MISTRAL 7B

architecture research paper decoded

SECTIONS

Here are the sections we are going to see about

Abstract

Introduction

Architectural Details

Result

Instruction Finetuning

Adding Guardrails

Conclusion

ABSTRACT

Mistral 7B -> a 7 billion parameter LM engineered for superior performance and efficiency that outperformas Llama2-13B on all benchmarks and most incase of Llama1-34B including code, reasoning, math, etc...

Mistral 7B uses:

- Grouped Query Attention
- Sliding Window Attention

for faster inferences and handling long sequences with reduced inference cost respectively

The model was finetuned on instructions known as Mistral 7B Instruct that surpasses Llama2 13B instruct and chat on various benchmarks.

The key highlight is that this model is under Apache2.0 license

INTRODUCTION

- **Challenge Existing:** With the race towards a higher model for performance the model size is being increased continuously which causes the barriers to deploy in practical real-world scenarios

- **Solution:** Mistral 7B is a model that has outperformed Llama2-13B on all tasks and Llama1-34B in most cases but also approaches the performance of Codellama without compromising the performance of the model in non-code benchmarks.

- **High Efficiency:** For high inference speed and reduced memory consumption, Mistral 7B uses GQA & SWA where the purposes are as follows -> (i) GQA -> Faster inferencing (ii) SWA -> Handle long sequences with reduced inference cost

- **Commercially available:** Another key highlight of this model is that this model is completely open-sourced and available for commercial usecases under the Apache2.0 license

ARCHITECTURAL DETAILS

KEY COMPONENTS

Sliding Window Attention
Handling long sequences with ease and reduced inference cost

Grouped Query Attention
For faster inferencing without loss of performance

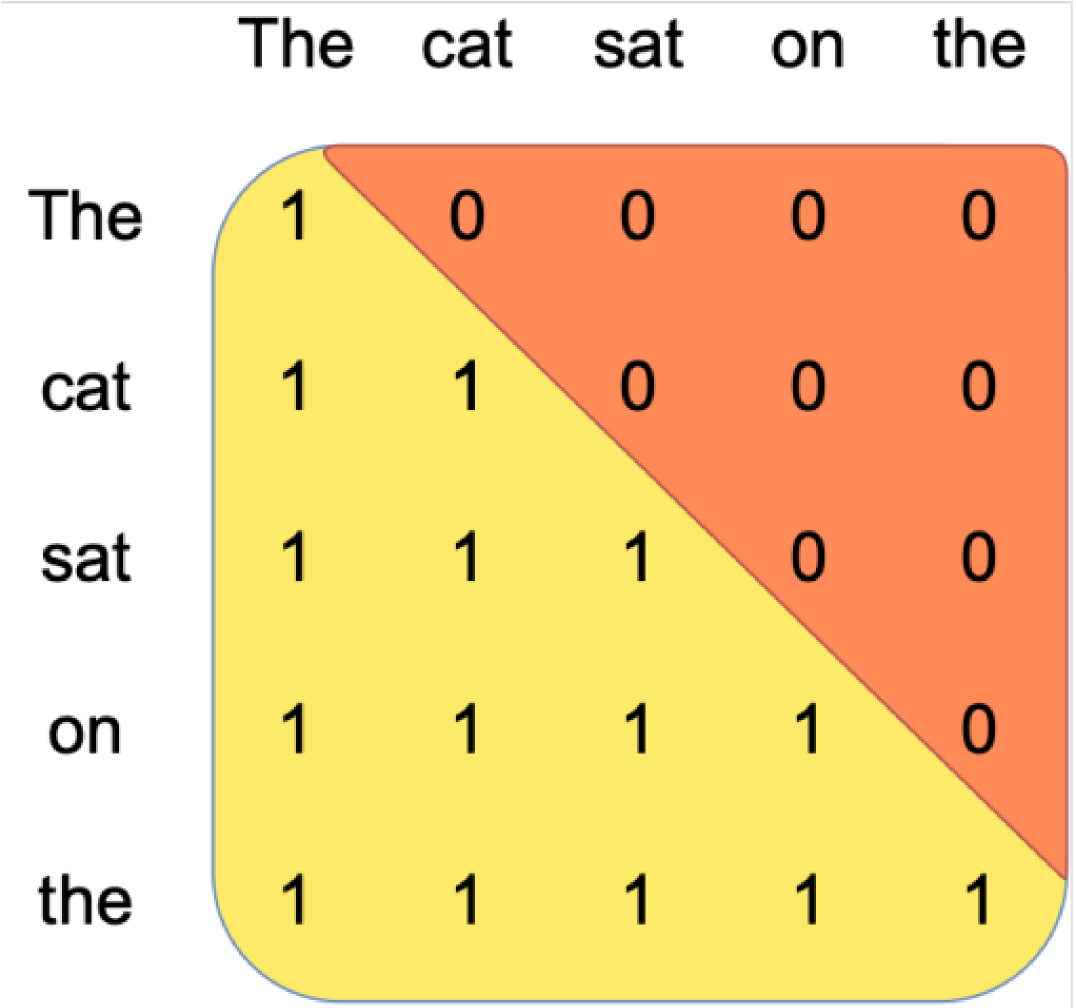
Rolling Buffer Cache
Efficient Cache memory usage without affecting model performance

Prefill & Chunking
Handling long prompt in the best way using prefill of cache

SLIDING WINDOW ATTENTION

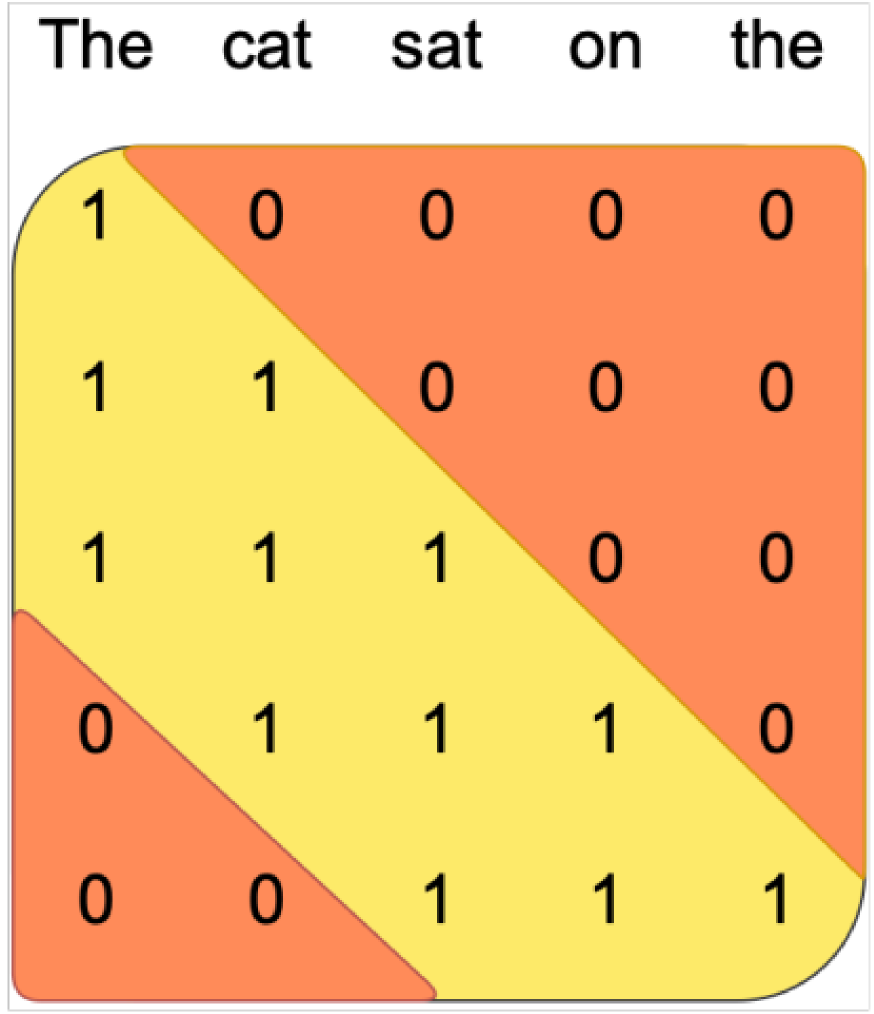
SWA -> Speedup inference and handle long sequences

- Exploits the stacked layers to attend information beyond window size W
- It attends hidden states [i-W, i]
- Tokens outside window will be unattended yet will influence the prediction
- At the last layer the effective span with W=4096 will be 131K tokens
- In practice, for 16k tokens has a speedup of 2x over vanilla attention



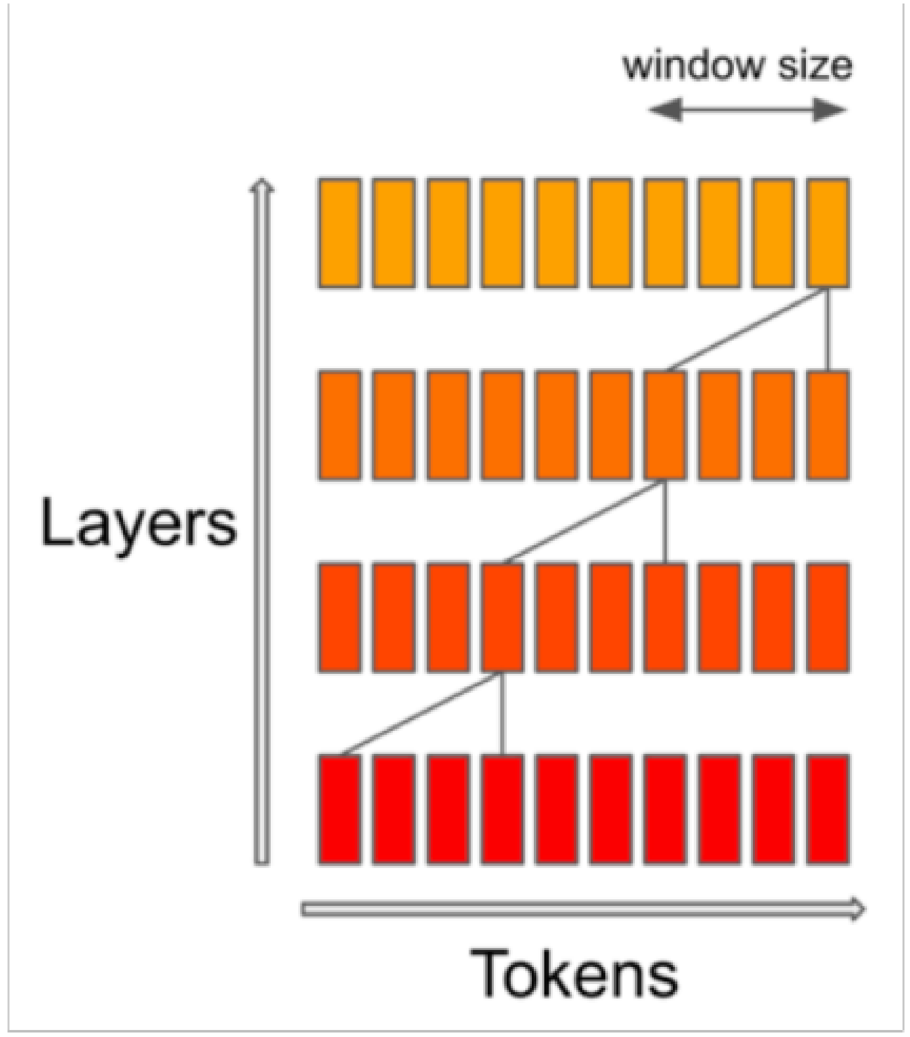
	The	cat	sat	on	the
The	1	0	0	0	0
cat	1	1	0	0	0
sat	1	1	1	0	0
on	1	1	1	1	0
the	1	1	1	1	1

Vanilla Attention



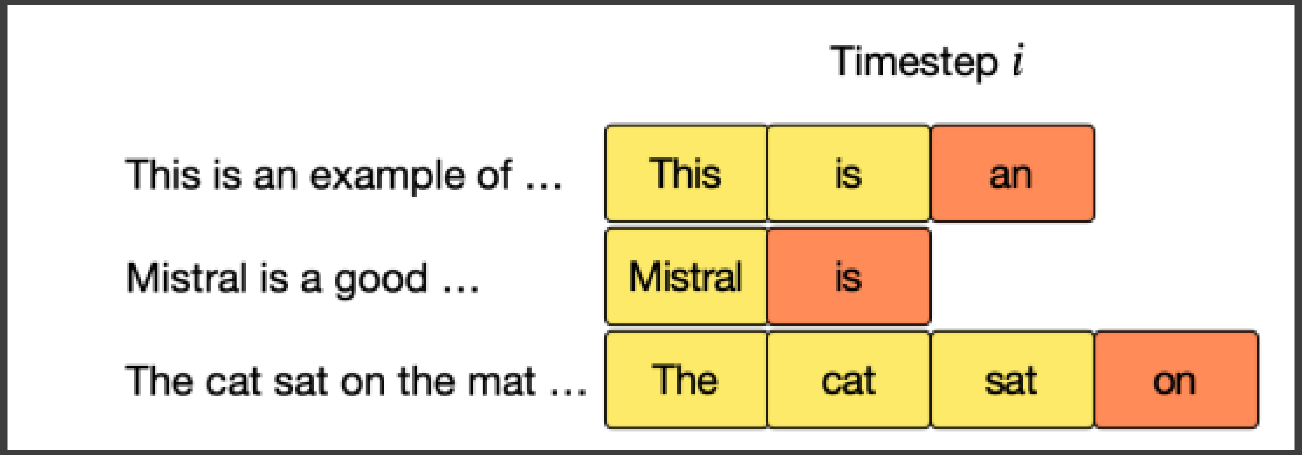
	The	cat	sat	on	the
	1	0	0	0	0
	1	1	0	0	0
	1	1	1	0	0
	0	1	1	1	0
	0	0	1	1	1

Sliding Window Attention



Effective Context Length

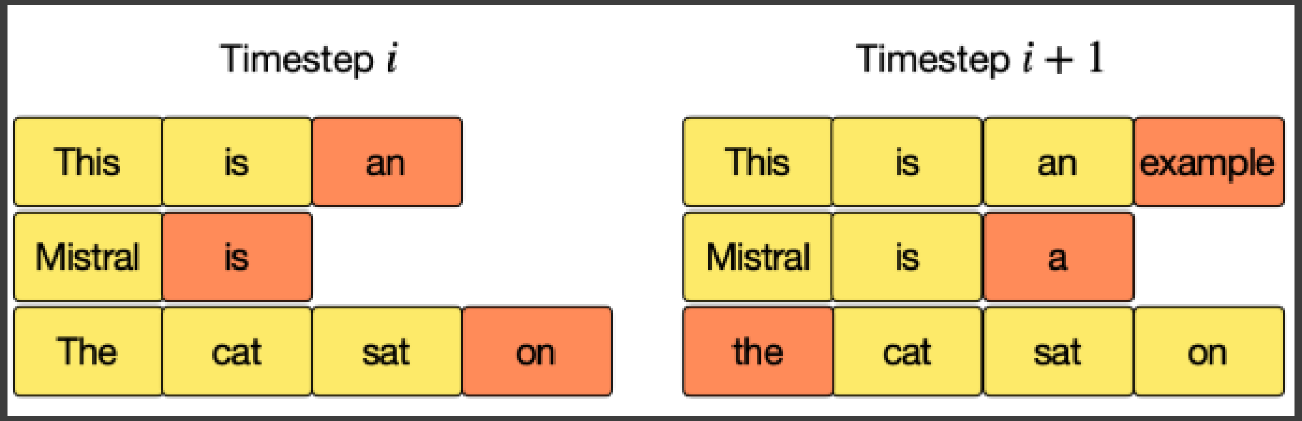
ROLLING BUFFER CACHE


Timestep i

This is an example of … | This | is | an

Mistral is a good … | Mistral | is

The cat sat on the mat … | The | cat | sat | on


Timestep i

This | is | an

Mistral | is

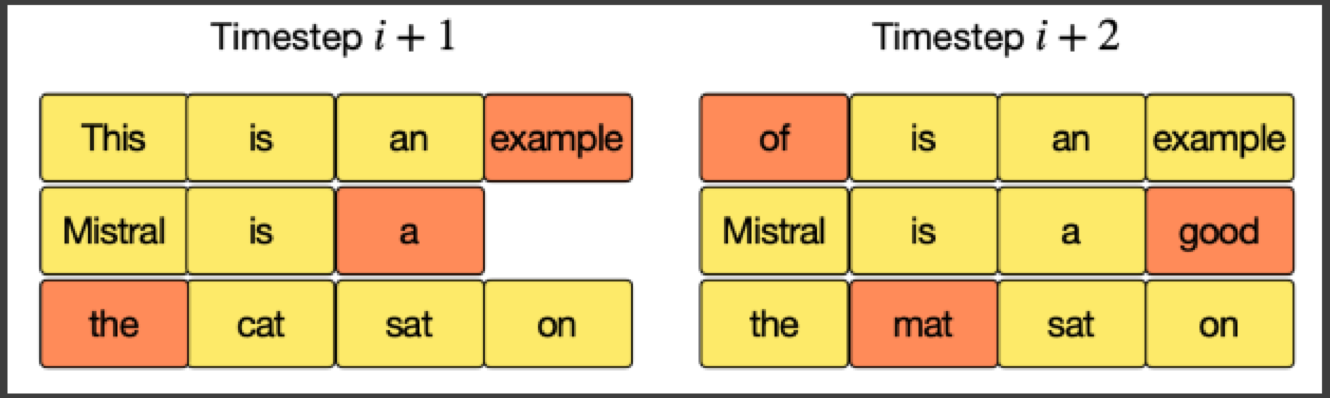
The | cat | sat | on

Timestep $i + 1$

This | is | an | example

Mistral | is | a

the | cat | sat | on


Timestep $i + 1$

This | is | an | example

Mistral | is | a

the | cat | sat | on

Timestep $i + 2$

of | is | an | example

Mistral | is | a | good

the | mat | sat | on

- Since we have a fixed attention span we can also have a limited cache size using a rolling buffer

- It has a fixed size of W. The keys and values of a timestep i are stored in position **i mod W**

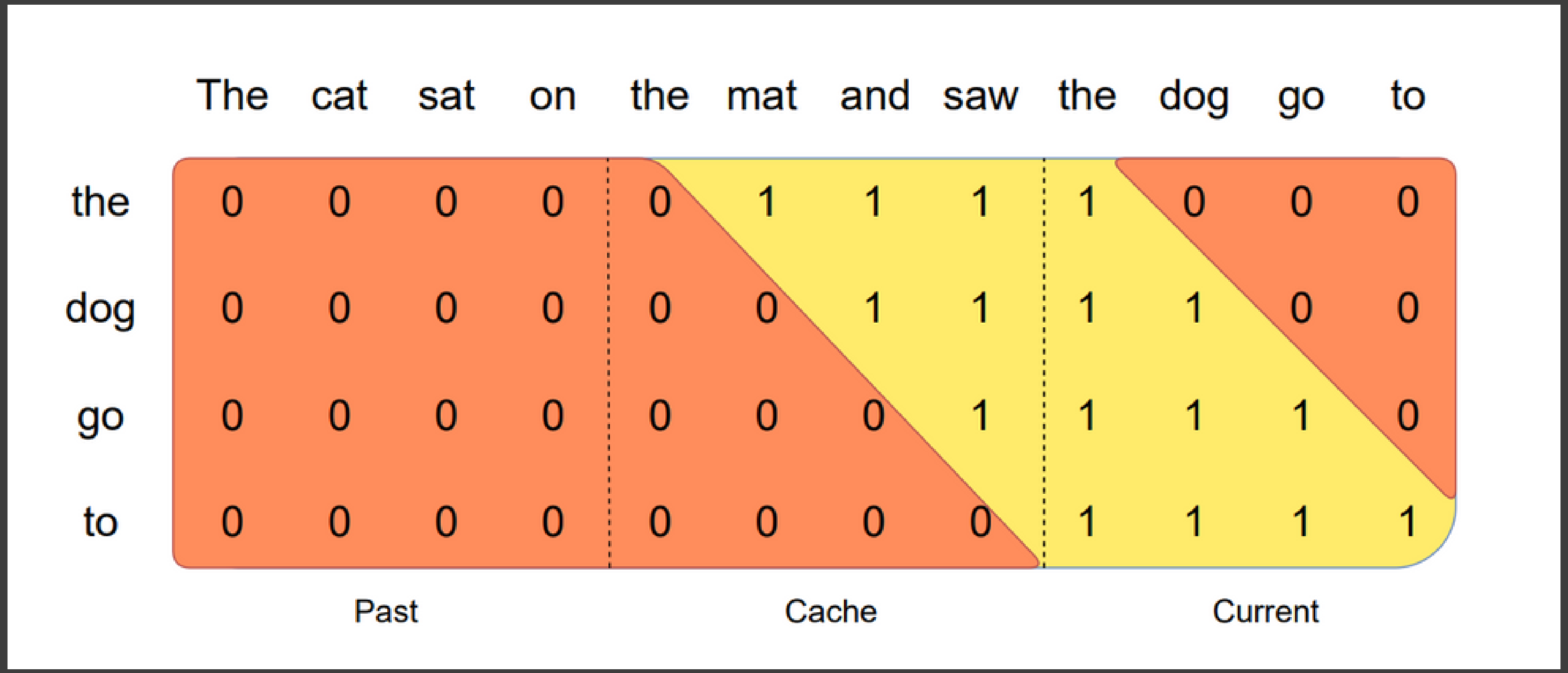
- When **i>W**, the cache of the past values and keys are overwritten

- This reduces the cache memory usage by 8x without impacting the model quality

PREFILL AND CHUNKING

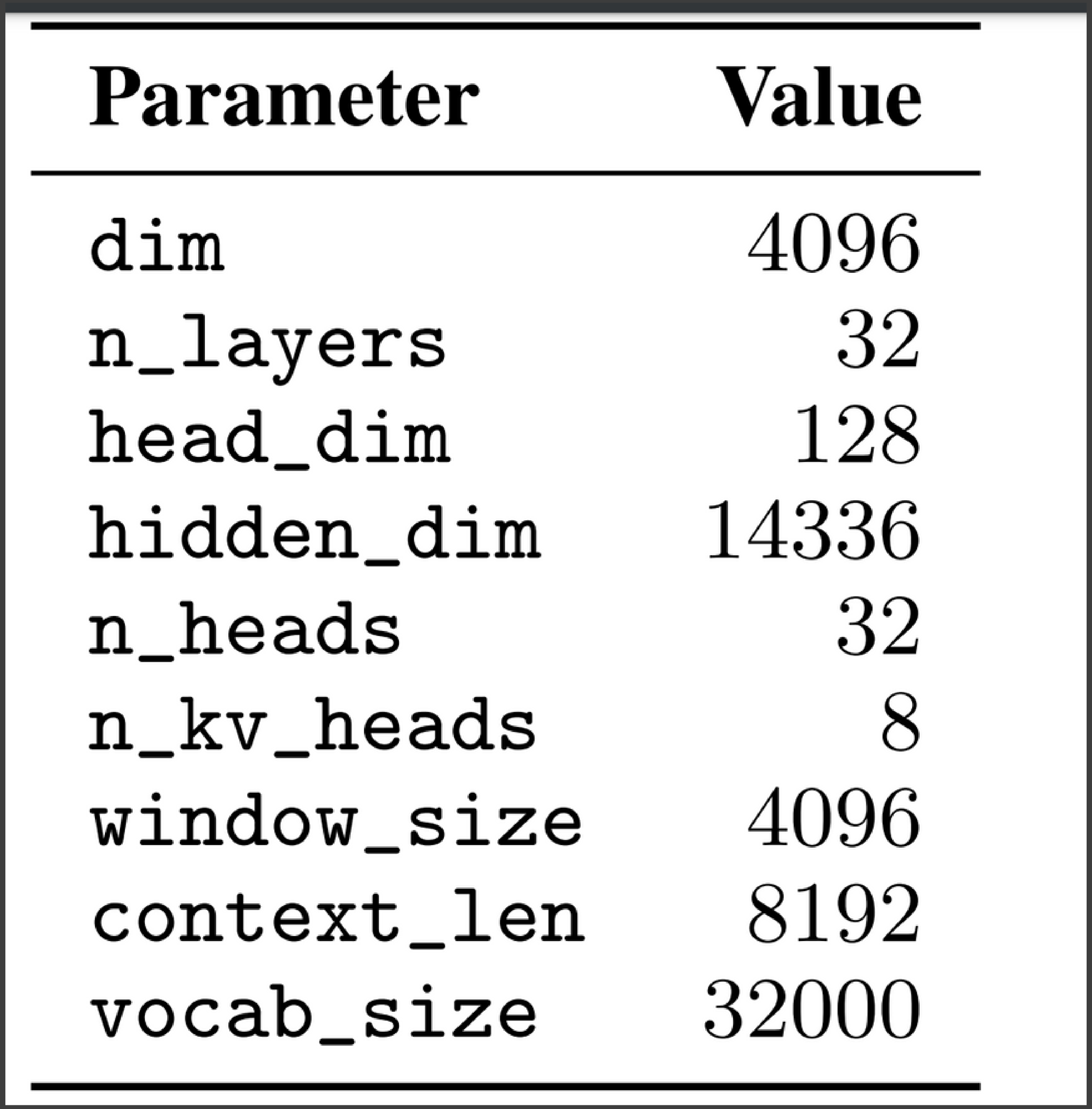
For generation, it needs to predict tokens one by one. But since the prompt is known prior we can prefill the cache. If the prompt is so large it can be chunked and the window size can be kept as chunk size.

In the figure,
Third chunk attends itself - Causal Mask
Center chunk attended - Sliding Window
First chunk attended - Past



	The	cat	sat	on	the	mat	and	saw	the	dog	go	to
the	0	0	0	0	0	1	1	1	1	0	0	0
dog	0	0	0	0	0	0	1	1	1	1	0	0
go	0	0	0	0	0	0	0	1	1	1	1	0
to	0	0	0	0	0	0	0	0	1	1	1	1

MODEL ARCHITECTURE



Parameter	Value
dim	4096
n_layers	32
head_dim	128
hidden_dim	14336
n_heads	32
n_kv_heads	8
window_size	4096
context_len	8192
vocab_size	32000

RESULTS

- Mistral 7B surpasses Llama2-13B across all benchmarks and in Llama1-34B on most particularly superior performance in code, mathematics, and reasoning benchmarks

- **Size and Efficiency:** When evaluated on reasoning, comprehension, and STEM, especially on MMLU, Mistral 7B matched the performance of the Llama2 model 3x its size. In knowledge benchmarks, it becomes 1.9x due to its limited parameter count that restricts the amount of knowledge it can store

RESULTS

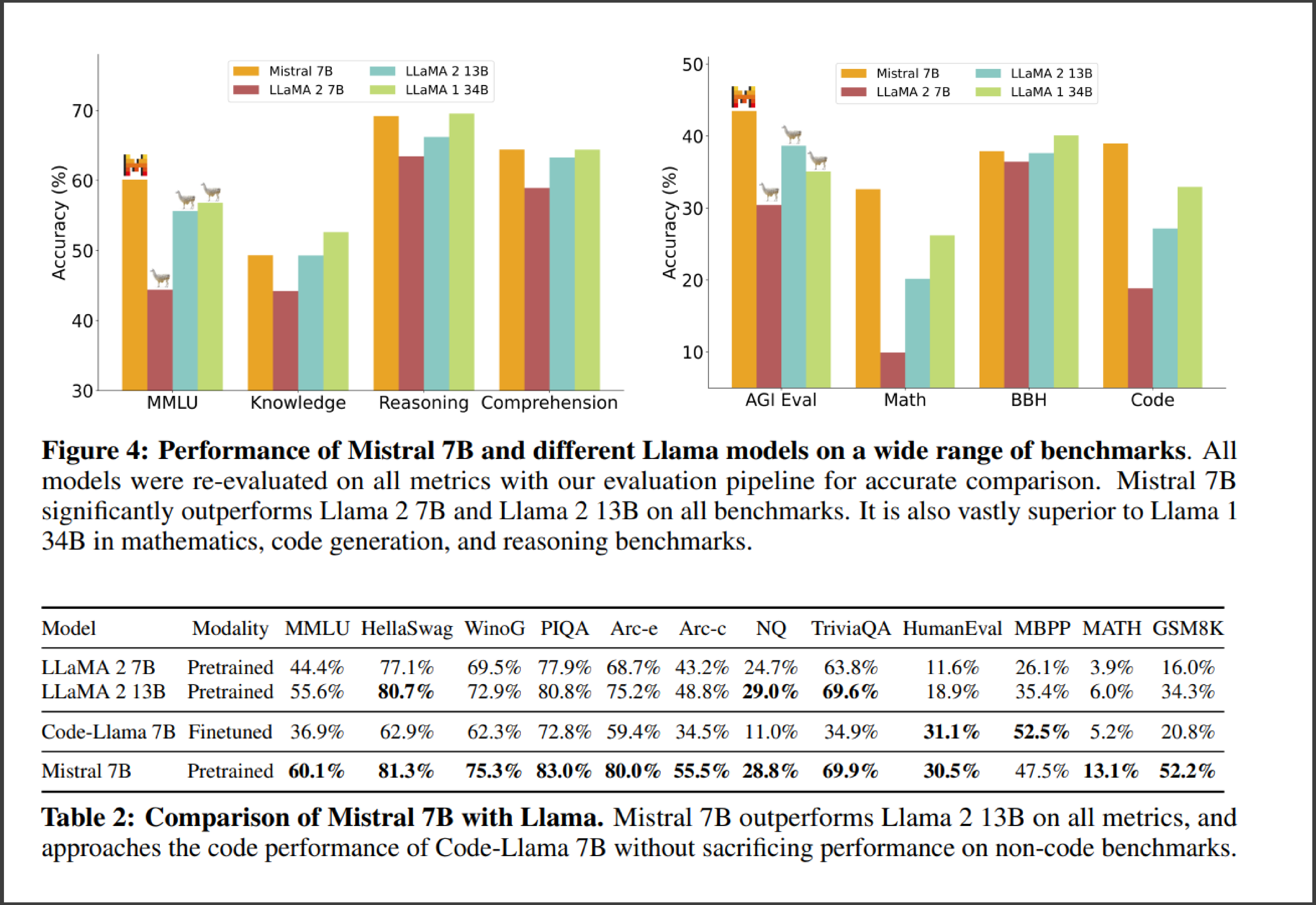


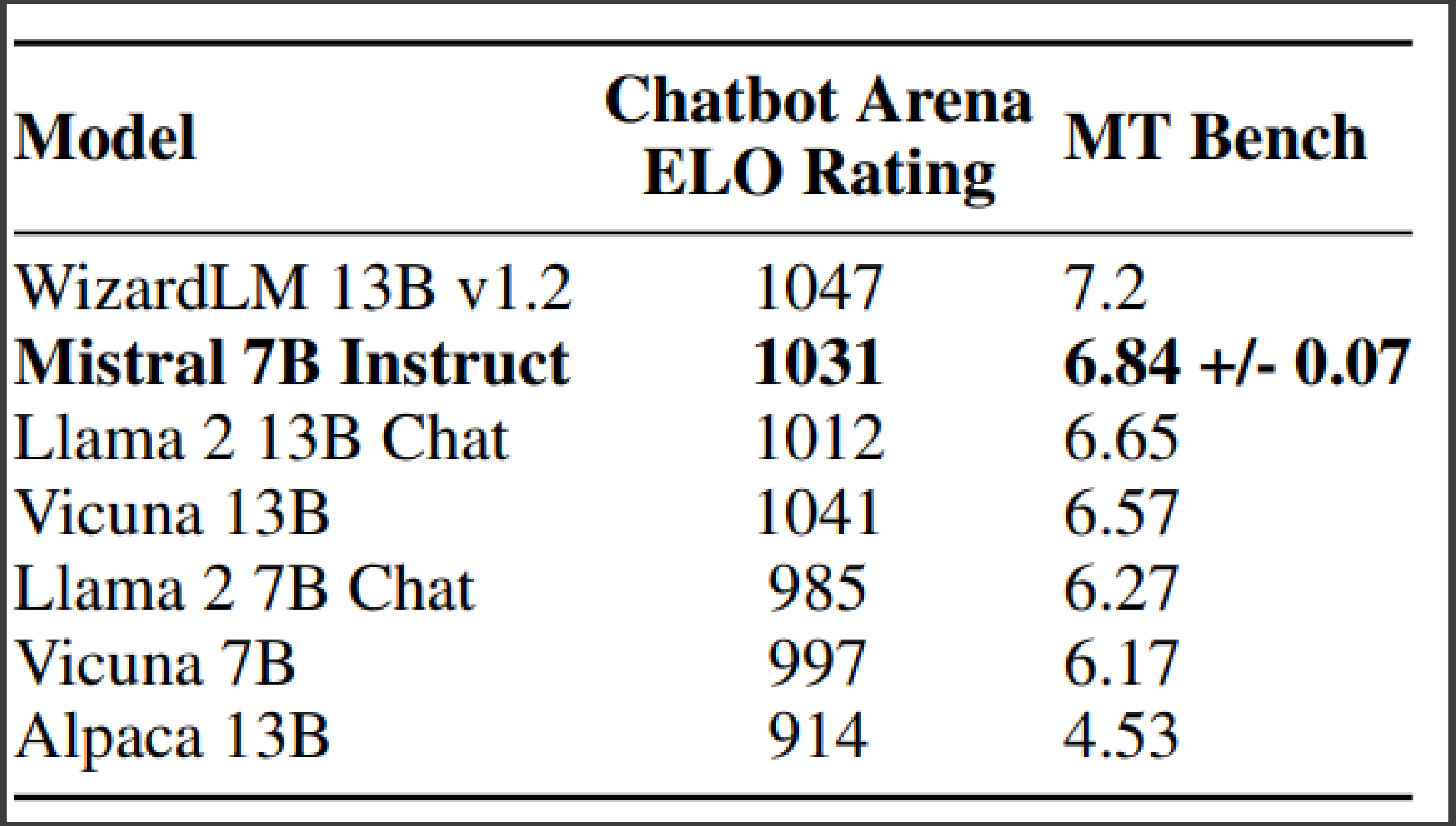
Figure 4: Performance of Mistral 7B and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mistral 7B significantly outperforms Llama 2 7B and Llama 2 13B on all benchmarks. It is also vastly superior to Llama 1 34B in mathematics, code generation, and reasoning benchmarks.

Model	Modality	MMLU	HellaSwag	WinoG	PIQA	Arc-e	Arc-c	NQ	TriviaQA	HumanEval	MBPP	MATH	GSM8K
LLaMA 2 7B	Pretrained	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	24.7%	63.8%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	Pretrained	55.6%	**80.7%**	72.9%	80.8%	75.2%	48.8%	**29.0%**	**69.6%**	18.9%	35.4%	6.0%	34.3%
Code-Llama 7B	Finetuned	36.9%	62.9%	62.3%	72.8%	59.4%	34.5%	11.0%	34.9%	**31.1%**	**52.5%**	5.2%	20.8%
Mistral 7B	Pretrained	**60.1%**	**81.3%**	**75.3%**	**83.0%**	**80.0%**	**55.5%**	**28.8%**	**69.9%**	**30.5%**	47.5%	**13.1%**	**52.2%**

Table 2: Comparison of Mistral 7B with Llama. Mistral 7B outperforms Llama 2 13B on all metrics, and approaches the code performance of Code-Llama 7B without sacrificing performance on non-code benchmarks.

INSTRUCTION FINETUNING

To evaluate the generalization capabilities it was finetuned on instruction datasets available on Hugging Face Hub & Mistral 7B Instruct model gave very good performance against the existing model in benchmarks



Model	Chatbot Arena ELO Rating	MT Bench
WizardLM 13B v1.2	1047	7.2
Mistral 7B Instruct	**1031**	**6.84 +/- 0.07**
Llama 2 13B Chat	1012	6.65
Vicuna 13B	1041	6.57
Llama 2 7B Chat	985	6.27
Vicuna 7B	997	6.17
Alpaca 13B	914	4.53

ADDING GUARDRAILS

> “
>
> The ability to enforce guardrails is very important for front facing end user applications

ADDING GUARDRAILS

System Prompt:
“Always assist with care, respect, and truth. Respond with utmost utility yet securely. Avoid harmful, unethical, prejudiced, or negative content. Ensure replies promote fairness and positivity.”

The above-specified system prompt ensures that the model declines to answer when the prompt is unsafe. When tested against 175 unsafe prompts the model declined to answer at every time.

But unlike Llama2 it doesn't decline to answer where it shouldn’t decline. For eg: When asked about how to kill a Linux process Mistral answered correctly while the Llama2 model declined to answer

Content moderation:
The model was able to act as a content moderator to classify a user prompt or generated answer as acceptable or not due to factors like hateful, discrimination, and so on and so forth