Towards Small Language Model

Nelie MAKENNE, Prof. Ashwin ITTOO
HEC Liège - ULiège
November 14, 2024

HEC LIÈGE
Management School - Liège Université

Short Bio

Past Academic Degree's:

- Master of Science in Computer Science and Mathematical Science, University of Dschang/ AIMS Cameroon, 2018-2020.
- Two years Research's Master, AIMS Rwanda, 2021-2023.

Ongoing Phd Journey

- Start: May 2023
- Duration: 4 Years
- Promotor: Prof. Ashwin ITTOO
- Industry Partner : Partenamut Insurance

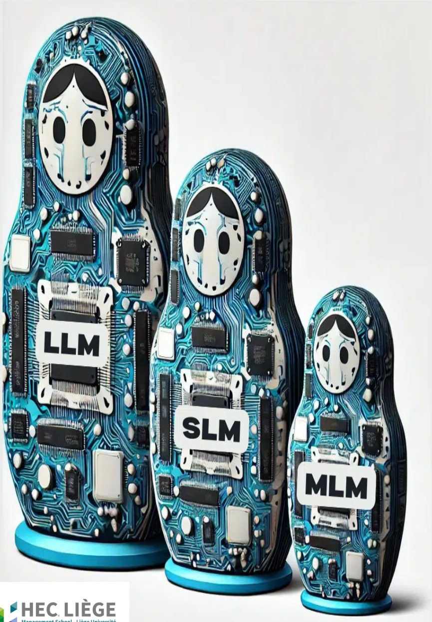
About My Research

- **Goal**: Automate customer interaction via a virtual conversational agent
- **Challenge & Constraints**
  1. Handling multiple languages
  2. Limited Resources

- **SOTA in NLP By today**: Large Language Model (LLM)
  - Trained on massive data from multiple sources in multiple languages *(challenge 1 solved)*
  - Ability to understand and generate human-like language
  - Multi-tasks: text generation, translation, question answering, summarization, and more.

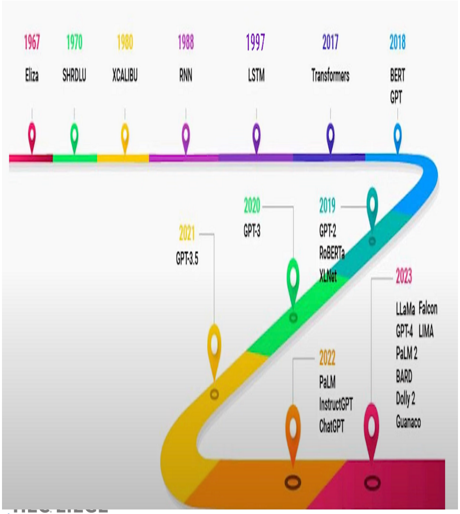
Research Question

How to compress LLM without compromising performance?


LLM
SLM
MLM

contents

Evolution of NLP and Large Language Models



Early Rule-Based and Symbolic Systems (1960s–1980s)

- **1967: Eliza** – One of the earliest NLP programs, designed to simulate conversation.
- **1970: SHRDLU** – A program that could execute commands in a "blocks world," demonstrating understanding of structured commands.
- **1980: XCALIBU** – A lesser-known system contributing to early AI advancements, possibly related to expert systems or knowledge representation.

Introduction of Neural Networks (1980s–1990s)

- **1988: RNN (Recurrent Neural Networks)** – Introduced sequential processing, enabling models to work with time-dependent data.
- **1997: LSTM (Long Short-Term Memory)** – A specialized RNN architecture designed to overcome issues with long-term dependencies in data, making it more effective for tasks like language modeling.

The Transformer Era and Breakthroughs (2017–2020)

- **2017: Transformers** – Revolutionized NLP by introducing self-attention, allowing models to process data in parallel and capture long-range dependencies.
- **2018: BERT, GPT** – BERT (Bidirectional Encoder Representations from Transformers) focused on understanding language context, while GPT (Generative Pretrained Transformer) focused on text generation.
- **2019: GPT-2, RoBERTa, XLNet** – Enhanced transformer models with better language generation and comprehension abilities.

Large-Scale Language Models and Fine-Tuning (2020–2022)

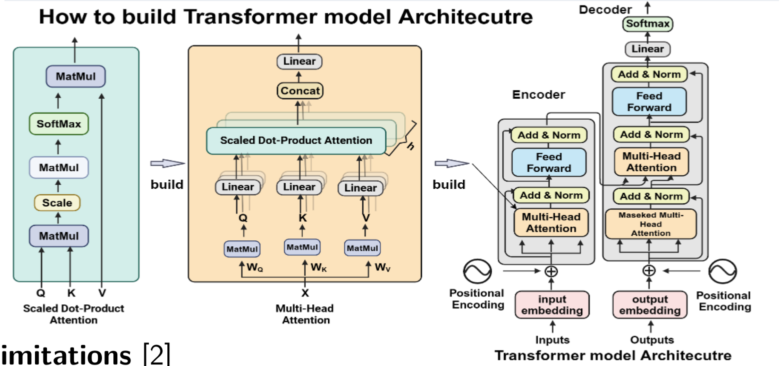
- **2020: GPT-3** – Known for its massive scale and ability to perform few-shot learning, GPT-3 demonstrated the power of large language models in a wide range of tasks.
- **2021: GPT-3.5** – An improved version of GPT-3 with better instruction-following capabilities.
- **2022: PaLM, InstructGPT, ChatGPT** – Focused on fine-tuning models for more effective conversational AI and instruction-following tasks.

State-of-the-Art and Specialized Models (2023)

- **2023: LLaMA, GPT-4, Falcon, LIMA, PaLM 2, BARD, Dolly 2, Guanaco** – The most recent models emphasizing scale, efficiency, and specialization for various NLP applications.

contents

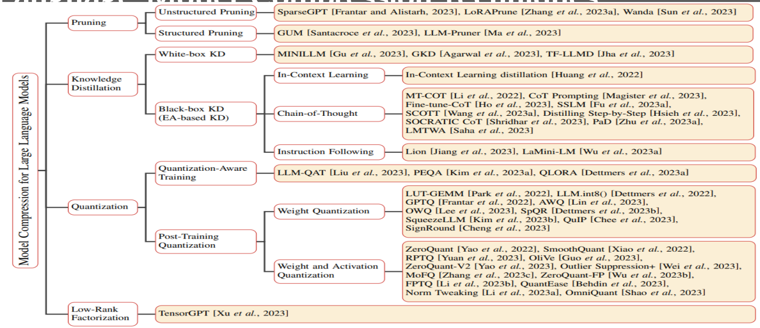
Architecture of Large Language Models



Limitations [2]

- High computational cost for training and inference
- Potential biases in model outputs due to training data
- Difficulty in interpreting and explaining model decisions
- Requires large datasets for effective performance

Large Language Model Compression techniques



Limitations [1]

- **LoRA**: Freezing weights and adapting the model can limit its ability to fully capture new patterns, potentially impacting accuracy.
- **Pruning**: Aggressively removing weights can lead to significant information and accuracy loss.
- **Quantization**: Reducing parameter precision can result in computational overhead and may degrade model performance.
- **Knowledge distillation**: Training a small model from scratch to mimic a larger model can consume considerable energy.

contents

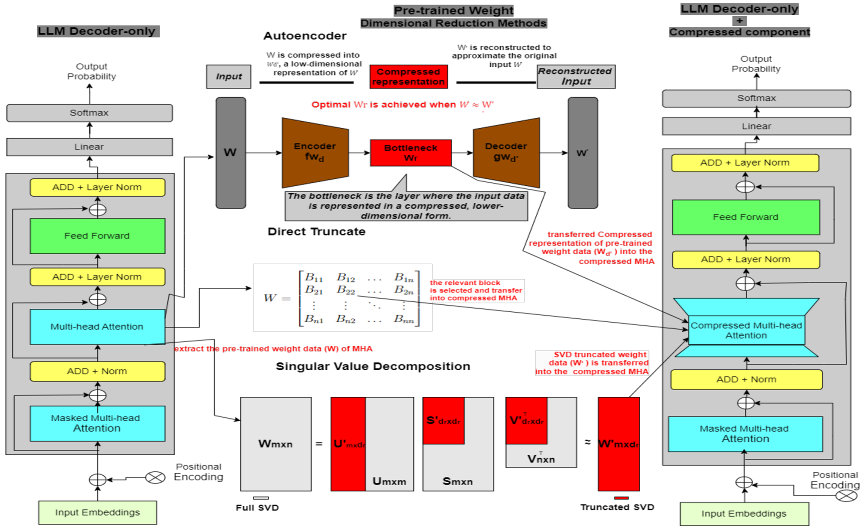
Proposed Method for LLM Size Reduction

- **Our Hypothesis**
  - Not all information's learned during training is necessary for specific tasks
- **Our Approach**
  - Reduce model size by selecting only the most relevant features from the weights.

- **Methods**
  - Direct Truncate
  - Singular Value Decomposition (SVD) [4]
  - Auto-encoder [3]

HEC LIÈGE Management School - Liège Université

Proposed Method for LLM Size Reduction



Truncated SVD for Matrix Size Reduction

- Only the top d_r largest singular values and their corresponding singular vectors are retaining:

$$W' = U_{d_r} S_{d_r} V_{d_r}^T$$

- This truncation captures the most significant information, reducing the model size by approximating W with W'.

- Large singular values capture the most significant patterns and variations in the data,
- The truncated matrix W' reduces the the number of parameters and the computational cost.

HEC LIÈGE Management School - Liège Université

contents

Model Baseline

- **Model Source**: Hugging Face
- **Working Environment**: CECI Lucia (60GB RAM, 1x NVIDIA A100 40GB)
- **Model Name**: LLaMA-3 (8 billion parameters)
- **Number of parameters**
  - Multi-head attention layers (MHA): **1,342,177,280**
  - Feed-forward network (MLP) layers: 5,637,144,576
  - Transformer block (MHA+MLP) : 6,979,584,000
  - Model: 8,030,261,248

- **Model configuration**
  - Embedding size: 4096
  - Number of attention heads: 32
  - number of key/value heads: 8
  - number of hidden layers: 32
- **LLM Component Size Reduction**:

  Multi-head Attention
  - **Query weight matrix dimension** is (embedding dim, query dim)
  - **Value weight matrix dimension** is (embedding dim, value dim)
  - **Key weight matrix dimension** is (embedding dim, key dim)
  - **Output weight matrix dimension** is (output dim, embedding dim)

Result

$d_r = \alpha d$ with $\alpha \in (0, 1)$ and $W_r = SVD(W)$, with $d_r = 1024 = 4096 \times 0.25$

Original Model Architecture

```
LlamaModel(
  (embed_tokens): Embedding(128256, 4096)
  (layers): ModuleList(
    (0-31): 32 x LlamaDecoderLayer(
      (self_attn): LlamaSdpaAttention(
        (q_proj): Linear(in_features=4096, out_features=4096, bias=False)
        (k_proj): Linear(in_features=4096, out_features=1024, bias=False)
        (v_proj): Linear(in_features=4096, out_features=1024, bias=False)
        (o_proj): Linear(in_features=4096, out_features=4096, bias=False)
        (rotary_emb): LlamaRotaryEmbedding()
      )
```

Compressed Model Architecture

```
CustomLlamaForCausalLM(
  (model): CustomLlamaModel(
    (embed_tokens): Embedding(128256, 4096)
    (layers): ModuleList(
      (0-31): 32 x CustomLlamaDecoderLayer(
        (self_attn): LlamaSdpaAttention(
          (q_proj): Linear(in_features=4096, out_features=1024, bias=False)
          (k_proj): Linear(in_features=4096, out_features=256, bias=False)
          (v_proj): Linear(in_features=4096, out_features=256, bias=False)
          (o_proj): Linear(in_features=1024, out_features=4096, bias=False)
          (rotary_emb): LlamaRotaryEmbedding()
        )
```

So, we retain 25% of the parameter features. Therefore, the total number of parameters MHA : $0.25 \times 1,342,177,280 =$**335,544,320**.

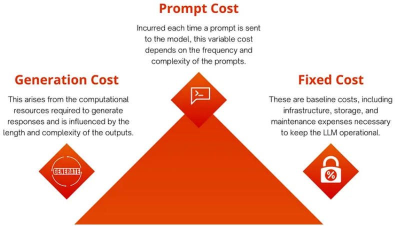
HEC LIÈGE
Management School - Liège Université

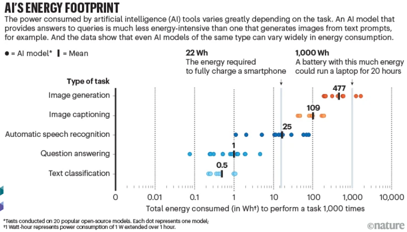
Result

- **Fine-tune:** Multi-head attention Layer using LoRA methods (rank=8).
- **Training dataset**: A set of 818 customer service requests from users of an online selling app and the corresponding intentions behind each request from Kaggle.
- **Evaluation metric**: Accuracy, F1 Score, Human Evaluation
- **Model Performance**

Dataset size	200
Accuracy	43%
F1 Score	50,5%

Why Move Towards Smaller Models?





- **For Society implication [5]**
  - Reduce infrastructure and operational costs.
  - Accessibility for researchers and small organizations.
- **For climate Change [6]**
  - Lower energy consumption.
  - Carbon Footprint Efficiency

Amazon EC2 G5 Instances Price

	Instance Size	GPU	GPU Memory (GiB)	vCPUs	Memory (GiB)	Storage (GB)	Network Bandwidth (Gbps)	EBS Bandwidth (Gbps)	On Demand Price/hr*	1-yr ISP Effective Hourly (Linux)	3-yr ISP Effective Hourly (Linux)
Single GPU VMs	g5.xlarge	1	24	4	16	1x250	Up to 10	Up to 3.5	$1.006	$0.604	$0.402
	g5.2xlarge	1	24	8	32	1x450	Up to 10	Up to 3.5	$1.212	$0.727	$0.485
	g5.4xlarge	1	24	16	64	1x600	Up to 25	8	$1.624	$0.974	$0.650
	g5.8xlarge	1	24	32	128	1x900	25	16	$2.448	$1.469	$0.979
	g5.16xlarge	1	24	64	256	1x1900	25	16	$4.096	$2.458	$1.638
Multi GPU VMs	g5.12xlarge	4	96	48	192	1x3800	40	16	$5.672	$3.403	$2.269
	g5.24xlarge	4	96	96	384	1x3800	50	19	$8.144	$4.886	$3.258
	g5.48xlarge	8	192	192	768	2x3800	100	19	$16.288	$9.773	$6.515

HEC LIÈGE Management School - Liège Université

References

[1] Zhu, Xunyu and Li, Jian and Liu, Yong and Ma, Can and Wang, Weiping. *A survey on model compression for large language models*. arXiv preprint arXiv:2308.07633, 2023.

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

[3] Wang, Yasi and Yao, Hongxun and Zhao, Sicheng, *Auto-encoder based dimensionality reduction*, Neurocomputing, 2016.

[4] Holmes, Michael and Gray, Alexander and Isbell, Charles, *Fast SVD for large-scale matrices*, Workshop on Efficient Machine Learning at NIPS,2007

[5] https://medium.com/lohith_gn/cost-optimization-in-generative-ai-strategies-for-llm-efficiency-74d2ea9dae77

[6] https://www.nature.com/articles/d41586-024-02680-3

- THE END -

Thank you for your attention!

If you have any questions, feel free to ask.