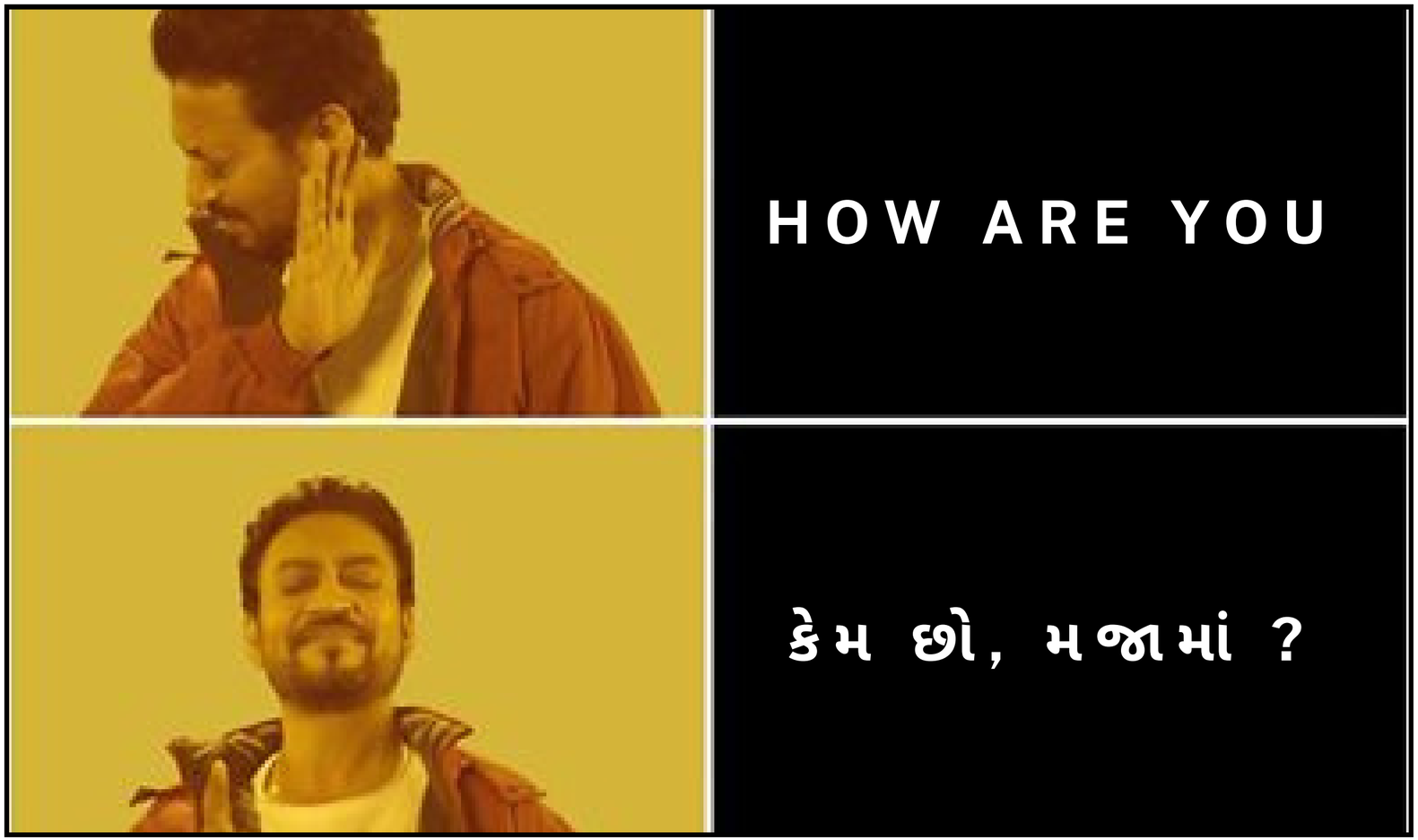
GUJJU LLAMA

ENHANCING LLMS WITH VERNACULAR LANGUAGE SUPPORT

यह धारावाहिक सत्य घटना पर आधारित है

WHY SETTLE FOR 'HOW ARE YOU' WHEN YOUR LLM CAN SAY 'કેમ છો, મજામાં ?' TOO?



A G E N D A

- Our Team
- Motivation
- Road Map
- Tokenization Training
- Pre-training
- Data Prepare
- Fine-tuning
- Benchmarking
- Challenges
- Overcoming
- CPT

MEET THE TEAM

Khyat Anjaria
ML Engineer
Rootle.ai
Linkedin

Dixit Trivedi
ML Engineer
Rootle.ai
Linkedin

Dhruv Bhatnagar
Research Engineer (ML)
Reverie Language Technologies
Linkedin

CURRENT LANDSCAPE OF INDIC AI

कृत्रिम
India's own AI

KissanAI

sarvam.ai

BharatGPT

Español Spanish
தமிழ் Tamil
हिंदी Hindi
Deutsch German
ગુજરાતી Gujarati
한국어 Korean
മലയാളം Malayalam
Türkçe Turkish
Ελληνικά Greek

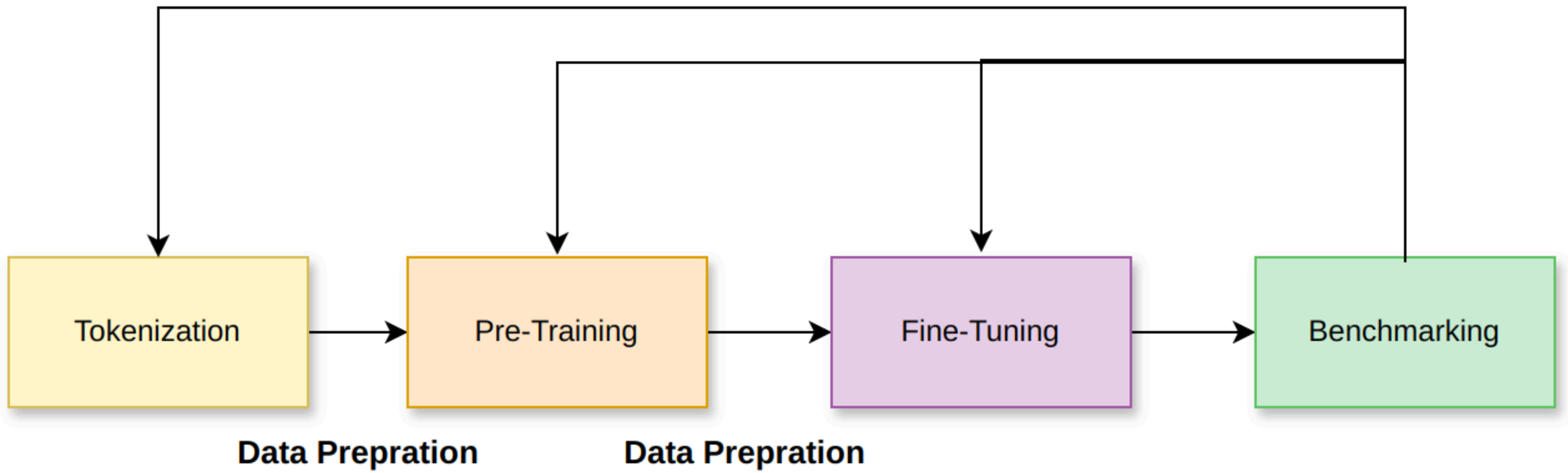
SUTRA

Polish Polski
Greek Ελληνικά
Vietnamese Tiếng Việt
French Français
Thai ไทย
Bengali বাংলা
Gujarati ગુજરાતી
Arabic العربية
Japanese 日本語

CHALLENGES IN DEVELOPING LLMS FOR INDIAN LANGUAGES:

- ***Lack of high quality datasets***
  - Although we can use large scale web crawls from websites like wikipedia for pretraining. the finetuning stage requires carefully crafted instruction datasets which are hard to find.
- ***Limited Vocabulary of Existing models***
  - The vocabularies of most large language models barely have any word/sub-words from under represented languages, which significantly hampers their text generation capabilities, already limited in efficiency.
- ***Lack of evaluation of benchmarks***
  - Creating an LLM for under represented languages like Gujarati is a challenging endaveour, and assessing their performance poses a significant challenge since there are no standardized benchmarks specifically designed for Gujarati language.

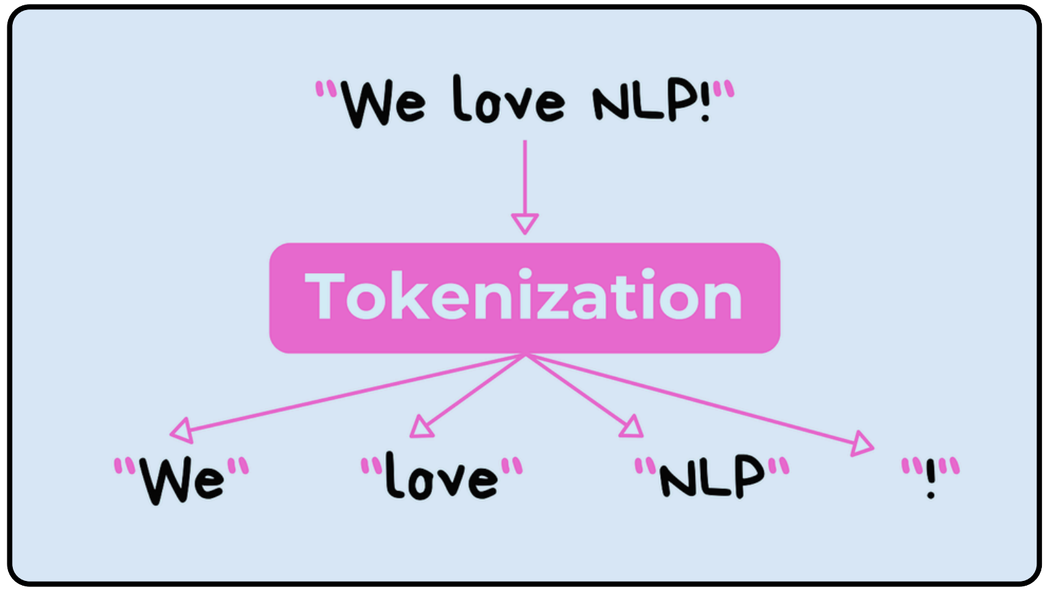
R O A D M A P


Tokenization
Pre-Training
Fine-Tuning
Benchmarking
Data Prepration
Data Prepration

“Tokenization”

TOKENIZATION

- Tokenization is the process of converting text into smaller units (tokens), which can be words, subwords, or characters, that models can process.

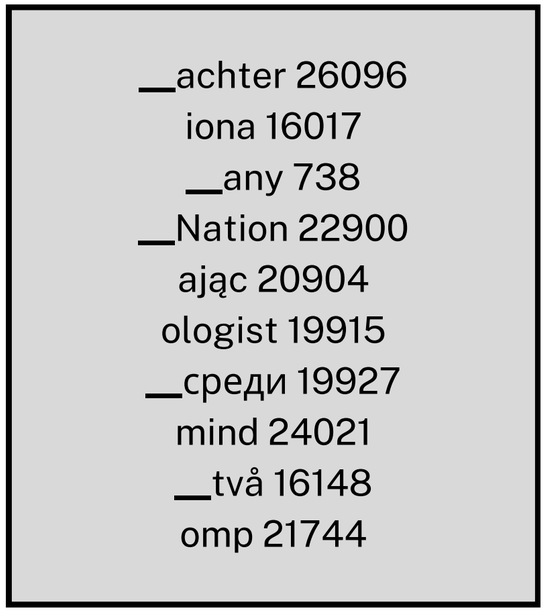


TOKENIZATION METHODS

- **SentencePiece**
- Tiktoken
- WordPiece
- Byte-Pair Encoding (BPE)
- Whitespace Tokenizer
- Huggingface Tokenization (Fast Tokenizer)
- N-Gram Tokenization

VOCAB FILE FOR TOKENIZATION

- A vocabulary (vocab) file is a list of tokens (words, subwords, or characters) that a language model uses during tokenization.
- Each token is assigned a unique ID used by the model for computations.
- The vocab file ensures consistency in how text is represented across training data.
- Example:
  - Text: "The quick brown fox"
  - Tokens: ["The", "quick", "brown", "fox"]
  - Token IDs: [512, 1033, 981, 284]



OUT OF VOCABULARY

- Unicode Transformation Format - 8 bit (UTF-8)
- Character encoding
- Every character of every languages has their

UTF-8 format

- The model can’t understand UTF-8 format.

['__', '<0xE0>', '<0xAA>', '<0xB6>', '<0xE0>', '<0xAB>', '<0x81>', '<0xE0>', '<0xAA>', '<0x82>', '__', '<0xE0>', '<0xAA>', '<0xAD>', 'ા', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAA>', '<0xA4>', '__', '<0xE0>', '<0xAA>', '<0xB5>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xB2>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xA1>', '__', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAA>', '<0xAA>', '__', '<0xE0>', '<0xAA>', '<0x9F>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAB>', '<0x8B>', '<0xE0>', '<0xAA>', '<0xAB>', '<0xE0>', '<0xAB>', '<0x80>', '__', '<0xE0>', '<0xAA>', '<0xB8>', '<0xE0>', '<0xAB>', '<0x81>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xB7>', '<0xE0>', '<0xAA>', '<0xBF>', '<0xE0>', '<0xAA>', '<0xA4>', '__', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAB>', '<0x80>', '__', '<0xE0>', '<0xAA>', '<0xB6>', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAA>', '<0xB6>', '<0xE0>', '<0xAB>', '<0x87>', '?', '<0x0A>', 'Can', '__India', '__secure', '__the', '__World', '__Cup', '__tro', 'phy', '?']

શું ભારત વર્લ્ડ કપ ટ્રોફી સુરક્ષિત કરી શકશે? Can India secure the World Cup trophy?

TOKEN DISTRIBUTION BY LANGUAGE IN LLAMA 2

Language	Tokens	Percentage
English	27007	84.40%
Tamil	19	0.06%
Hindi	39	0.12%
Marathi	39	0.12%
Gujarati	1	0.00%

SENTENCEPIECE TOKENIZATION TRAINING

- Why We Need
- Data Requirements
- Parameters
    - Vocab Size (Example: 16000, 20000, 3000)
    - Model Type (Example: Unigram/ BPE)
    - Character Coverage (Example: 0.0-1.0)
- Llama2 Tokenizer: 32000 Tokens
- Gujju Llama Tokenizer: 48000 Tokens

TOKENIZER DIFFERENCE

શું ભારત વર્લ્ડ કપ ટ્રોફી સુરક્ષિત કરી શકશે? Can India secure the World Cup trophy?

['__', '<0xE0>', '<0xAA>', '<0xB6>', '<0xE0>', '<0xAB>', '<0x81>', '<0xE0>', '<0xAA>', '<0x82>', '__', '<0xE0>', '<0xAA>', '<0xAD>', 'ા', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAA>', '<0xA4>', '__', '<0xE0>', '<0xAA>', '<0xB5>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xB2>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xA1>', '__', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAA>', '<0xAA>', '__', '<0xE0>', '<0xAA>', '<0x9F>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAB>', '<0x8B>', '<0xE0>', '<0xAA>', '<0xAB>', '<0xE0>', '<0xAB>', '<0x80>', '__', '<0xE0>', '<0xAA>', '<0xB8>', '<0xE0>', '<0xAB>', '<0x81>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAB>', '<0x8D>', '<0xE0>', '<0xAA>', '<0xB7>', '<0xE0>', '<0xAA>', '<0xBF>', '<0xE0>', '<0xAA>', '<0xA4>', '__', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAA>', '<0xB0>', '<0xE0>', '<0xAB>', '<0x80>', '__', '<0xE0>', '<0xAA>', '<0xB6>', '<0xE0>', '<0xAA>', '<0x95>', '<0xE0>', '<0xAA>', '<0xB6>', '<0xE0>', '<0xAB>', '<0x87>', '?', '<0x0A>', 'Can', '__India', '__secure', '__the', '__World', '__Cup', '__tro', 'phy', '?']

LLaMA tokenizer n_tokens=125

['શું', '__ભારત', '__વર્લ્ડ', '__કપ', '__ટ્ર', 'ોફી', '__સુરક્ષિત', '__કરી', '__શકશે', '?', '<0x0A>', 'Can', '__India', '__secure', '__the', '__World', '__Cup', '__tro', 'phy', '?']

Gujju LLaMA tokenizer n_tokens=21

“In the world of AI,
TOKENS are the small
pieces that make big ideas
possible.”

“Pre-Training”

PRE-TRAINING

- Pre-training is the process of training a model on humongous data to learn the structure, grammar, and semantics of a language, enabling it to generate coherent text, perform translations, and answer questions.

- Why We Need
    - Language adaptation
    - Domain adaptation
    - Preventing catastrophic forgetting
    - No labelled data

PRE-TRAINING ALGORITHMS

- **Causal Language Modeling (CLM) - GPT-3, GPT-4, Llama 2**
- Masked Language Modeling (MLM) - BERT, RoBERTa, ALBERT
- Permuted Language Modeling (PLM) - XLnet
- Denoising Autoencoding - DeBERTa
- Next Sentence Prediction (NSP) - BERT
- Span Masking - SpanBERT
- Sequence-to-Sequence Pre-Training - T5, mT5

COMPUTATION DEVICE

- **Platform**: Runpod Cloud
- **GPU**: NVIDIA A100
    - **VRAM**: 80 GB
- **Memory**: 117 GB RAM
- **Runpod Website**: https://www.runpod.io/

RunPod

DATA UTILIZATION

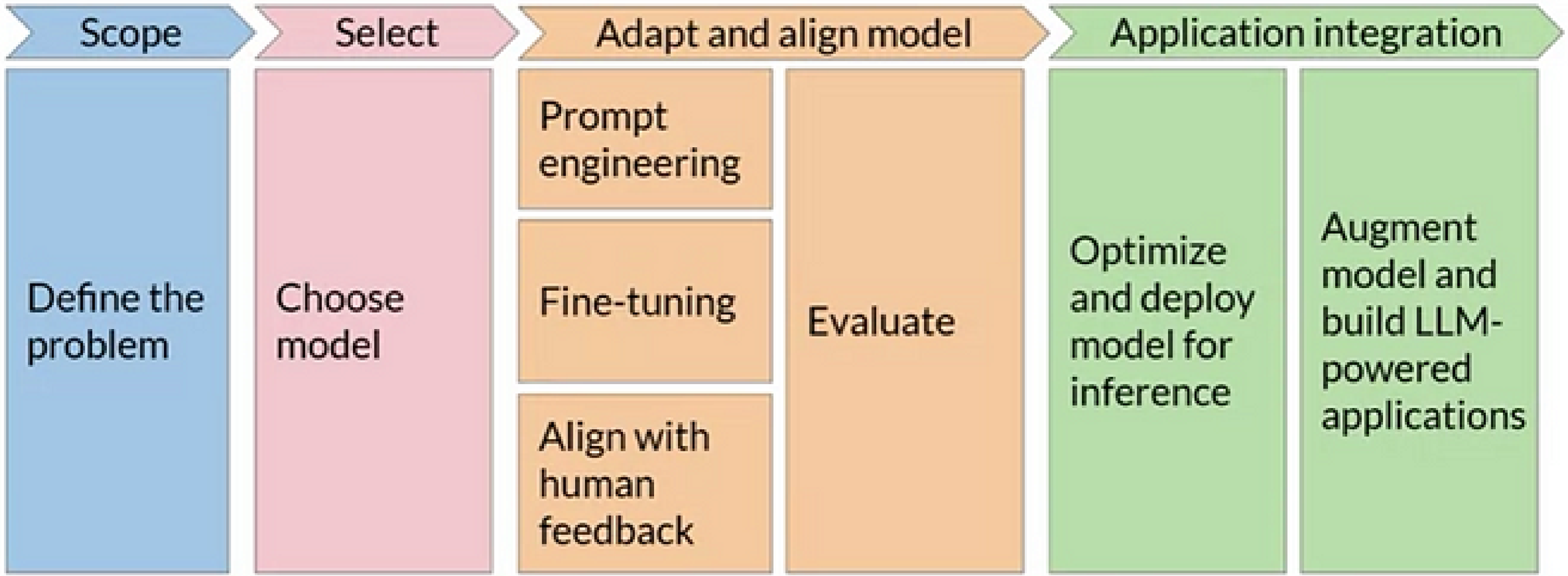
- **CulturaX:**
  - **Stored Multi-languages dataset**
  - **Dataset Link:** https://huggingface.co/datasets/uonlp/CulturaX/viewer/gu
  - **Total Size**: 1.1 Million Documents
  - **Utilized**: 500k Documents

UNDER THE HOOD OF CLM

- Start
- Tokenization: Tokenize sentence into words
- Context Preparation: Context: Words before the current word
- Prediction: Use context to predict the next word
- Generate Sequence:
  - Append predicted word to sequence
  - Update context with the new word
- Loss Calculation (Training Phase)
  - Compare predicted sequence with actual words
  - Calculate loss and update model parameters
- Repeat: Continue until the sequence is complete (during generation)
- End

“Fine-Tuning”

WORKFLOW FOR FINETUNING



PRACTICAL APPROACH OF INSTRUCTION TUNING

- Figure out your task.
- Collect data related to the task's inputs/outputs.
- Generate data if you don't have enough data.
- Finetune a Small Model (e.g. 400M - 1B).
- Vary the amount of data you give the model.
- Evaluate your LLM to what's going well vs what not.
- Collect more data to improve.
- Increase task complexity.
- Increase model size for performance.

HOW TO PREPARE PROMPT TEMPLATES

Classification/sentiment analysis

“Given the following review:

{{review_body}}

predict the associated rating from the following choices (1 being lowest and 5 being highest)”

Text generation

“Generate a {{star_rating}}-star review (1 being lowest and 5 being highest) about this product {{product_title}}. {{review_body}}”

Text summarization

"Give a short sentence describing the following product review:

{{review_body}}

{{review_headline}}"

INSTRUCTION TUNING

- **Key Aspects of Instruction Tuning:**
  - **Pre-trained Base Model:** The model is initially pre-trained on large corpora, which allows it to have a general understanding of language patterns.
  - **Task Agnostic:** Unlike task-specific fine-tuning, instruction tuning does not focus on one specific task. Instead, it is tuned to handle a variety of tasks, making the model more versatile.
  - **Instruction-Based Training Data:** During instruction tuning, the training data consists of input-output pairs where the input includes both the task description (instruction) and any relevant context. For example, “Translate the following text into Gujarati: [text].”
  - **Improved Generalization:** By training the model with explicit instructions, it becomes better at understanding and following commands from users, even for tasks that were not directly seen during training.

EXAMPLE USE CASE OF IFT:

- ***Input***: “Summarize the following paragraph: [text].”
- ***Model Output***: A summarized version of the text.

- ***Input***: “Classify the sentiment of the following review: [review text].”
- ***Model Output***: The sentiment label, such as "positive" or "negative

SUPERVISED FINE-TUNING

- **Key Aspects of Supervised Fine-Tuning:**
  - **Label-Based Training:** The model learns from input-output pairs where the correct output (label) is provided for each input. It uses supervised learning methods such as cross-entropy loss to optimize model predictions.
  - **Task-Specific Supervision:** The supervision is tailored to a specific task, such as question answering, summarization, or classification.
  - **Selective Parameter Updating:** While SFT often updates all parameters, it can also be done using methods like LoRA or adapters to fine-tune only a subset of parameters to reduce computational cost.

EXAMPLE USE CASE OF SFT:

- ***Input*:** “Chandrayaan-2 was launched from Sriharikota on July 22, 2019.”
- ***Expected Output*:** "Chandrayaan-2" → MISSION_NAME, "Sriharikota" → **LAUNCH_SITE,** "July 22, 2019" → DATE.

FULL FINE-TUNING

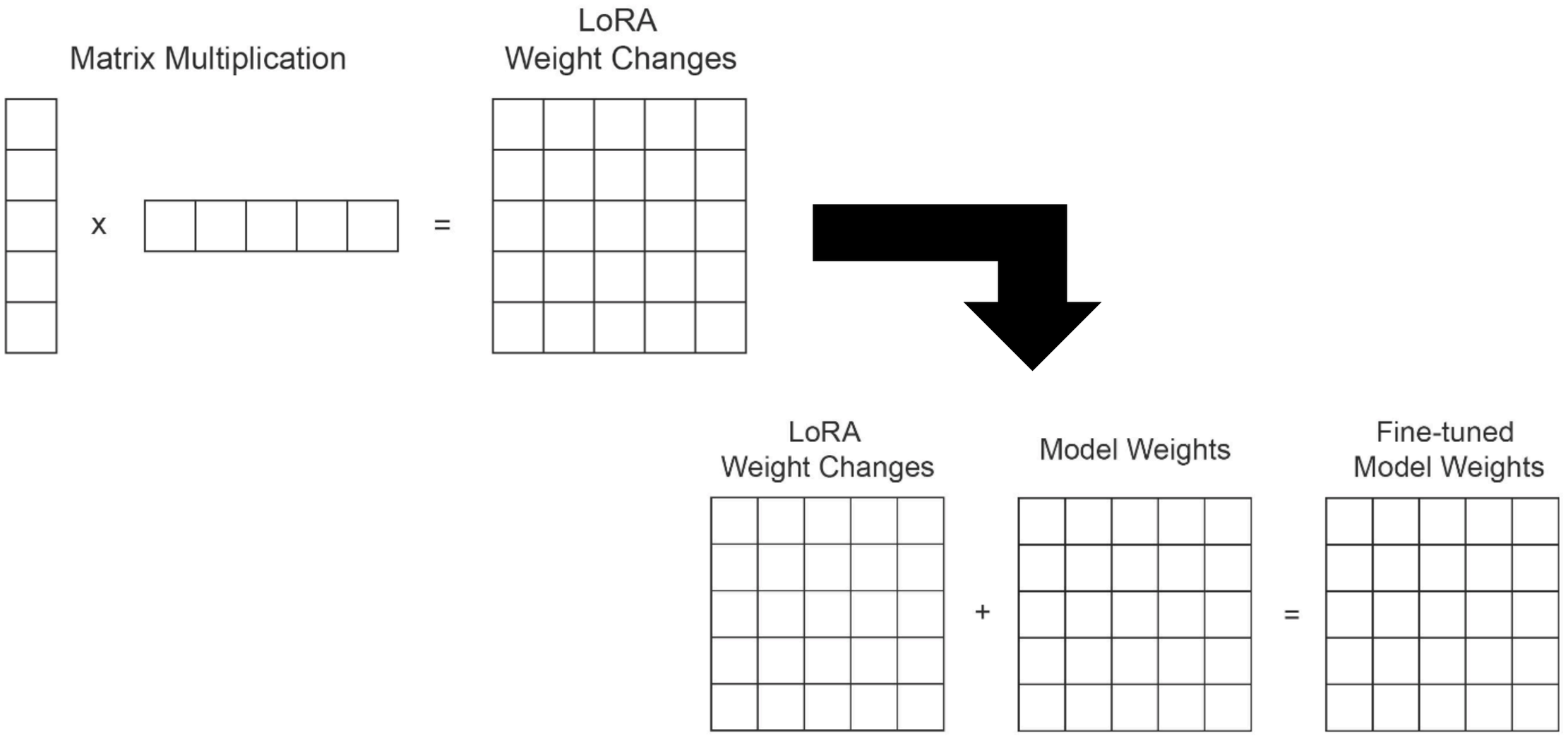
- **Key Aspects of Full Fine-Tuning:**
  - **All Parameters Updated:** Unlike other fine-tuning methods, full fine-tuning adjusts all layers and weights in the model.
  - **Task-Specific Training:** The model is optimized for a specific downstream task, such as sentiment analysis, machine translation, or text generation, using supervised or unsupervised learning methods.
  - **Requires Large Data:** Full fine-tuning often requires a significant amount of labeled data specific to the task in question to avoid overfitting.
  - **Computationally Expensive:** Since all model parameters are being updated, full fine-tuning is resource-intensive in terms of both time and hardware.

EXAMPLE USE CASE OF FFT:

- ***Input:*** “The patient shows symptoms of [description].”
- ***Label:*** “Condition A”
- ***The model is fine-tuned to predict medical conditions based on input text.***

“LoRA in Everyday Life: Learning a New Skill without Overloading Your Brain.”

HOW LORA WORKS



INCREASING THE PRECISION OF LORA WITH RANK

Rank	7B	13B	70B	180B
1	167,332	228,035	529,150	848,528
2	334,664	456,070	1,058,301	1,697,056
4	669,328	912,140	2,116,601	3,394,113
8	1,338,656	1,824,281	4,233,202	6,788,225
16	2,677,312	3,648,561	8,466,404	13,576,450
512	85,673,987	116,753,964	270,924,934	434,446,406

Rank	7B	13B	70B	180B
1	0.002%	0.002%	0.001%	0.000%
2	0.005%	0.004%	0.002%	0.001%
4	0.010%	0.007%	0.003%	0.002%
8	0.019%	0.014%	0.006%	0.004%
16	0.038%	0.028%	0.012%	0.008%
512	1.224%	0.898%	0.387%	0.241%

CHALLENGES & POSSIBLE SOLUTIONS

- ***Hallucination***
  - User Input: "Tell me about the Eiffel Tower in London."
  - Model Response: "The Eiffel Tower in London is a famous historical landmark near the River Thames."
  - Issue: The model hallucinates that the Eiffel Tower is in London instead of Paris.
- ***Repeated Token Generation***
  - User Input: "Generate a short story about a hero."
  - Model Response: "The hero was brave, brave, brave, brave, brave, and everyone loved, loved, loved, loved him."
  - Issue: The model repeatedly generates the same tokens, resulting in nonsensical output.
- ***Catastrophic Forgetting***
- ***Latency and Efficiency***

BENCHMARKS: LLAMA2-BASE VS. GUJJU LLAMA

METRIC	LLAMA 2 - BASE	GUJJU LLAMA
MMLU	42.5	48.6
ARC-Easy	0.39	0.43
ARC-Challenge	0.22	0.25
Boolq	0.69	0.72
Hellaswag	0.38	0.40

GENERATED BY GUJJU LLAMA

SUBJECT: રાજીનામા... માટે, અરજી / RESIGNATION APPLICAT...

મોહત્યાર સાહેબ...

મારે તમારું તમારું આભાર માનવું છે કે તમે, તમે મને આ સંસ્થામાં સંસ્થામાં કામ કરવાની તક... આપી. હું, હું, તમારા માર્ગદર્શન હેઠળ ઘણું ઘણું શીખ્યો છું. જો કે, હાલની હાલની પરિસ્થિતિ પરિસ્થિતિઓને ધ્યાનમાં રાખીને, હું હું મારી જવાબદારીથી રાજીનામું રાજીનામું આપવું છું.
મારા રાજીનામાનો અમલ તારીખ (કોઈ તારીખ) થી માન્ય માન્ય રહેશે રહેશે. મને આ આ સંસ્થામાં સંસ્થામાં કામ કામ કરવું ખુબ ગમ્યું, અને મેં મારી શ્રેષ્ઠ શ્રેષ્ઠ ક્ષમતા ક્ષમતા મુજબ બધું કર્યું છે. હું આ છોડી રહ્યો છું... મારા મારા સ્વપ્નને સ્વપ્નને આગળ આગળ વધાવવા માટે અને નવા નવા મોકાઓ મેળવવા માટે.

આભાર આભાર, તમારાં તમારાં સહયોગ માટે અને માર્ગદર્શન માટે...

ધન્યવાદ,
[તમારું નામ]

WHY ASK 'CAN YOU SPEAK GUJARATI?' WHEN OUR LLM'S ALREADY SAYING, 'હા ભાઈ, બોલી રહ્યો છું!!'

આભાર!