LLM Translation Locally with Small Models

Dante Paul Villalobos, Paulo Eduardo Carvalho Mansano

TU Berlin

dpvillal@uci.edu, p.mansano@edu.pucrs.br

1. Introduction

With the exponential growth of internet access and digital communication, the world is becoming increasingly connected. People are no longer limited by geographical boundaries when accessing information, consuming content, or interacting with others. However, one significant barrier remains: **language**. We believe that individuals should be able to understand their surroundings—whether navigating a foreign country, reading online content, or simply interacting with a multilingual environment—without being hindered by linguistic differences.

While numerous translation tools, such as Google Translate and DeepL, already exist and offer reasonable solutions for many high-resource languages, these tools often fall short in critical areas. **Low-resource languages** continue to struggle with poor translation quality, contextual misunderstandings, and inconsistent accuracy, even when processed by some of the most prominent existing platforms. Furthermore, most of these solutions rely heavily on cloud processing, making them impractical for real-time, offline use on mobile devices due to their computational demands and storage requirements.

In this context, **Large Language Models (LLMs)** present a promising alternative. We believe that, with focused effort, LLMs could evolve into a practical solution for overcoming the language barrier, even for less-resourced languages. However, achieving this vision would require significant work, from developing new datasets to refining training processes, and ultimately compressing models to a size suitable for efficient deployment on portable devices.

The objective of our research is to evaluate how current LLMs perform in translation tasks across languages of varying resource levels. Specifically, we aim to compare **performance, accuracy, and energy consumption** among LLMs of different sizes to assess their viability as a practical translation tool.

2. Methodology

For this study, we utilized datasets comprising seven language pairs: English-Spanish, English-Portuguese, English-Italian, English-Chinese, English-German, English-Amharic, and English-Dyula. This selection includes both high-resource languages (e.g., Spanish, Chinese) and low-resource languages (e.g., Amharic, Dyula) to ensure a comprehensive analysis.

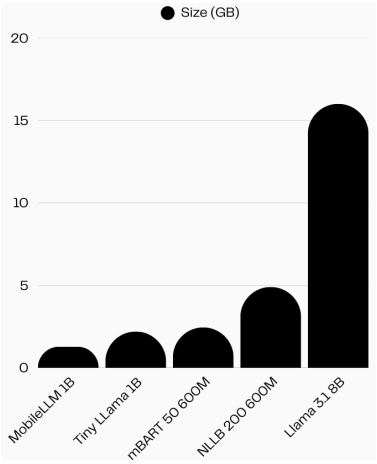
To evaluate translation quality, we employed two key metrics: **BLEU score[1]** and **BERT Score[2]**. BLEU provides a traditional reference-based evaluation, which is useful for assessing literal accuracy, though it can be overly rigid and sometimes misrepresentative of true translation quality. On the other hand, BERTScore leverages contextual embeddings, allowing a more nuanced and semantically aware evaluation. While a high BLEU score would indicate a model's competence, we consider BERTScore a more reliable indicator of real-world translation effectiveness, especially in cases where BLEU may fall short.

For this study, we sourced our datasets from the **OPUS[3] (Open Parallel Corpus)** repository, a widely used collection of multilingual datasets suitable for machine translation research. Our language selection process did not follow a strict criterion for defining high-resource and low-resource languages; instead, we identified low-resource languages as those with significantly fewer available parallel datasets compared to others within OPUS.

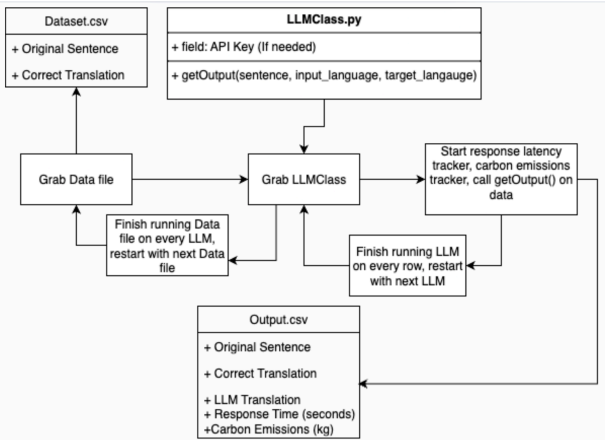
To maintain feasibility within our computational limits, we randomly selected **50 sentences from each language pair**. This sample size was chosen to balance between achieving representative results and managing the runtime constraints of processing multiple models and metrics on local hardware.

We evaluated the translation outputs of five different **Large Language Models (LLMs)**, each varying in size and architectural complexity. We used two types of LLMs. Chat LLMs and Translation LLMs. The first is a model in which the user must provide a prompt. This model is the type people are more familiar with such as chatGPT.com. The latter works more like a function inside a program, where parameters must be specified such as the input language, the target language, and the sentence to be translated.

We included 3 chat LLMs of varying sizes. Llama 3.1 8B[4] was used as our baseline, to represent the typical LLM people use every day. This model is not suitable to run locally in a practical context due to its size and hardware requirements. Tiny Llama 1.1B is representative of what is possible locally on a desktop. MobileLLM 1B is representative of what is possible locally on a mobile device such as a smartphone. For Translation LLMs, we used NLLB[5] (No Language Left Behind) for our low resource language testing as it supports 200 languages.



Physical Size of Each Model Tested



Data Processing Pipeline

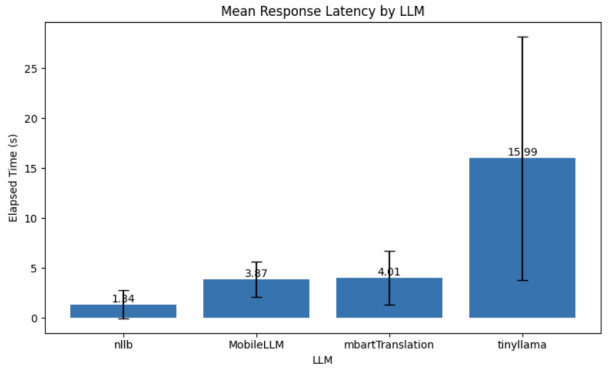
For each model, we computed both the **BLEU score** and **BERTScore** to assess translation quality. BLEU was used as a baseline metric to capture surface-level n-gram overlaps, while BERTScore provided a more context-aware evaluation by leveraging deep semantic embeddings. Given the known limitations of BLEU, particularly in low-resource contexts, we considered BERTScore to be the more indicative measure of translation effectiveness in our analysis.

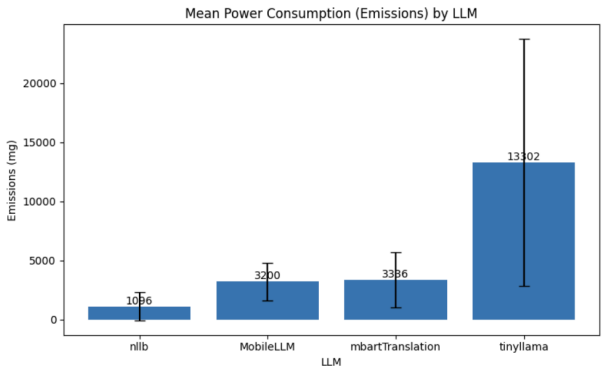
In addition to accuracy metrics, we also measured the **energy consumption** of each LLM during the translation tasks to evaluate their practical viability for mobile and low-power devices. This was achieved by using a software called codeCarbon[6]. codeCarbon can utilize native energy consumption measuring software from macOS to measure how much energy a python script consumes. The software then directly converts the energy consumption to a carbon emissions measurement. This is based on the computer's location. Therefore, the data for CO2 emissions from our testing is representative of Berlin.

Our last metric was measuring how long a LLM took to produce an output. This was simply done by wrapping our python code with two lines to start a timer and end it for each LLM call. With both energy consumption and latency, we have a solid understanding of the difference between each LLM in regards to efficiency.

3. Discussion

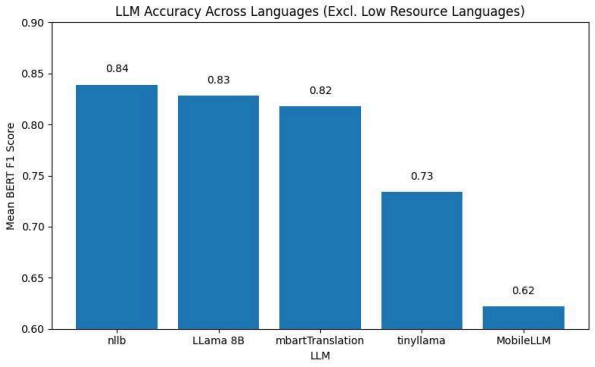
For the efficiency testing, the results for chat LLM represented what we expected. An increase in model size resulted in an increase in energy consumption and response latency for a given model. However, expectation deviated with the translation LLMs, as despite being larger than the non-standard chat LLMs, they were the most efficient of all the LLMs. This could be for two reasons. First, the translation LLMs total size is accounting for all the languages it supports, and therefore the components for translation of one pair of languages is a smaller size. Second, because the LLMs are trained specifically for translation, there could be less processing required because there is no need to evaluate the prompt, identify the intentions of the user, and to identify the languages and the sentence itself to be translated. All of these tasks, chat LLMs must do while translation LLMs do not.



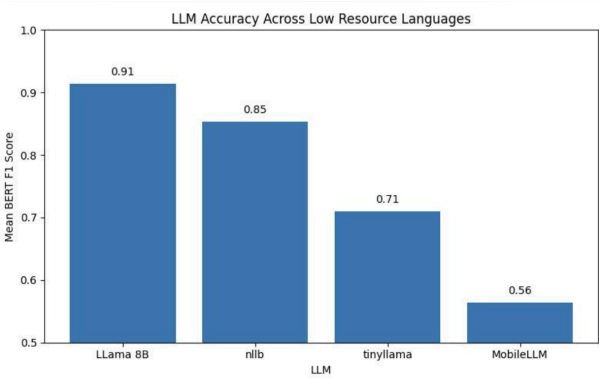


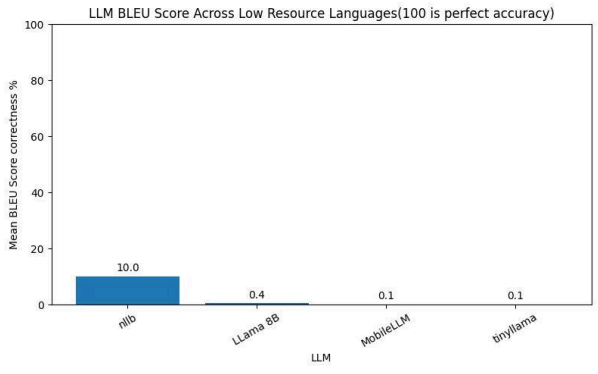
NLLB used only 11.5% of the energy Tinyllama used (the second largest chat LLM) and only used a miniscule 0.16% of the energy compared to Llama 3.1 8B(This model is not included in the graphs because with it the differences between the other models cannot be noticed because of the scale). With the results we got with efficiency, it was our hope that NLLB would perform competitively, given that it was the most

efficient model.



The results for high resource languages were promising. Both translation LLMs performed at approximately the same accuracy as the standard Llama 8B LLM with only 1% of difference in scores. Tinyllama performed decently at 73% accuracy, but experienced a steep drop compared to the other 3 LLMs. Finally, MobileLLM performed the worst at 62% accuracy. MobileLLM was never expected to be a useful model as in very early testing most results were pure hallucination. MobileLLM was only included to show the current capability of true, local models on mobile devices such as smartphones. The results with NLLB lead us to believe that it may be viable to have on-device translation with a LLM today.





Unfortunately, low resource languages did not achieve similar results. We measured polar opposite accuracy results for low resource languages depending on if we measured with BERT or BLEU. The BERT results were high and promising, but we were skeptical. When reviewing the data manually, the sentences between the data set and the LLM output did not seem to be similar. We suspected that the reason why BERT gave such high results was because it is based on pre-training of languages. So if BERT was never trained on a language, it cannot evaluate it accurately. Therefore, for languages which do not have extensive training data available, we believe that BERT is not the correct measurement score.

We then measured with BLEU, which does not require any pre-training data, and focuses on purely comparing the LLM output to the correct translation in the data set. Additionally, related work in the field of low resource languages, also used BLEU for measurement[7]. With BLEU, we obtained more believable results. NLLB, which claims to support the low resource languages we tested, only obtained 10% accuracy. And the rest of the LLMs achieved virtually 0% accuracy.

4. Related Work

Existing research was able to achieve significantly better results for low resource languages. Being able to achieve a 23 BLEU score compared to our high of 10 with NLLB[7]. However, these higher results were only producible with significant tooling. For example, the aforementioned result was achieved by splitting the sentence into fragments, and translating each fragment into different high resource languages, then translated again into the target language[7]. Another paper that achieved more promising results did so with a RAG-based approach where key terms were translated from existing data rather than the sentence and its entire context included[8].

5. Conclusions

We believe that today there exists a viable solution for the translation of high resource languages that is more efficient than standard LLMs and just as accurate. No Language Left Behind proved this with its accuracy being slightly better than generic Llama 3.1 8B and by being far more efficient, using only a fraction of the energy.

However, the performance of low resource languages leaves much to be desired. With no LLM performing anywhere close to achieving correct translation. It appears that traditional translation methods are not capable of reliable translation today. More robust and complex methods are needed.

With the poor performance of small chat LLMs, it seems that for local models, the future is trending towards LLMs that are specialized in one function. Such as the LLMs we used in our testing that were designed only for translation.

6. References

[1] "BLEU - a Hugging Face Space by evaluate-metric," huggingface.co. https://huggingface.co/spaces/evaluate-metric/bleu

[2] "BERT Score - a Hugging Face Space by evaluate-metric," huggingface.com

[3] "OPUS - an open source parallel corpus," opus.nlpl.eu. https://opus.nlpl.eu/

[4] "meta-llama/Llama-3.1-8B · Hugging Face," Huggingface.co, Sep. 25, 2024. https://huggingface.co/meta-llama/Llama-3.1-8B

[5] "200 languages within a single AI model: A breakthrough in high-quality machine translation" https://ai.meta.com/blog/nllb-200-high-quality-machine-translation

[6] "CodeCarbon.io," codecarbon.io. https://codecarbon.io/

[7] “Universal Neural Machine Translation for Extremely Low Resource Languages” https://arxiv.org/abs/1802.05368v2
[8] “Transcending Language Boundaries: Harnessing LLMs for Low Resource Translation” https://arxiv.org/abs/2411.11295