, ĩa, arton, ĩ, Kit, ọt, pend
Layer 25	ĩa, ười, ĩ, 熟, Mash, 254, ọt, enda, mash, chín
Layer 26	ĩ, ười, ısından, DataURL, arton, ĩa, ọt, .hashCode, έν, tham
Layer 27	inode, angu, ĩ, вос, ten, oni, chín, 楽, दस, ười
Layer 28	ten, 10, 十, Ten, TEN, Ten, _ten, ten, деся, 十
Layer 29	十, , ısından, 10, ten, दस, деся, десяти, 十, Ten
Layer 30	十, ten, еся, ısından, , olio, oria, десяти, दस,
Layer 31	t, n, ten, mb, m, s, z, k, mat, de
Generated Text	tano

Table 7: Input and intermediate and final response for the word arithmetic task with Marathi. Intermediate correct responses are highlighted in blue. We see that even though the correct answer is discovered in intermediate layers, the final answer is wrong.

L   Examples for sentence-level translation

While this work focuses on single word translation, we show examples of applying iterative logit lens as described in § 3 for a sentence-level translation task. We observe that intermediate layers exhibit code-switching behavior that interpolates between the correct target language and English. We invite future work in studying the translation barrier for sequence-level tasks.

L.1   Example 1

Source Sentence	Mes 3 minutes n'ont pas déjà commencé, n'est-ce-pas ?
Correct Translation	我的三分钟还没开始吧?
Layer 32 (Final Layer)	我的3分钟还没有开始，是吗?
Layer 29	我的3 minute还没有开始，是吗?
Layer 27	my three minutes chua chua开始吧 right吗?
Layer 25	my three minutes already chua开始吧 right吧?

L.2   Example 2

Source Sentence	Je veux dire, je suis déjà assez nerveuse comme ça.
Correct Translation	我是说，我已经够紧张的了
Layer 32 (Final Layer)	我的意思是，我已经很紧张了。
Layer 29	我的意思是我我already足够紧张了。
Layer 27	我mean是我我already cukup nervous了。
Layer 25	我already là, already already already nervous了。

L.3   Example 3

Source Sentence	De plus, au Canada, nous avons un formidable système de santé.
Correct Translation	在加拿大，我们有很好的医保制度
Layer 32 (Final Layer)	此外，在加拿大，我们有一个很棒的医疗系统。
Layer 29	此外，我们加拿大，我们拥有fantast好的的健康system。
Layer 27	此外，我们Canada我们我们有fantast好的的healthcare system。
Layer 25	此外，我们Canada we我们有superb好的的healthcare system。

L.4   Example 3

Source Sentence	Mais, tout d'abord, vous devez savoir 2 choses à mon propos
Correct Translation	Aber Sie müssen zuerst zwei Dinge über mich wissen.
Layer 32 (Final Layer)	Aber zuerst, Sie müssen 2 Dinge über mich wissen.
Layer 29	But zuerst müssen ihr müssen zwei2 things über mich wetenissen.
Layer 27	but first einmal you must know two things things about me know know.

M More Related Work

Cross-lingual Representations Parallel sentences exhibit highly similar hidden states in early layers of multilingual LLMs (Muller et al., 2021; Conneau et al., 2020), and fine-tuning solely on English tasks further strengthens this alignment (Li and Murray, 2023), enabling cross-lingual transfer. Moreover, the degree of representation similarity predicts performance in other languages (Zhang et al., 2023; Li et al., 2025b; Liu and Niehues, 2025; Li et al., 2024). Our work adds to this discourse by characterizing the task-solving languages of the model, and providing evidence of some degree of target-language agnosticity of the task-solving stage.