Dialogue Systems for Informal Malagasy : A Comparative Evaluation of LLaMA 3.2 and Mistral 7B

Francis Rakotomalala
Laboratory for Mathematical and Computer Applied to the Development Systems (LIMAD)
University of Fianarantsoa
Fianarantsoa, Madagascar
francis_rakotomalala@ymail.com

Aimé Richard Hajalalaina
Laboratory for Mathematical and Computer Applied to the Development Systems (LIMAD)
University of Fianarantsoa
Fianarantsoa, Madagascar
arhajalalaina@yahoo.fr

Ndaohialy Manda Vy Ravonimanantsoa
Engineering and Innovation Sciences and Techniques (STII)
University of Antananarivo
Antananarivo, Madagascar
ndaohialy@gmail.com

***Abstract*—In the context of low-resource language modeling, this article focuses on adapting the pretrained language models LLaMA and Mistral for the development of a dialogue system in informal Malagasy. Malagasy, being a rich but underrepresented language in data corpora, presents unique challenges in terms of vocabulary, syntactic structure, and informal variations. The aim of this research is to demonstrate the ability of modern language processing models to overcome these challenges and generate relevant responses in this language. The results show a notable improvement in model performance following a specific adaptation phase. A significant reduction in loss and perplexity was observed, indicating the models' effectiveness in learning and adjusting to the unique characteristics of informal Malagasy. The training of conversational agents also helped maintain good fluency and coherence in dialogues, although further adjustments are needed to improve lexical and syntactic alignment, which are essential for smooth and natural interaction. When comparing the performance of LLaMA 3.2 3B and Mistral 7B, Mistral stands out for its ability to generate more natural and fluid dialogues, while LLaMA excels in tasks requiring strict and precise content matching. These findings highlight the effectiveness of these models for developing dialogue systems tailored to the specificities of informal Malagasy, while also underlining the potential for ongoing improvement.**

Keywords*— *Chatbot, Langage Informal Malagasy, LlaMA, Mistral, NLP

I. INTRODUCTION

Text-based dialogue systems, also known as chatbots or conversational agents, are technologies designed to interact with users through textual exchanges. By analyzing user queries and generating appropriate responses, these systems aim to simulate human conversation in a fluent and coherent manner [1]. The development of such agents relies on various artificial intelligence techniques, including natural language processing (NLP) and deep learning [2]. Depending on their design, some models rely on predefined rules, while others use advanced language modeling architectures, such as causal models, which generate the next sentence based on the conversational context [3].

One of the main challenges for these systems is to ensure the coherence and relevance of responses—an especially complex issue for underrepresented languages such as informal Malagasy. Due to the lack of annotated corpora and appropriate linguistic resources, the automatic processing of this language remains a significant challenge [4]. In this context, pretrained language models such as LLaMA [5] and Mistral [6] offer a promising avenue for the development of dialogue systems tailored to the linguistic specificities of informal Malagasy. However, their effectiveness depends on specific adaptation processes aimed at capturing the subtleties of the language to enhance both understanding and response generation in a conversational framework [7].

This article explores the adaptation of these models to the characteristics of informal Malagasy by implementing targeted pretraining and specific architectural adjustments. We detail the methodological choices made, including tokenizer adaptation [8], the construction of a representative dataset—such as [9] and CC100-Malagasy[1]—and the fine-tuning process. A comparative analysis of the performance of the adapted LLaMA and Mistral models is also conducted, using metrics such as loss, perplexity, and conversational quality scores (BLEU, ROUGE, BertScore).

The organization of this article is as follows: we begin with a state-of-the-art review of dialogue systems and the specific challenges of low-resource languages. Next, we describe the selected models and the rationale behind their selection. We then present the adaptation to informal Malagasy, detailing the pretraining, fine-tuning stages, and dataset construction. Finally, we present the obtained results and analyze the performance of the models before discussing potential improvements and future applications.

II. RELATED WORKS

Language models such as LLaMA[2] and Mistral represent significant advances in the field of text generation, particularly

[1] https://metatext.io/datasets/cc100-malagasy

[2] Large Language Model Meta AI

Cite as: F. Rakotomalala, A. R. . Hajalalaina, and N. M. V. Ravonimanantsoa, "Dialogue Systems for Informal Malagasy : A Comparative Evaluation of LLaMA 3.2 and Mistral 7B", *Syst. Theor. Control Comput. J. ,* vol. 5, no. 1, pp. 26–40, Dec. 2025.
DOI: 10.52846/stccj.2025.5.1.69

in dialogue systems. LLaMA and Mistral stand out for their optimized size and efficiency, aiming to strike a balance between text generation performance and resource consumption. LLaMA was designed to be lighter than massive models like GPT-3 while offering comparable performance on text generation tasks. This model is especially valued for its ability to be deployed with fewer resources while maintaining robust results on complex tasks, including multilingual applications.

On the other hand, Mistral is a dense model specifically optimized for multilingual languages, with a particular focus on underrepresented languages. This model has demonstrated the ability to handle a wider range of linguistic diversity while being lighter and faster than other models of similar size. It represents an efficient and flexible alternative for processing low-resource languages while maintaining high performance in text understanding and generation tasks [10].

These models are based on the Transformer architecture [2], which has become the standard for text processing tasks due to its ability to capture complex relationships between words in a sequence. The Transformer architecture—especially autoregressive models like LLaMA and Mistral—generates each word based on previous ones, making it particularly suitable for chatbot systems that require dynamic management of conversational contexts.

Informal Malagasy presents a unique set of challenges for text generation systems. It is a language rich in orthographic variation and influenced by several other languages, notably French and English. This variability makes informal Malagasy difficult to model, especially for models that were initially trained on more standardized languages or more formal variants. Moreover, the lack of suitable textual and linguistic resources for this register complicates the effective training of language models for this specific case.

Handling this variability is crucial to maintaining fluency in responses in informal contexts, such as social media conversations or text messaging. Indeed, adapting models to the specificities of this register requires a targeted approach, including the consideration of slang terms, orthographic variations, and the cultural and linguistic imprints unique to informal Malagasy.

Pretrained models trained on large multilingual datasets offer considerable potential for processing informal Malagasy. These models can leverage linguistic patterns shared across different languages, facilitating their adaptation to the specificities of informal Malagasy, particularly through their ability to learn varied lexical and syntactic relationships.

The transformer architecture allows autoregressive models to generate each word based on the preceding one, which is essential for chatbots where continuity and coherence in exchanges must be maintained. These models learn to manage long contexts and long-range dependencies, characteristic of conversations that span multiple exchanges. This ability is particularly useful in the context of informal Malagasy, where flexible lexical and syntactic forms frequently appear.

One of LLaMA's key advantages is its ability to generate text with efficiency comparable to that of larger models, while being optimized for low-resource languages. This makes LLaMA particularly well-suited for languages like Malagasy, which have limited textual resources. Furthermore, Mistral, being denser and optimized for multilingual tasks, can also effectively handle the diverse linguistic forms of informal Malagasy, making it especially well-suited for this type of challenge.

III. Model Description

In this study, we selected causal decoder-based models, namely Llama and Mistral, which were evaluated independently in order to compare their performance. These two models were chosen for their ability to handle text generation tasks while taking into account the linguistic characteristics of the studied corpus, as well as for their distinct architectural properties, allowing us to analyze the impact of their design choices on the obtained results.

A. Causal Language Model formulation

For a causal Transformer model based on a decoder architecture, the mathematical formulation can be presented as follows.

Let $x = (x_1, x_2, \dots, x_T)$ be a sequence of tokens. The objective is to model the joint probability of the sequence as a product of conditional probabilities:

$$P(x_1, \dots, x_T) = \prod_{t=1}^{T} P(x_t \mid x_{<t}) \tag{1}$$

where $x_{<t} = (x_1, \dots, x_{t-1})$ denotes the preceding context. This formulation is referred to as causal because each token is predicted only from the previously observed tokens.

Each token is first transformed into a vector representation by combining token embedding and positional encoding:

$$h_t^{(0)} = E(x_t) + p_t \tag{2}$$

where $E(x_t)$ denotes the embedding of token x_t, and p_t denotes its positional encoding.

At layer l, the self-attention mechanism computes the query, key, and value matrices as follows:

$$Q^{(l)} = H^{(l-1)}W_Q^{(l)}, K^{(l)} = H^{(l-1)}W_K^{(l)}, V^{(l)} = H^{(l-1)}W_V^{(l)} \tag{3}$$

where $H^{(l-1)} \in \mathbb{R}^{T \times d}$.

In a decoder-based model, masked self-attention is applied so that each position cannot access future tokens:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top + M}{\sqrt{d_k}}\right)V \quad (4)$$

where $M$is a causal mask defined by:

$$M_{ij} = \begin{cases} 0 & \text{if } j \le i \\ -\infty & \text{if } j > i \end{cases}$$

After several self-attention and feed-forward blocks, the final representation $h_t^{(L)}$is projected onto the vocabulary space in order to predict the next token:

$$P(x_t \mid x_{<t}) = \text{softmax}(W_o h_t^{(L)} + b_o) \quad (5)$$

Training generally consists of minimizing the negative cross-entropy loss:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}) \quad (6)$$

This architecture is used by autoregressive language models such as GPT, Llama, and Mistral.

B. LLaMA

LLaMA is one of the leading open-source state-of-the-art language models released by Meta in 2023. It is a lighter model in terms of size while remaining powerful for low-resource languages such as Malagasy. It has a Transformer-based architecture, similar to GPT, and is particularly well-suited for multilingual tasks.

1) Specific Features

a) Structure

LLaMA 1 was initially released with four different variants with 6.7B, 13B, 32.5B, and 65.2B parameters. The number of heads in the multi-head attention mechanism for each of these variants is 32, 40, 52, and 64 respectively, unlike the original Transformer which used 8 attention heads. In LLaMA, each token from the input embedding is represented by a vector whose dimensionality varies depending on the model size. Specifically, for the 6B parameter model, each token is represented by a 4,096-dimensional vector. For the 13B model, the dimensionality increases to 5,120. In the case of the 32.5B model, tokens are represented by 6,656-dimensional vectors, while for the largest 65.2B model, tokens are represented by 8,192-dimensional vectors. This contrasts with the original Transformer architecture, which used 512-dimensional vectors to represent each token. Moreover, the input embeddings in LLaMA models are not static but are learned during training itself.

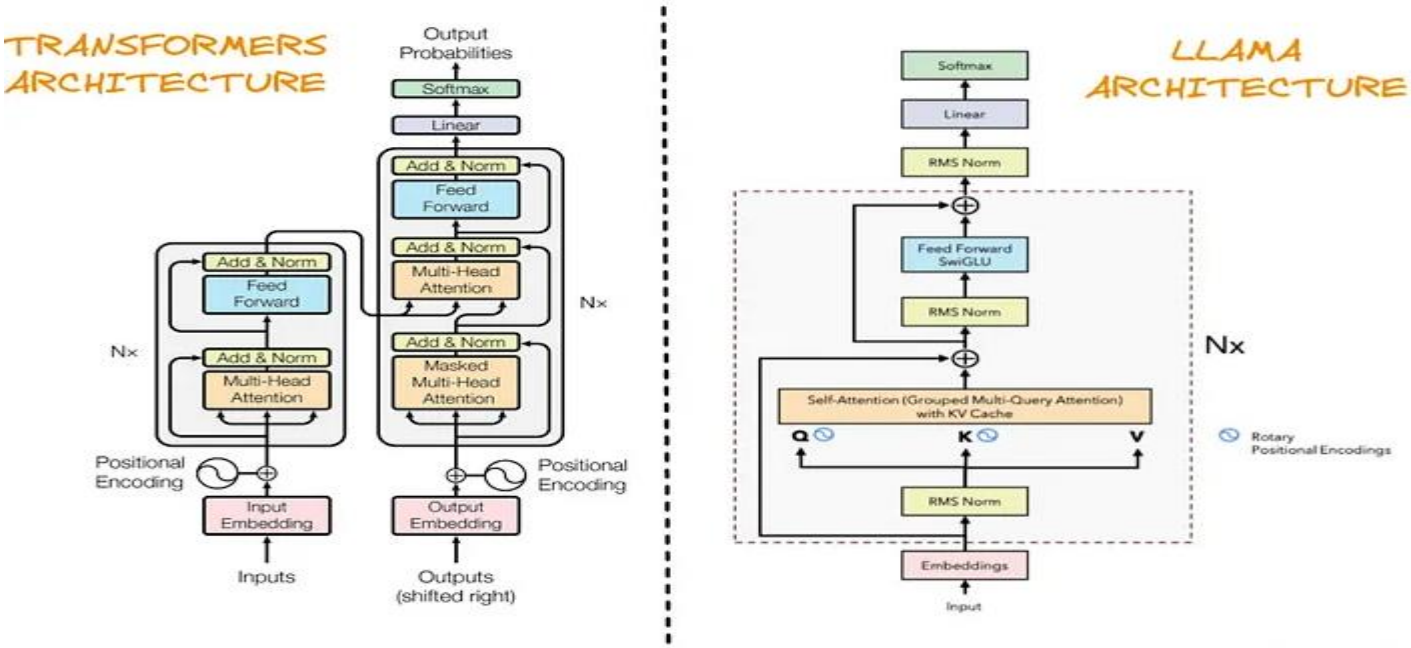

Fig. 1. Transformers architecture VS LLaMa architecture[3]

[3] https://medium.com/@pranjalkhadka/llama-explained-a70e71e706e9

b) Normalization

One of the main differences between Llama and the original Transformer architecture lies in the normalization mechanism. In the original Transformer, Layer Normalization is applied after each multi-head attention block and feed-forward block. For each token representation, the activations are normalized by subtracting their mean and dividing by their standard deviation. This operation stabilizes the scale of the activations during training.

LLaMA adopts a different variant called Root Mean Square Layer Normalization (RMSNorm) [11]. Unlike Layer Normalization, RMSNorm does not subtract the mean. Instead, it rescales the input using only its root mean square value. The underlying idea is that the main benefit of normalization comes from controlling the magnitude of the activations rather than centering them around zero.

Formally, for an input vector $x \in \mathbb{R}^d$, RMSNorm is defined as:

$$\text{RMSNorm}(x) = \gamma \cdot \frac{x}{\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2 + \epsilon}} \tag{7}$$

where γ is a learnable scaling parameter, d is the dimensionality of the representation, and ϵ is a small constant for numerical stability.

By avoiding mean subtraction, RMSNorm reduces computational overhead while preserving stable training dynamics, which contributes to the efficiency of LLaMA models.

c) Positional Encoding

In the Transformer architecture, positional encoding is done by adding a vector of the same size to the input embedding to ensure that positions matter in an input sequence. These positional encoding vectors are computed once and reused throughout the training process. LLaMA, on the other hand, uses something known as rotary positional embedding. This method introduces rotational operations into the positional encoding process [12], allowing the model to learn dynamic positional representations during training rather than relying on precomputed static positional encoding vectors.

d) Token Prediction

Like most large language models, LLaMA is trained on the next-token prediction task. Inference is generally performed by starting with a special token called the start token as input. From there, it generates the first word based on the start token. Then, using both the start token and the first word as context, the model generates the second word, and this process continues iteratively until another special token, called the end token, is encountered.

Thus, at each time step, the model predicts the next token, concatenates it with the input, and repeats the process.

This iterative process involves many redundant computations of the same tokens that were already calculated at previous time steps. Therefore, a method known as the key-value (KV) cache is used in LLaMA to optimize this process, where only the key and value vectors are cached, while the query vector is updated at each step. This allows the model to reuse the key and value vectors across multiple steps, reducing redundant computations during token generation and thereby speeding up inference.

e) Attention Mechanism

Another major component is grouped multi-query attention. It is based on multi-query attention introduced in [13]. It was found that while GPUs are fast at performing operations on tensors or matrices, the bottleneck arises when data needs to be moved between memory and processing units like the GPU. Thus, as memory access increases, the overall time complexity of performing matrix computations during inference in large language models also increases. With the addition of KV caching in LLaMA, this issue becomes even more persistent.

To address this, multi-query attention removes the head dimension (h) from the key (K) and value (V) while keeping it for the query (Q). This means that all different query heads will share the same keys and values. As a result, inference becomes faster than standard multi-head attention with KV caching, with only a slight degradation in output quality.

Grouped multi-query attention extends this idea by dividing the query into different groups, and for each group, there is a set of keys and values. Thus, grouped multi-query attention lies between multi-query attention and standard multi-head attention, maintaining a balanced trade-off between speed and quality.

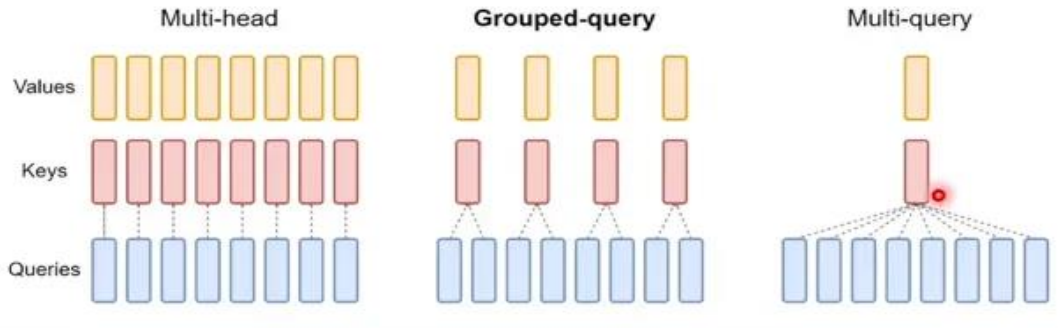

Fig. 2. Concept of grouped multi-query attention[4]

f) Activation Function

Another important difference between Llama and the original Transformer concerns the activation function used in the feed-forward layer. In the standard Transformer, this layer generally uses the ReLU (Rectified Linear Unit) activation. In contrast, LLaMA adopts SwiGLU, introduced in [14], as a more expressive variant of GLU (Gated Linear Unit).

The main idea is that, instead of applying a simple nonlinear transformation to the input, SwiGLU introduces a gating mechanism. One part of the input produces candidate information, while another part controls which information should be amplified or suppressed. This allows the model to more selectively retain useful information during training.

In the feed-forward layer, for an input x, SwiGLU is defined as:

$$\mathrm{SwiGLU}(x, W, V, b, c, \beta) = \mathrm{Swish}_\beta(xW + b) \odot (xV + c) \quad (8)$$

where $W$and $V$are weight matrices, $b$and $c$are bias terms, $\odot$denotes element-wise multiplication, and $\mathrm{Swish}_\beta$is defined as:

$$\mathrm{Swish}_\beta(z) = z\,\sigma(\beta z) \quad (9)$$

with σ denoting the sigmoid function.

Compared with ReLU, which simply sets negative values to zero, SwiGLU provides a more flexible and selective transformation. This property generally improves the representational capacity of the network and promotes more stable learning.

2) Justification of the choice

LLaMA represents a promising advancement in the field of large language models, offering innovative modifications to traditional attention mechanisms. Its open-source nature has led to widespread adoption for fine-tuning on various tasks.

a) Efficiency in Low-Resource Languages

LLaMA is optimized to perform well with languages that are underrepresented in large datasets, making it an excellent candidate for processing Malagasy texts.

b) Resource-Efficient

This model is less demanding in terms of computing power while maintaining high performance, making it suitable for environments with limited resources.

c) Multilingual Capability

Its ability to effectively handle multiple languages allows it to better capture the nuances of Malagasy in informal contexts.

C. Mistral

Introduced in early 2024, Mistral represents a significant advancement in the field of open-source natural language models. This model stands out for its ability to generate and understand human language with enhanced accuracy and versatility.

Its architecture enables the creation of fluent and coherent content, making it particularly useful for conversational tasks.

1) Specific Features

Compared to LLaMA, it introduces some changes:

a) Sliding Window Attention

[4] https://medium.com/@pranjalkhadka/llama-explained-a70e71e706e9

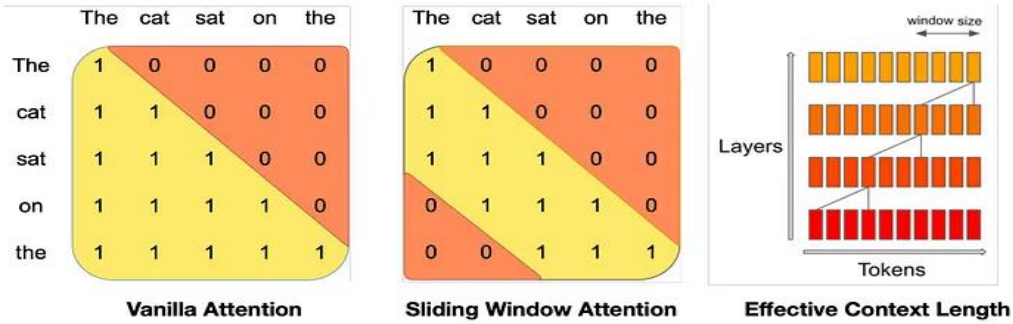

Fig. 3. Concept of SWA[5]

Sliding Window Attention (SWA) leverages the layers of a transformer model to extend its attention beyond a fixed window size W. In this mechanism, the hidden state at position i in layer k can attend to the hidden states of the previous layer within the range of positions i–W to i, enabling access to tokens up to W × k tokens away. Using a window size of W = 4096, FCA theoretically achieves an attention span of about 131,000 tokens. In practice, with a sequence length of 16K and W = 4096, FCA modifications in FlashAttention and xFormers result in a 2x speed improvement compared to standard attention methods.

b) Rolling buffer cache

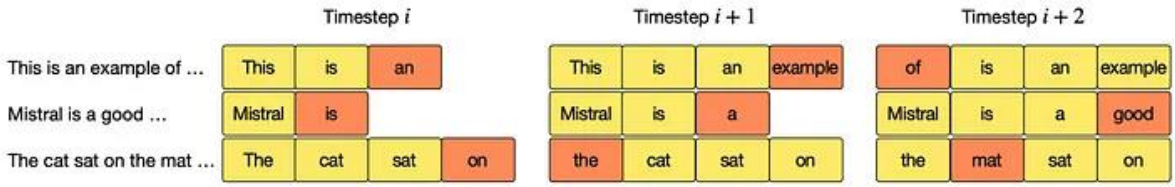

Fig. 4. Concept of rolling buffer cache[5]

A rolling buffer cache uses a fixed attention duration to limit the size of the cache. The cache has a fixed size *W* and stores the keys and values for time step *i* at position *i%W* in the cache. When i exceeds W, previous values are overwritten, stopping the growth of the cache size. For example, with *W = 3*, on a 32K token sequence, the cache memory usage is reduced by 8x without compromising model quality.

c) Pre-filling and Chunking

In sequence generation, tokens are predicted sequentially based on prior context. To optimize efficiency, a (k, v) cache is pre-filled with the known prompt. If the prompt is very long, it is divided into smaller segments using a chosen window size. Each segment is used to pre-fill the cache. This approach involves calculating attention both in the cache and on the current segment, contributing to more efficient sequence generation.

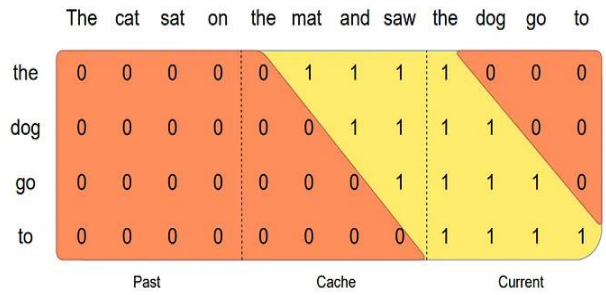

Fig. 5. Concept of pre-filling and chunking[5]

2) Justification of the Choice

With its advanced capabilities in information generation and language understanding, Mistral not only delivers high performance but also offers flexibility and adaptability, making it valuable for a wide range of tasks in NLP.

a) Text Generation Performance

Mistral can be used to generate coherent and contextually relevant text, which is essential for a dialogue system.

[5] https://medium.com/dair-ai/papers-explained-mistral-7b-b9632dedf580

b) Adaptability and Customization

As an open-source model, Mistral provides greater flexibility to be adapted and customized to meet specific needs, especially for underrepresented languages such as Malagasy.

IV. Extension to the New Language: Informal Malagasy

LLMs demonstrate remarkable capabilities in understanding and generating human language, but adapting to a new grammar and vocabulary can be quite challenging. When an LLM encounters a new language, it must learn the specific syntactic and morphological rules that govern sentence structure, as well as the unique lexical elements that make up the vocabulary. This process typically involves pre-training and/or fine-tuning the model on a corpus rich in the target language. During this process, the model adjusts its weights to better capture the linguistic patterns and nuances of the new language. This, in turn, allows the LLM to generate grammatically correct and contextually appropriate text in the given target language.

A. Language-Specific Pre-training

The first step in language-specific pre-training involves collecting a large and diverse text corpus in the target language. This corpus should be as varied as possible to capture a wide range of linguistic patterns and vocabulary. Sources can include books, articles, websites, social media, etc. While having a large amount of data is recommended, experiments such as [15] have been conducted where the model was adapted to a new language using only 16,000 tokens for pre-training. Once the data is collected and pre-processed, the LLM is trained on the data corpus. The model learns the statistical properties of the language, including grammar, syntax, and vocabulary. Although this step is considered computationally expensive, it depends on the size of the pre-training corpus.

1) Tokenizer Adaptation to the New Vocabulary

The tokenizer of an LLM plays a crucial role in managing the new vocabulary. Tokenizers break the text into smaller units, such as words or subwords, which are then processed by the model. When adapting an LLM to a new language, the tokenizer must be adjusted to appropriately recognize and segment the unique words and subword units of that language.

There are several strategies for tokenizer adaptation:

- Training from scratch: This involves creating a new tokenizer based on a full corpus of the target language. While this ensures that the tokenizer is well-suited to the new language, it can be computationally expensive.
- Augmenting existing tokenizers: Existing tokenizers can be augmented with new vocabulary items. This method requires less computation and allows the model to retain its capabilities in the original language while supporting the new language.

2) Instruction Fine-Tuning

The pre-trained model is then fine-tuned on specific instructions. During this phase, the model learns to apply its understanding of language to effectively perform particular tasks. Fine-tuning adjusts the model's weights according to the specific requirements of the tasks, thereby improving its performance and accuracy. This step is followed by further alignment and evaluation stages.

3) Alignment

Alignment ensures that the model's outputs align with human values, ethical guidelines, and cultural sensitivities. This may include integrating human feedback to refine the model's behavior, avoid biases, and ensure that the generated content is suitable for the target audience.

B. Dataset

1) FLICs

This dataset for informal Malagasy language is designed to capture the nuances of informal Malagasy usage. It focuses primarily on daily conversations, social media exchanges, and informal texts. The goal of this dataset is to provide a wide range of texts reflecting the informal use of Malagasy, including characteristics of informal communication such as abbreviations, dialects, slang, emoticons, and keywords. Additionally, it highlights the less-studied Malagasy language, used in a highly informal manner and mixed with French through code-switching.

- Type of documents: Informal texts (posts, comments, social media conversations).
- Language: Informal Malagasy variant.
- Tasks: Pre-training language models, informal language analysis.
- Usage: Training models to better understand variations of Malagasy in informal contexts.

2) CC100-Malagasy Dataset

The CC100-Malagasy Dataset is part of the CC100 project, which collected extensive multilingual corpora for a set of underrepresented languages in the context of multilingual model pre-training. The Malagasy subset of CC100 contains web data from various sources, mainly in formal domains such as press articles, blogs, and other types of published texts in Malagasy.

- Type of documents: Formal web texts (blogs, articles, websites).

- Language: Formal Malagasy.
- Tasks: Linguistic modeling, text classification.
- Usage: Training language models for understanding Malagasy in more formal contexts.

Integrating the FLICs and CC100-Malagasy datasets into our approach allows for the design of a robust and versatile language model capable of understanding and generating text in various contexts, from informal to formal registers. This strategic combination ensures comprehensive coverage of the different nuances and variations of the Malagasy language, providing a solid foundation for diverse applications. It effectively addresses the requirements of dialogue tasks, whether managing informal conversations or performing linguistic modeling in more structured contexts, ensuring enhanced flexibility and accuracy in the dialogue system.

C. Implementation

1) Pre-training

Pre-training allows the model to adapt to linguistic variations that are often overlooked in general corpora. In particular, the model is trained to understand informal sentence structures, syntactic inconsistencies, and common dialectal variations found in everyday conversations. This approach enhances the understanding of informal texts.

This pre-training is essential in cases where the target language, such as Malagasy, is underrepresented in classic multilingual corpora, requiring a tailored approach to capture the specifics of daily and informal usage.

The outcome of this step is a model that can now perform automatic text entry in informal Malagasy. To make it capable of having conversations, we must perform dialogue fine-tuning.

2) Dialogue System Training

As part of the development of a conversational agent, we have designed a structured training process based on a dialogue dataset. The first step involves formatting the prompts for inputs and outputs in a way that encourages relevant and coherent responses.

a) Prompt Preparation

The prompt used to generate dialogues follows the structure below:

“””

Ity dia misy fifanakalozan-kevitra. Omeo ny valiny arakaraka ny fangatahana.

Fangatahana:

{d1}

Valiny:

{d2}

“””

For each pair of dialogues (dialog1 and dialog2), we add an end-of-sequence token to ensure that the text generation stops appropriately.

b) Training with UnslothTrainer

We use the UnslothTrainer for model training. This trainer is configured with various parameters to optimize the training process.

The training of the conversational agent is carried out in a structured manner, allowing for proper data preparation and resource optimization during the training process. This approach ensures that the model can generate relevant and informative responses based on user queries.

V. RESULTS

A. Fine-tuned LLaMA

1) Pre-training for the New Language

a) Loss

- Initial Loss: The initial loss is 3.28 for training and 3.36 for testing, indicating that the model is still in the learning phase and has more to learn in order to master the linguistic peculiarities of Malagasy. These relatively high values reflect the need for a deep understanding of the data.
- Loss After One Epoch: After one epoch of training, the loss significantly decreased, reaching 1.52 for training and 1.43 for testing. This notable reduction confirms the effectiveness of the learning process and the model's ability to adapt to the linguistic data.

The figure 6 illustrates the evolution of the loss value for LLaMA 3.2 3B, recorded at every 1,000-step interval:

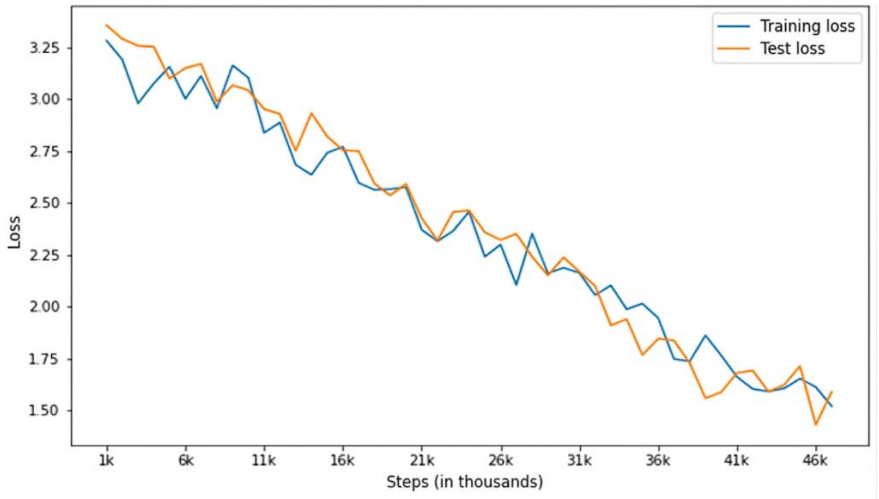

Fig. 6. Evolution of the loss function (LLaMA 3.2 3b)

b) Perplexity

- Initial Perplexity: The model's initial perplexity was measured at 28.79, reflecting some difficulty in correctly predicting word sequences in the Malagasy language.
- Final Perplexity: By the end of training, the perplexity was reduced to 4.1, indicating a substantial improvement in the model's ability to understand and predict linguistic structures. This decrease in perplexity demonstrates better performance in generating relevant text.

2) Conversational Agent Training

a) Loss

- Training Loss: The initial training loss is 1.5 and rapidly decreases to 0.74 after just one epoch, indicating that the model is learning effectively and adapting well to the training data. This sharp drop in loss is a positive sign, suggesting that the model is quickly capturing relevant linguistic structures during the early iterations.
- Test Loss: The initial test loss is slightly below 1.4 and also drops to 0.66 after one epoch. The relatively small difference between training and test losses indicates good generalization, meaning the model can apply what it has learned during training to unseen data without overfitting.

b) Perplexity

The measured perplexity is 1.93, which is an excellent indicator for a language model. Such a low perplexity reflects high confidence in sequence prediction and the model's ability to produce coherent and appropriate responses within the conversational task. This demonstrates a deep understanding of syntactic and semantic relationships—a key step for an effective conversational agent.

c) Evaluation with Additional Metrics

The results highlight various aspects of the conversational agent's performance based on LLaMA 3.2 3B, as shown in the table 1:

TABLE 1 . Excerpt from additional metrics on LLaMA 3.2 3b

dialog2	pred	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BertScore
io n tanananay ts foinay e	io n tanananay ts foinay e	100.000000	1.000000	1.000000	1.000000	1.000000
mankasitraka tompoko manana anay manana anao m...	mankasitraka président, tena miasa ianao, tsy ...	6.167638	0.428571	0.230769	0.357143	0.847617
salut daholo nareo ilalao iaraka amin'i tatiana	coucou hilalao hiaraka amin'i tatiana	14.320952	0.428571	0.333333	0.428571	0.886662
marina mintsy zany e,mba t hiomehy e	marina zany e,tena tiko be le izy	19.070828	0.375000	0.142857	0.375000	0.863918
hita f tia raha :-p	ts tia raha aon ko?	16.233396	0.400000	0.250000	0.400000	0.826847
mia andrianarivo je veux tu sais déjà ce que j...	erica randrianarisoa je veux que tu me le dema...	8.054496	0.333333	0.090909	0.250000	0.851048
d aon n fomba anomezako anzay andreo	d aon zany	14.506310	0.400000	0.250000	0.400000	0.855319
tadidiko mints io kabary io	tadidiko mints io kabary io	100.000000	1.000000	1.000000	1.000000	1.000000
aty aminay 8 isa 500ar	aty aminay 7000 a kg..manga diego	9.535414	0.333333	0.200000	0.333333	0.884070

- Average BLEU Score: 18.75

A moderate score, indicating that the agent partially reproduces the target sequences, though with some divergence in structure or vocabulary.

- Average ROUGE Scores:

ROUGE-1 (0.25) and ROUGE-2 (0.16) reveal limited overlap in words and bigrams with the reference texts, which is common in dialogue tasks where responses may vary.

ROUGE-L (0.23) shows a low-to-moderate sentence-level similarity between the agent's responses and the target responses in terms of the longest matching sequences.

- BertScore: 0.82

Indicates strong semantic similarity, suggesting that the agent's responses generally capture the intended meaning, even if the wording differs.

In summary, the fine-tuned LLaMA model demonstrates a solid grasp of semantics with content that reasonably aligns with the targets. However, further refinement could enhance lexical and syntactic precision.

B. Fine-Tuned Mistral

1) Pretraining for the New Language

a) Loss

- Initial Loss: The loss was measured at 3.4 for training and 3.8 for testing, indicating that the model was still in the learning phase. These relatively high values suggest that the model still needs to acquire knowledge of the specific linguistic features of the Malagasy language.
- Loss after Three Epochs: After one epoch of training, the loss decreased significantly, reaching 1.4 for training and 1.3 for testing. This substantial reduction demonstrates the effectiveness of the learning process and the model's ability to adapt to linguistic data.

This figure 7 shows the loss evolution for Mistral 7B at every 1000 steps:

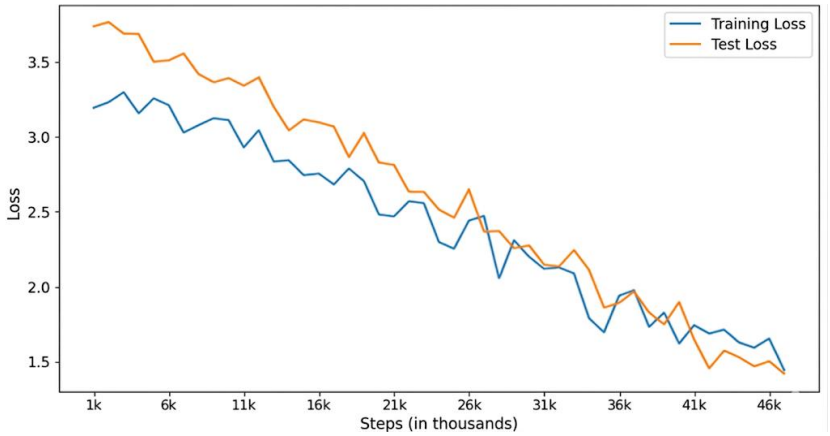

Fig. 7. Evolution of the loss function (Mistral 7b)

b) Perplexity

- Initial Perplexity: The model's perplexity was initially measured at over 28, reflecting some difficulty in correctly predicting word sequences in the Malagasy language.
- Final Perplexity: By the end of training, the perplexity was reduced to 4, indicating a substantial improvement in the model's ability to understand and predict linguistic structures. This drop in perplexity demonstrates better model performance in generating relevant text.

2) Conversational Agent Training

a) Loss

The results show a significant decrease in training loss, dropping from 1.4 to 0.58. This indicates that the model effectively learned from the training data, suggesting an improved ability to predict appropriate responses.

Regarding the test data, the initial loss is slightly lower than the training loss (1.3 vs. 1.4), which is a good sign. The final test loss of 0.56, close to the final training loss, suggests that the model generalizes well to new data without significant overfitting.

This comparable performance between training and test sets reinforces the idea that Mistral 7B has developed a solid understanding of the input data and is capable of generating appropriate responses in various dialogue contexts.

b) Perplexity

The perplexity was measured at 1.75, indicating that the model maintained good performance in understanding word sequences and making predictions in a dialogue context. A relatively low perplexity reflects better mastery of linguistic structures.

c) Evaluation with Additional Metrics

The scores obtained with Mistral 7B show overall encouraging performance, although some aspects could be improved to enhance the model's effectiveness on specific tasks.

The table 2 presents an overview of the expected and generated responses along with additional metrics:

TABLE 2 . EXCERPT FROM ADDITIONAL METRICS ON MISTRAL 7B

dialog2	pred	BLEU	ROUGE-1	ROUGE-2	ROUGE-L	BertScore
zah kosa aloa tena nalahelo e	za tena latsaka ranomaso mitsy	8.745825	0.181818	0.000000	0.181818	0.834444
mankasitraka ny vao2 mbs tv	mankasitraka ny mbs. miarahaba ny mpanaraka ny...	8.295194	0.428571	0.166667	0.428571	0.869901
hummm tsy manana io jio itano	tsy mahay manaja ny zavatra rehetra ny olona s...	4.767707	0.133333	0.000000	0.133333	0.824628
izay vao tena lamban akoho	ny tompo no mahalala izay ataon'ny olona rehetra	5.522398	0.142857	0.000000	0.142857	0.819153
mbola nahafinaritra ny fiaramonina tamin'izany...	mba te hande azy itony ny tenako	4.278179	0.105263	0.000000	0.105263	0.774245
feony iny ka ,ny habeany no nihena	tsy mahay mihitsy izy ka	5.862502	0.166667	0.000000	0.166667	0.831836
herinjaka rakotoarivelo io lesy ny ambom a!	io n tanananay ts foinay e	5.815868	0.153846	0.000000	0.153846	0.797908
de sendra n'y omby maditra zany ka misaoty de ...	tsy mahazo mandeha amin'io zany ny ankizy	4.278179	0.105263	0.000000	0.105263	0.843842
am 06h00 maraina ny départ eo am gare soarano.	mba te hande azy itony ny tenako	4.278179	0.117647	0.000000	0.117647	0.804165
ny olona efa mahavita mivezivezy eny amin'ny v...	tsy mahagaga fa tsy mahazaka ny fahasambarany ...	4.338332	0.320000	0.086957	0.160000	0.829621

- Average BLEU Score: 16.7

This BLEU score indicates a moderate match between the generated responses and the reference responses, but it remains relatively low compared to expectations for advanced text generation systems. A BLEU score below 20 suggests that the model may not perfectly capture n-gram overlaps, which could be due to the language's complexity and the linguistic variations present in the corpus. It may also indicate that the model lacks precision in generating phrases closely aligned with the reference examples.

- Average ROUGE Scores – ROUGE-1 (0.20), ROUGE-2 (0.10), and ROUGE-L (0.19)

ROUGE scores measure the overlap between generated and reference responses in terms of lexical units and word sequences. A ROUGE-1 score of 0.20 means the model captures around 20% of the unigram overlaps, while the lower ROUGE-2 score (0.10) suggests limited coverage of bigrams or adjacent word pairs. The ROUGE-L score (0.19) shows that the model retains a coherent sequence structure, though improvements in fluency and coherence are still possible.

- Average BertScore: 0.80

A BertScore of 0.80 is relatively high and indicates strong performance in terms of semantic similarity. This score shows that Mistral 7B is capable of capturing important information and producing responses with relevant semantic structures, even if exact lexical matches are sometimes weak, as reflected in the BLEU and ROUGE scores.

C. Model Usage

The fine-tuned models have shown promising ability to generate relevant and context-appropriate responses across various conversational settings. The following examples, generated by Mistral 7B, illustrate its effectiveness in informal situations.

Example 1 :

➔ Input: Bjr sarotr be v n hianaran gaucher guitare zfd
➔ Response: Tsy sarotra fa mila mikotrana kely fotsiny.

In this example, the model responds in a friendly and reassuring tone, suggesting that learning to play the guitar as a left-hander isn't difficult, but just requires some practice. This demonstrates its ability to engage in light conversation while providing helpful advice.

Example 2 :

- ➔ Input: Inona angaha ry reto no mahatonga an'ilay tanana sy tongotra vakivaky foana mandavan-taona a?? Tsy mafana tsy mangatsiaka!
- ➔ Response: Mila manao analyse diabeta fa mety misy.

Here, the model answers a health-related concern with a concrete suggestion. By recommending a diabetes test, it shows understanding of the user's concern and proposes a plausible solution, demonstrating competence in providing informative responses.

Example 3 :

- ➔ Input: Vana misy mahasaky manologna ampela ve ny kilongalahy sy mian-tsena
- ➔ Response: Efa vitsy ny kilongalahy mahasaky mangala ampela any.

In this last example, the model successfully understands and responds using a regional dialect, showing its ability to adapt to localized language forms. The response conveys an opinion on young men's behavior in a specific cultural context, highlighting the model's linguistic flexibility.

These examples underscore the fine-tuned model's ability to deliver appropriate and relevant responses regardless of the nature of the input message.

VI. Discussion and Recommendations

A. Self-Generalization to the New Language

The pretraining of models on the new language, Malagasy, has shown encouraging results despite a difficult initial phase. Indeed, after 50 training epochs, the loss dropped significantly for both training and testing, which demonstrates the model's strong adaptability. This sharp reduction suggests that the model quickly learned and adapted to the Malagasy linguistic structures, reflecting the effectiveness of the fine-tuning process.

Simultaneously, perplexity also decreased at the end of training. This indicates that the models significantly improved their ability to correctly predict word sequences—an essential aspect of generating coherent and relevant Malagasy text. Lower perplexity reflects better handling of linguistic variation.

These results are very promising, but there is still room for improvement, particularly in optimizing the model's performance in contexts where data is less structured or more complex.

B. Quality of the Generated Dialogue

Training the conversational agent based on the models also produced promising results, notably with low loss values in both training and testing. The minimal gap between these values clearly indicates the models' ability to generalize learning without overfitting. This ensures robustness in handling dialogue, especially in Malagasy, where the models seem to adapt well to informal linguistic variation.

The measured perplexity supports this observation, showing that the models maintain good fluency and coherence in sequence predictions—two essential aspects of a dialogue system.

Additionally, the results from supplementary metrics show a strong overall semantic understanding, as evidenced by the BertScore, which indicates that responses capture the expected core meaning. The moderate BLEU score suggests that the model partially reproduces the target sequences, although discrepancies remain in terms of structure and vocabulary.

ROUGE-1 and ROUGE-2 scores reveal limited alignment with reference words and bigrams, reflecting the frequent variation in dialogue tasks. ROUGE-L shows low-to-moderate similarity in sentence structure, highlighting potential for improvement in lexical and syntactic accuracy.

Thus, while the responses are semantically relevant, adjustments are recommended to strengthen lexical and syntactic alignment.

C. Comparison of the Two Fine-Tuned Models

The table presents a comparison of the performance of LLaMa 3.2 3B and Mistral 7B on dialogue tasks, examining their generalization, fluency, and fidelity to the target content. Differences in their loss, perplexity, BLEU, ROUGE, and BertScore metrics highlight their respective strengths in distinct usage contexts.

TABLE 3 . PERFORMANCE COMPARISON

	LLaMA 3.2 3b	**Mistral 7b**
Training Loss	0.74	0.58
Validation Loss	0.66	0.56
Perplexity	1.93	1.75
BLEU	18.75	16.7
ROUGE-1	0.25	0.20
ROUGE-2	0.16	0.10
ROUGE-L	0.23	0.19
BertScore	0.82	0.80

1) Generalization and Fluency

Mistral 7B stands out with lower training and validation losses and a reduced perplexity (1.75), indicating better fluency and adaptability to data variations. These results suggest that Mistral has a stronger generalization capacity, which can make its responses more natural and flexible across various contexts.

2) Content Fidelity

LLaMa 3.2 3B, although showing slightly higher perplexity, achieves higher BLEU, ROUGE, and BertScore values, reflecting its ability to align responses more precisely with target sequences. The ROUGE scores, in particular, indicate that LLaMa better captures the expected words and sequences, while the higher BertScore points to stronger semantic alignment. Therefore, LLaMa excels in applications that require closely matched responses to target content.

In summary, Mistral is ideal for more fluid and natural dialogues, whereas LLaMa is better suited for tasks requiring strict adherence to content and target phrasing.

D. Advantages

1) Relevance of Responses

The fine-tuned models demonstrate an ability to generate relevant and contextually appropriate responses, which is essential for maintaining engaging and informative conversations.

2) Engagement in Interactions

The models succeed in producing friendly and reassuring replies, fostering a positive interaction with the user.

3) Linguistic Adaptability

The models show an ability to understand and respond in dialect, highlighting their linguistic flexibility and adaptation to regional language forms. This allows for better alignment with users' cultural and linguistic expectations.

E. Limitations

1) Limitation in Lexical Matches

The moderate BLEU and ROUGE scores show that the models struggle with consistency at the bigram level. This limits their ability to reproduce sentence structures closely following the style and structure of the reference data, which can be a barrier in contexts requiring a strict reproduction of certain text sequences.

2) Heavy Model Training

Models based on LLaMA or Mistral are particularly resource-intensive and complex. This significantly slows down training and limits the ability to experiment with larger data volumes or more complex configurations.

3) Model Hallucination

The models present a notable risk of generating incorrect information, especially when dealing with topics where available data is insufficient or poorly structured. This phenomenon typically occurs in situations where models are forced to fill knowledge gaps with imprecise generated answers, which can compromise the reliability of the results, particularly in contexts requiring high accuracy.

F. Recommendations

1) Optimization of Exact Matches

To improve BLEU and ROUGE scores, it would be beneficial to optimize the models to better capture exact lexical matches. This can be achieved by adding more examples to the training dataset or fine-tuning lexical sequences to encourage better n-gram matching.

Another strategy could be targeted fine-tuning on specific data types to train the models to better reproduce specific structures. This could also include fine-tuning hyperparameters to encourage better retention of certain lexical patterns.

Finally, it would be wise to continue developing the models with a balance between semantic and lexical aspects. This could involve adjusting the loss function during training to emphasize both semantic accuracy and lexical coherence to produce more consistent results.

2) Improving Data Processing

A good alignment between the query and response is essential for an effective conversational model. This can be achieved through more thorough preprocessing, such as cleaning the data to remove noise, typographical errors, and ambiguities.

3) Extend Training by Increasing Epochs

The training and validation loss curves show potential for improvement with a higher number of epochs, suggesting that the model has not yet reached its optimal learning point. Continuing training could help the model strengthen its understanding of complex relationships within the data while maintaining its generalization ability. However, to avoid the risk of overfitting, where the model becomes too specific to the training data and loses performance on new data, it will be

essential to monitor the evolution of the loss curves. By fine-tuning the number of epochs, the model's efficiency and ability to respond accurately in various contexts can be optimized.

4) Increasing Computing Power

Logically, to address the issue of heavy model training, it would be necessary to increase computational power. This could include gaining access to more powerful infrastructures or optimizing code for better resource usage.

5) Introduction of RAG (Retrieval-Augmented Generation)

Integrating RAG could significantly enhance the quality of generated responses by allowing the model to access relevant information from external documents in the target language. This information retrieval mechanism helps mitigate the risk of hallucinations, which occur when the model generates answers based on inaccurate or unverified data. By relying on reliable databases, RAG enriches the text generation process with real-time factual information, ensuring more accurate and contextually appropriate responses. This approach also promotes a better match between the questions asked and the information provided, thereby enhancing the relevance of the responses.

VII. CONCLUSION

This paper explored the adaptation of pre-trained language models LLaMA and Mistral for the development of an informal Malagasy dialogue system. The analysis of the results highlighted significant progress in the models' ability to adapt to this low-resource language, notably through specific pre-training and meticulous fine-tuning.

The pre-training results showed that the loss significantly decreased after 50 epochs, suggesting a strong adaptability of the models to Malagasy linguistic structures. The perplexity also dropped, indicating an improvement in the models' ability to generate coherent and relevant word sequences. This reflects the effectiveness of the fine-tuning process and the mastery of linguistic variations characteristic of informal Malagasy.

The training of the conversational agents revealed promising performance with low loss for both training and testing, suggesting that the models generalize well to informal Malagasy dialogues. The measured perplexity and additional metric scores (BertScore, BLEU, ROUGE) also indicated that the models are capable of generating coherent responses, although adjustments are needed to improve lexical and syntactic alignment.

When comparing the performance of LLaMA 3.2 3B and Mistral 7B, it emerged that Mistral excels in fluency and generalization, with lower loss and perplexity scores, making it more flexible in varied contexts. In contrast, LLaMA, though slightly less fluent, performed better in terms of content fidelity, with higher BLEU, ROUGE, and BertScore scores, illustrating its ability to better align its responses with target data. These results suggest that Mistral is more suited for natural, fluid dialogues, while LLaMA proves more effective for tasks requiring a strict match with expected sequences.

Overall, this work demonstrates the value of adapting pre-trained language models to low-resource languages such as informal Malagasy, and paves the way for future improvements, such as optimizing lexical matches and integrating advanced techniques like Retrieval-Augmented Generation (RAG).

ACKNOWLEDGMENT

The author would like to express sincere gratitude to the Politehnica University of Timișoara (UPT)[6] for the warm welcome and the logistical support provided during the research stay. Special thanks are also extended to the research supervisor for valuable guidance and support, as well as to the research teams for their collaboration and availability throughout this period.

REFERENCES

[1] H. Shum, X. He, et D. Li, « From Eliza to XiaoIce: challenges and opportunities with social chatbots », *Frontiers Inf Technol Electronic Eng*, vol. 19, nº 1, p. 10-26, janv. 2018, doi: 10.1631/FITEE.1700826.

[2] A. Vaswani, « Attention is all you need », *Advances in Neural Information Processing Systems*, 2017.

[3] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, et I. Sutskever, « Language models are unsupervised multitask learners », *OpenAI blog*, vol. 1, nº 8, p. 9, 2019.

[4] B. P. King, « Practical Natural Language Processing for Low-Resource Languages. », PhD Thesis, 2015.

[5] H. Touvron *et al.*, « Llama: Open and efficient foundation language models », *arXiv preprint arXiv:2302.13971*, 2023.

[6] Q. J. Albert, A. Sablayrolles, A. Mensch, C. Bamford, et D. S. Chaplot, « Mistral 7B », *arXiv*, 2023.

[7] H. Zhang, H. Song, S. Li, M. Zhou, et D. Song, « A Survey of Controllable Text Generation Using Transformer-based Pre-trained Language Models », *ACM Comput. Surv.*, vol. 56, nº 3, p. 1-37, mars 2024, doi: 10.1145/3617680.

[8] R. Sennrich, « Neural Machine Translation », *Institute for Language, Cognition and Computation University of Edinburgh*, vol. 18, 2016, Consulté le: 2 janvier 2025.

[9] F. Rakotomalala, A. R. Hajalalaina, M. V. Ravonimanantsoa Ndaohialy, A. Andriavelonera Alexandre, et A. H. Ranaivoson, « FLICs (Facebook Language Informal Corpus): a novel dataset for informal language », *Int J Data Sci Anal*, vol. 18, nº 4, p. 393-403, oct. 2024, doi: 10.1007/s41060-023-00460-2.

[6] https://www.upt.ro/

[10] A. Bendale, M. Sapienza, S. Ripplinger, S. Gibbs, J. Lee, et P. Mistry, « SUTRA: Scalable Multilingual Language Model Architecture », 7 mai 2024, *arXiv*: arXiv:2405.06694. doi: 10.48550/arXiv.2405.06694.

[11] B. Zhang et R. Sennrich, « Root mean square layer normalization », *Advances in Neural Information Processing Systems*, vol. 32, 2019, Consulté le: 12 mai 2025.

[12] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo, et Y. Liu, « Roformer: Enhanced transformer with rotary position embedding », *Neurocomputing*, vol. 568, p. 127063, 2024.

[13] N. Shazeer, « Fast Transformer Decoding: One Write-Head is All You Need », 6 novembre 2019, *arXiv*: arXiv:1911.02150. doi: 10.48550/arXiv.1911.02150.

[14] N. Shazeer, « GLU Variants Improve Transformer », 12 février 2020, arXiv: arXiv:2002.05202. doi: 10.48550/arXiv.2002.05202.

[15] A. Balachandran, « Tamil-Llama: A New Tamil Language Model Based on Llama 2 », 10 novembre 2023, arXiv: arXiv:2311.05845. doi: 10.48550/arXiv.2311.05845.