Rise of Efficient Language Model with Gemma 2 2B

Hrishikesh Yadav

ABOUT

Hrishikesh Yadav

Developer Advocate @TwelveLabs
AI Engineer @ShagaLabs
Kaggle 2x Expert
Applied Gen AI Researcher
Member @SuperTeamDao

About Myself

Deep Learning and Applied Generative AI Researcher

I like to participate and judge the hackathon

Worked on the product - RetroNexus and CrimeDekho

Published Research Work around the Predictive Policing and Time Forecasting

TABLE OF CONTENTS

TABLE OF CONTENTS

1

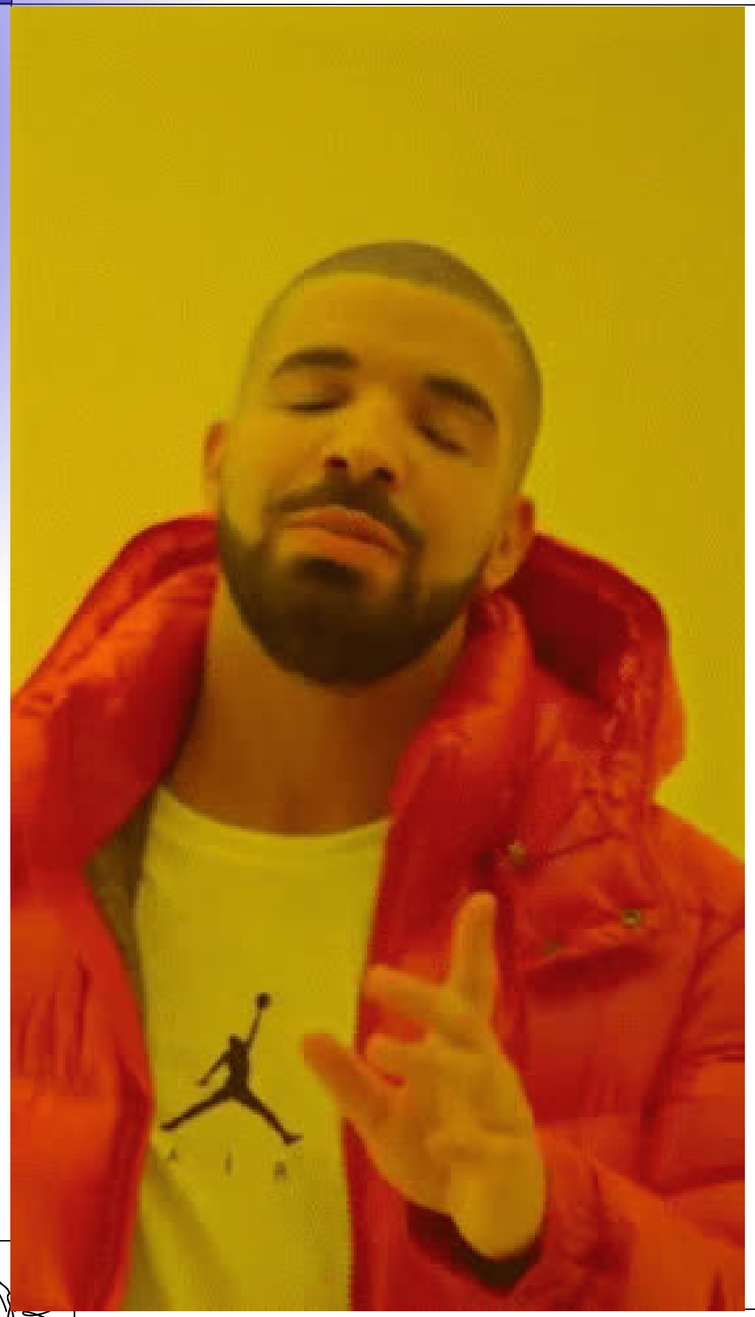
Understanding Problems with LLMs

Introduction to LLMs

- LLMs are generalists.
- Works amazing on the zero and few shot prompting performance on unseen tasks.
- Ability come at a cost - Difficult to deploy.

Problems with LLMs

- LLMs are very large - Deploying the 175 B model requires 350 GB+ of GPU memory.
  100 GB params ---> 200 GB (FP 16)
- Requirement of Specialized Infrastructure.
- LLM API endpoints are expensive.
- High CO2 Emission.



Solution to the Problem

- Don’t Deploy Large Generalist Models
- Deploy Small, Specialized Models

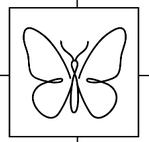
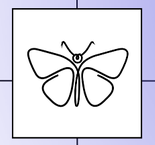
2

Open source LLMs vs Enterprise LLMs

Open source LLMs vs Enterprise LLMs

- Lower initial cost but the high maintainence.
- Collaborative and fast innovation from community.
- Highly customizable with access to the source code.
- Enhanced Data Security

- Higher initial cost but the lower maintainence.
- Slower innovation directed by a single entity.
- Customization possible within the platform’s limit.
- **Varies**

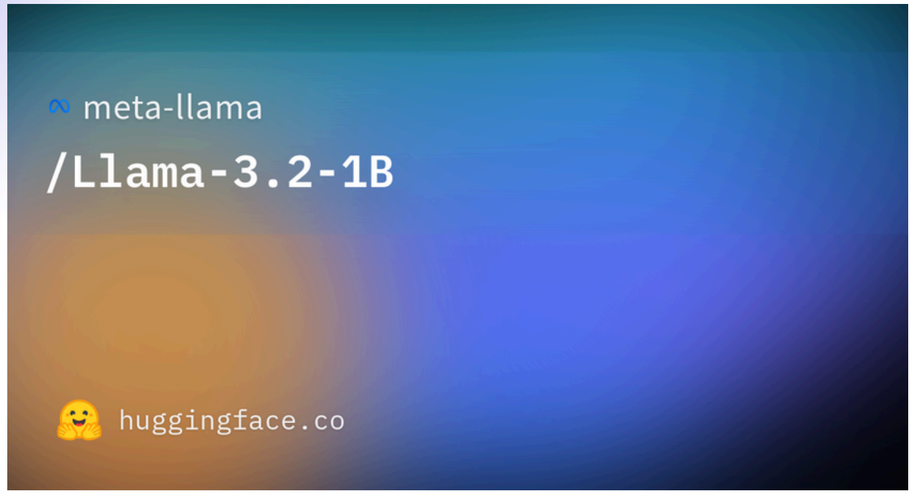



https://mistral.ai/news/ministraux/

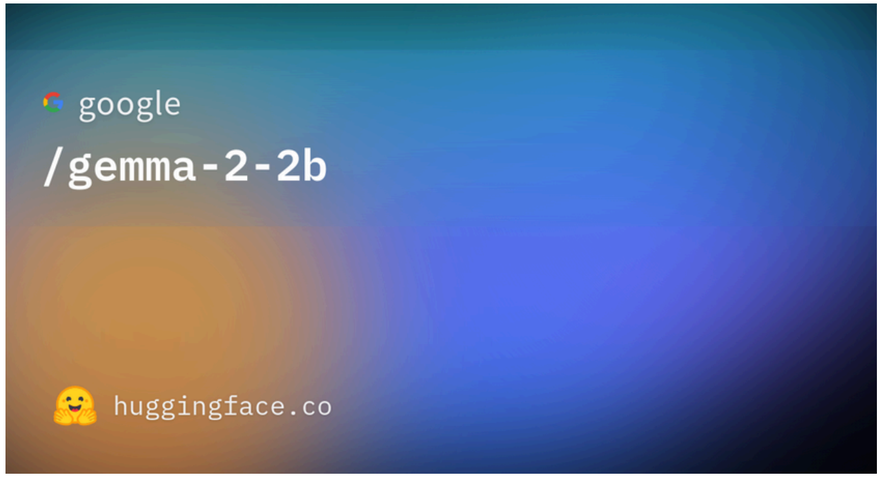
3

Rise of Open Source Compact Efficient SLMs

Rise of SLMs

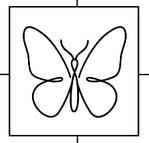

meta-llama
/Llama-3.2-1B
huggingface.co

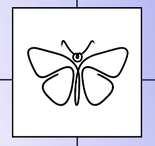

ministral
/Ministral-3b-instruct
huggingface.co


google
/gemma-2-2b
huggingface.co

Rise of SLMs

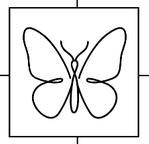
- SLMs are compact versions of larger language models, designed to perform specific tasks efficiently with fewer parameters.
- SLMs often use techniques like knowledge distillation or Quantization.
- Sacrifice some versatility and general knowledge for improved performance in specific domains or tasks.

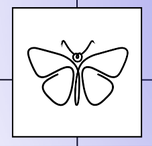




Problems which Efficient Open Source LLMs solve

- **Offline Settings** - On device or on premise where local inference may be required.
- **Handling Latency** - Situations were quick response are essential.
- Great for usecases with cost limitations, those having simpler task.
- Work well in **developer environments** were resources are limited.
- Specific **domain tasks** can be achieved better with the finetuning SLMs (Better than Out of box LLMs)





Problems which Efficient Open Source LLMs solve

Problems which Efficient Open Source LLMs solve

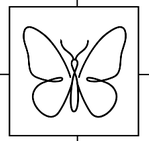
- SLMs on Edge devices serves the best privacy method for healthcare or any AI Assitive task for sensitive user data points.

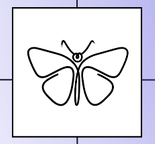
Research and Challenges

- Better Reasoning of pretrained Efficient Compact LLMs in Sub 10 B params.
- How to use this LLM models in production for Edge devices.
- Homomorphic encryption to increase the privacy behind the model.

Small Language Models aren’t perfect either

- Limited Contextual Understanding.
- Reduce Accuracy and Performance.
- Limited Creativity and Variability.
- Data Dependency.
- Scalibility Issue (Now, Solved with tool calling)



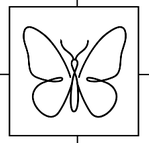


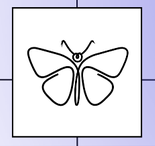
4

Sudden Rise of Usecases of SLMs

Sudden Rise of Usecases of SLMs

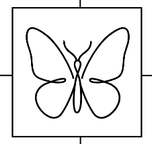
- Local on Premise, Offline working.
- Privacy first inference for on device task - Translation, Summarization.
- Internet Less Smart Assistants.
- Local Analytics.
- Autonomous Agents.

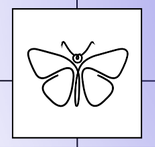




Sudden Rise of Usecases of SLMs - Applicative

- Input Parsing.
- Task Routing.
- Calling APIs based on the user intent across multiple context at low latency and low cost.





5

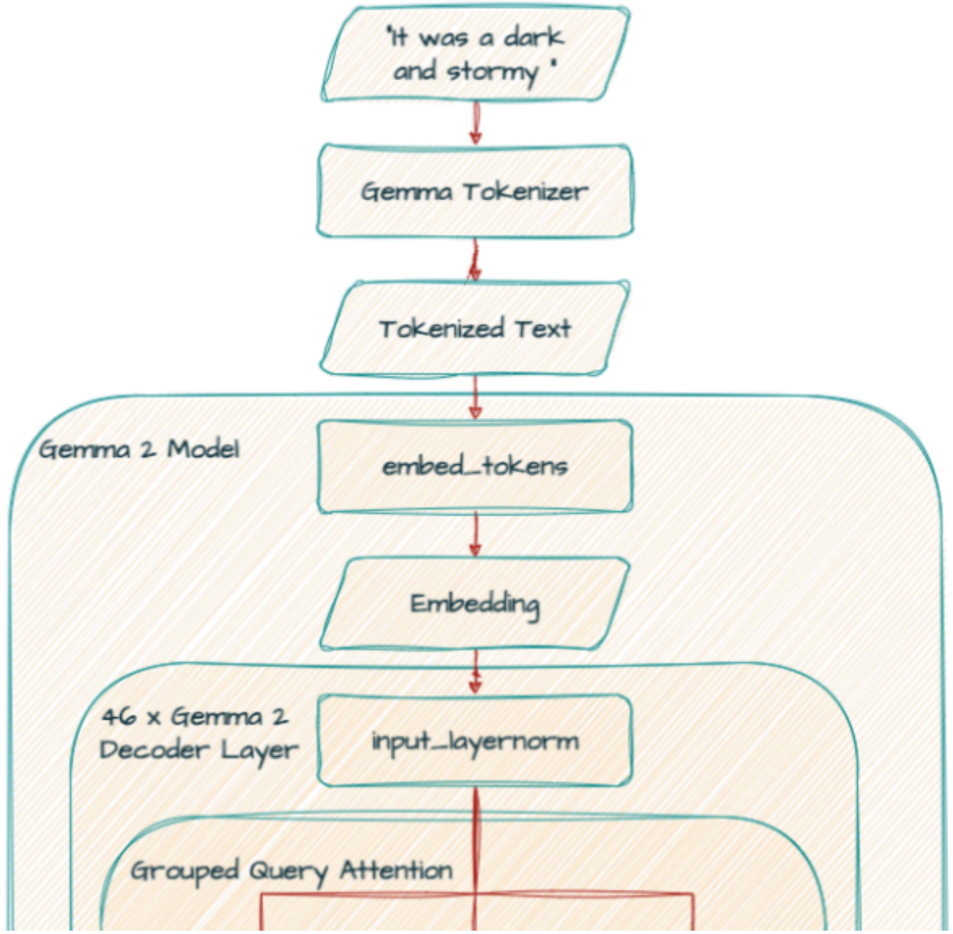
Understanding

Gemma 2 2B

Gemma 2 2B

- Gemma 2 2B was trained on 2T tokens using the Knowledge Distillation using a Large Gemma.
- Needs ~ 1 GB of memory in int4 Quantization
- Context window - 8K Tokens
- Run in browsers, on phones and on laptop.
- Strong performance for its size due to KD with 56.7 % on IFEVAL.

Gemma 2 27B Architecture

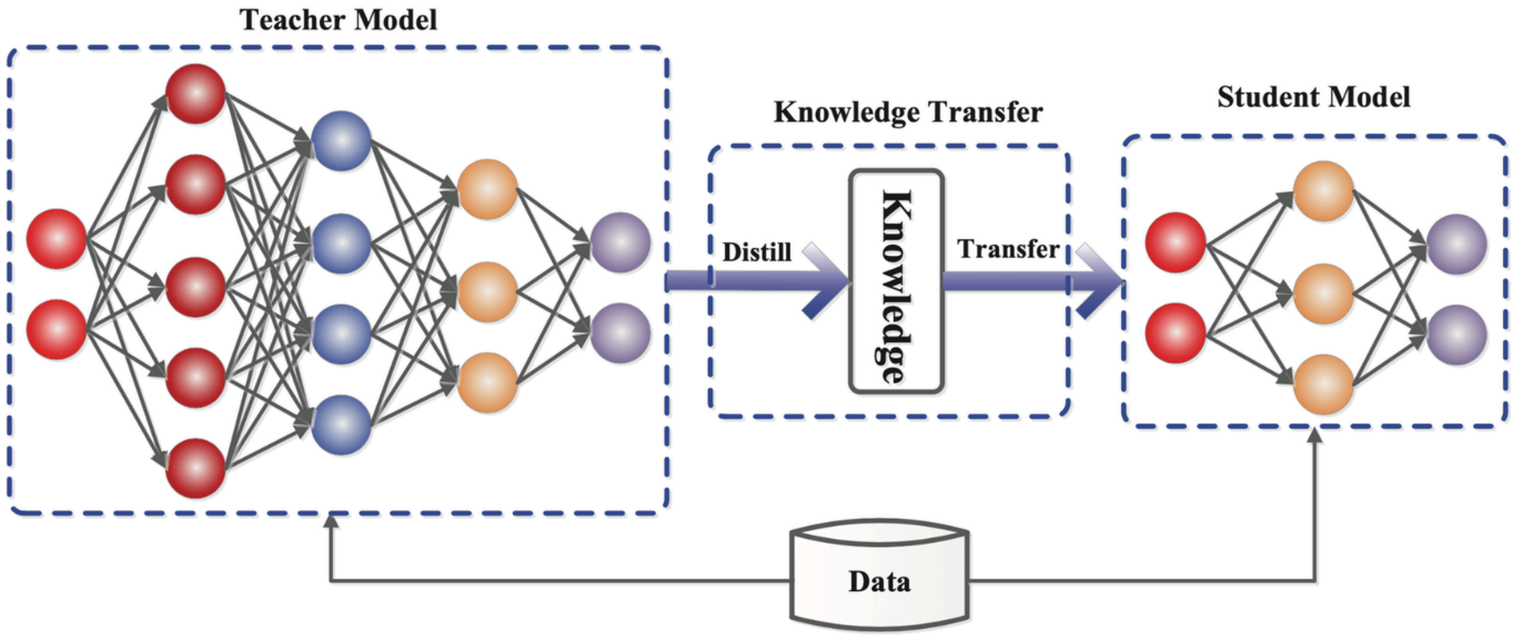


https://developers.googleblog.com/en/gemma-explained-new-in-gemma-2/

6

Specialized Model Building Technique for Deployment

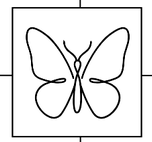
Model Knowledge Distillation

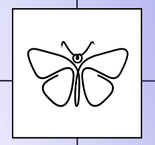


- Transfer capabilities of a larger, more powerful model to smaller model.
- Use teacher model to generate the pseudo labels for examples.
- Train student model on these pseudo labelled examples.

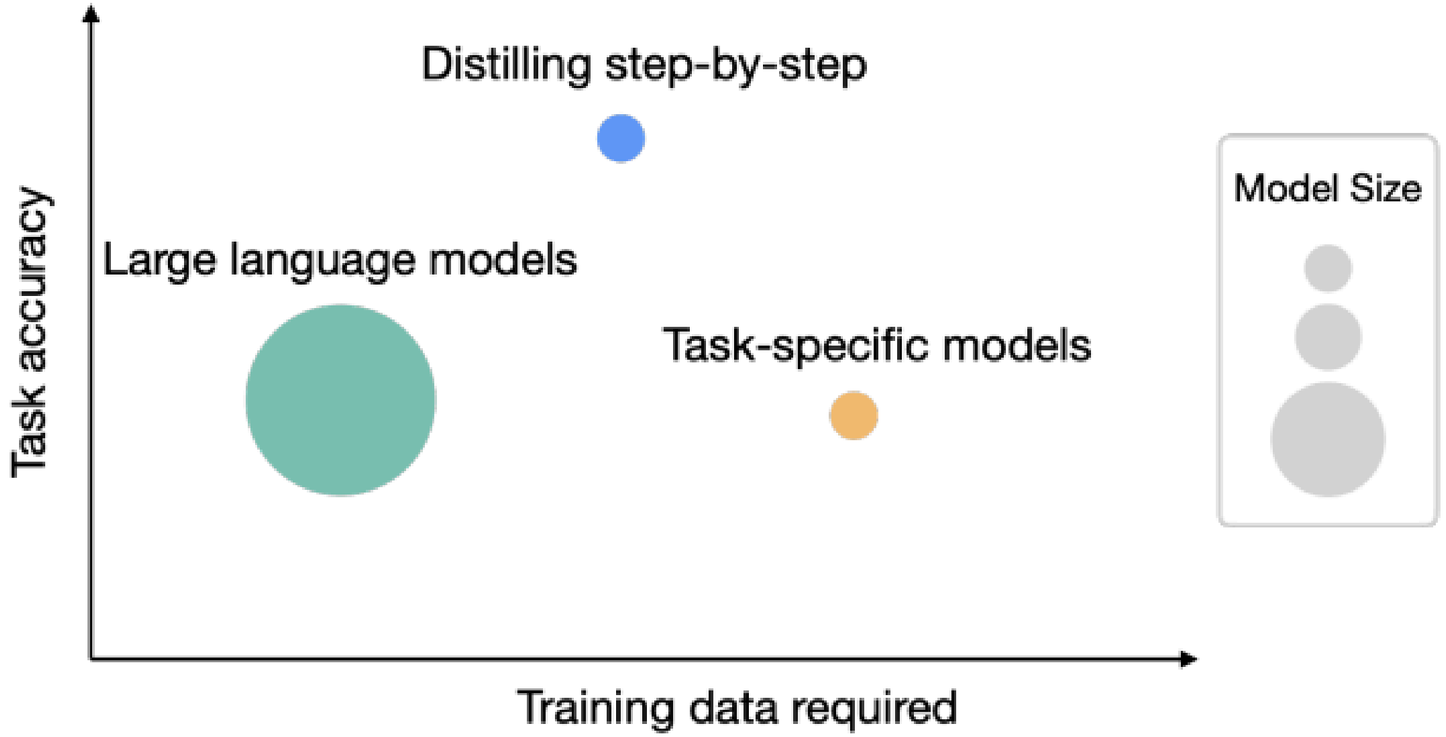
Model Knowledge Distillation

Requires Large Unlabelled datasets
(May not be available)

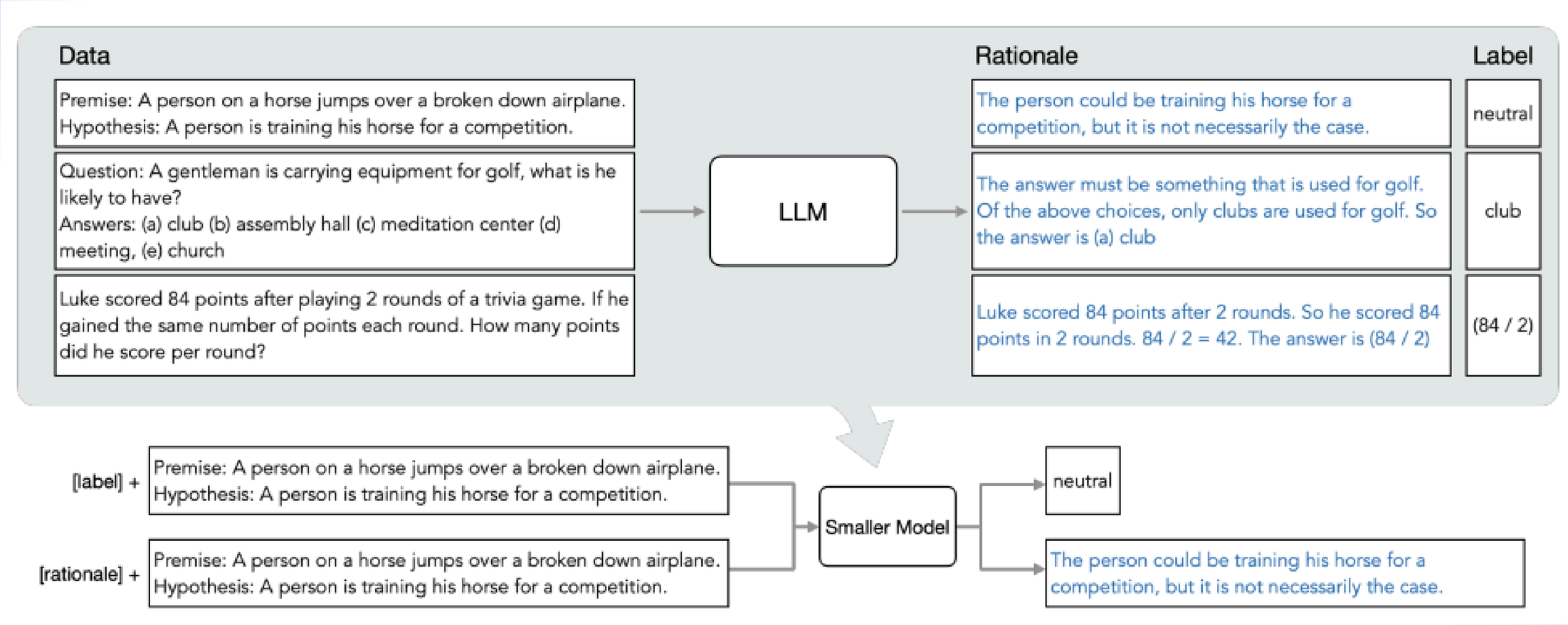




Model Knowledge Distillation

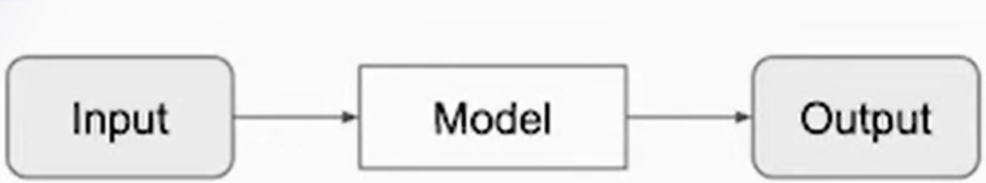


Model Knowledge Distillation



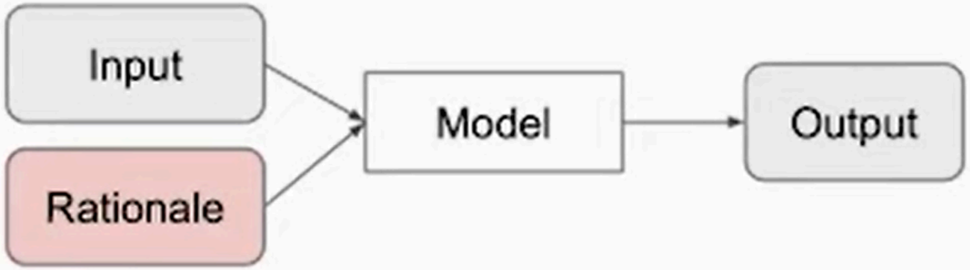
Mathematical Intution of KD step by step

Standard Loss for Finetuning on Pseudo Labels



$$\mathcal{L}_{\text{label}} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i), \hat{y}_i),$$

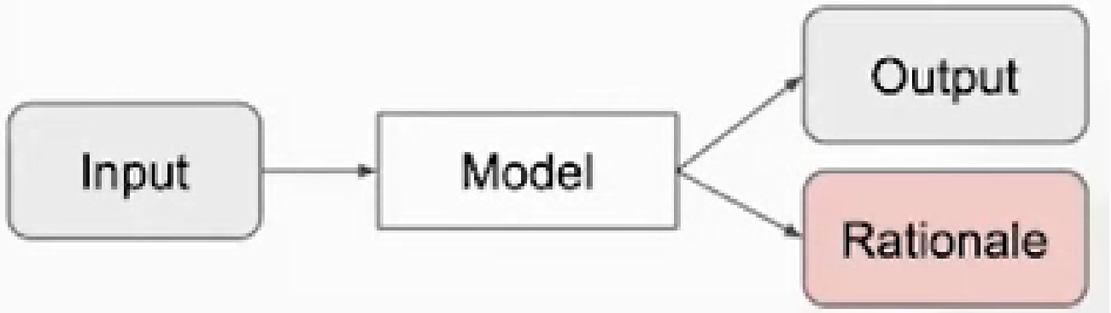
Train on Text + Rational



$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(x_i, \hat{r}_i), \hat{y}_i).$$

https://arxiv.org/pdf/2305.02301

Mathematical Intution of KD step by step



Rational as Supervision

$$\mathcal{L} = \mathcal{L}_{\text{label}} + \lambda \mathcal{L}_{\text{rationale}},$$

$$\mathcal{L}_{\text{rationale}} = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i), \hat{r}_i).$$

Removes dependency of rationals on the test time.

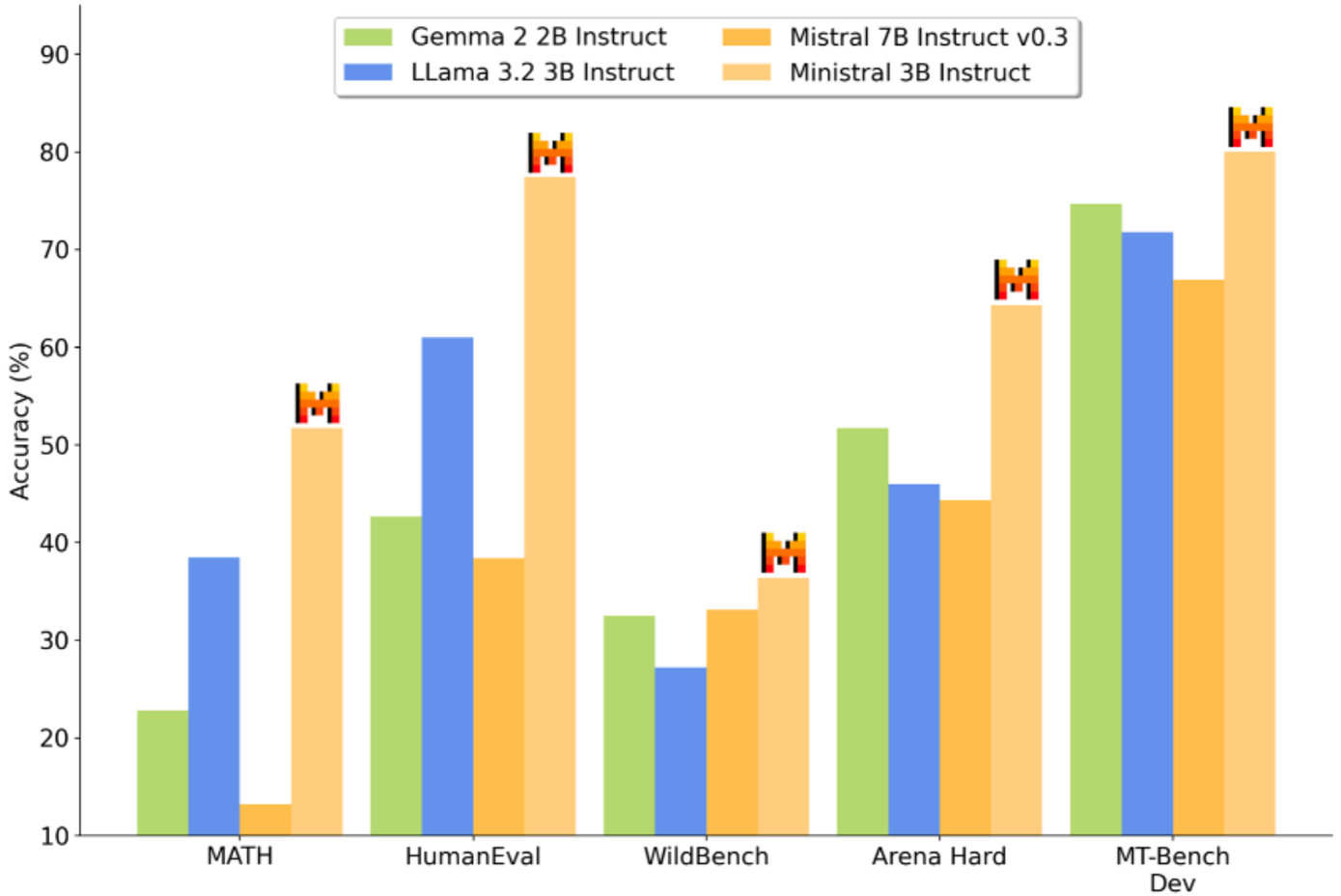
https://arxiv.org/pdf/2305.02301

7

Benchmark

Sub 10B model

3B Instruct model on Eval



https://mistral.ai/news/ministraux/

Pretrained Model Benchmark

Model	MMLU	AGIEval	Winogrande	Arc-c	TriviaQA	HumanEval pass@1	GSM8K maj@8	French MMLU	German MMLU	Spanish MMLU
Gemma 2 2B	52.4	33.8	68.7	42.6	47.8	20.1	35.5	41.0	40.1	41.7
Llama 3.2 3B	56.2	37.4	59.6	43.1	50.7	29.9	37.2	42.3	42.2	43.1
Ministral 3B	**60.9**	**42.1**	**72.7**	**64.2**	**56.7**	**34.2**	**50.9**	**49.1**	**48.3**	**49.5**
Mistral 7B	62.5	42.5	74.2	67.9	62.5	26.8	51.3	50.6	49.6	51.4
Llama 3.1 8B	64.7	44.4	74.6	46.0	60.2	**37.8**	61.7	50.8	52.8	54.6
Ministral 8B	**65.0**	**48.3**	**75.3**	**71.9**	**65.5**	34.8	**64.5**	**57.5**	**57.4**	**59.6**
	Knowledge & Commonsense					Code	Math	Multilingual		

https://mistral.ai/news/ministraux/

Gemma Model on MMLU

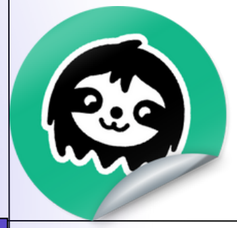
https://ai.google.dev/gemma

9 Accessing the Open Source Model, Right way!!!

Keras

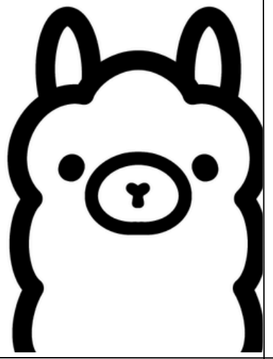
kaggle

LLaMA C++



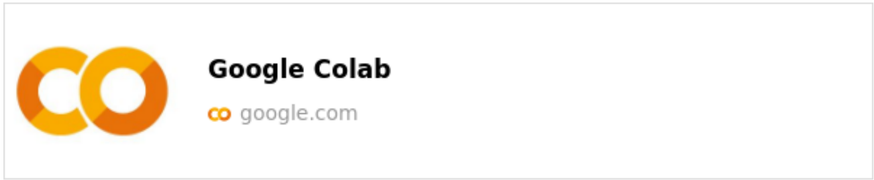
unsloth

LangChain



10

Hands-On Gemma 2 2B [Basic Prompting]



11

Fine Tuning
Gemma 2 2B

12 Kaggle Gemma Competition

GOOGLE · ANALYTICS COMPETITION · 3 MONTHS TO GO

Join Competition

Google - Unlock Global Communication with Gemma

Create Gemma model variants for a specific language or unique cultural aspect

Overview Data Code Models Discussion Rules

This competition requires identity verification
To submit to this competition, you'll need to verify your identity. Learn More

Verify now

Overview

This competition invites you to fine-tune Gemma 2 for a specific language or cultural context. By creating clear, easy-to-follow notebooks, you'll empower others to learn and contribute to the development of language models for diverse communities.

Start
15 days ago

Close
3 months to go

Competition Host
Google

Prizes & Awards
$150,000 Edit
Does not award Points or Medals

Participation
3,027 Entrants

Gemma 2 2B Tutorial Guide Github Repo

ABOUT

Twitter - @hrishikesh_ai

Developer Advocate @TwelveLabs
Prev. AI Engineer @Shaga
Kaggle 2x Expert
Applied Gen AI Researcher
Member @SuperTeamDao

Research Projects Listing to Collaborate

QnA

THANKS!

CREDITS: This presentation template was created by **Slidesgo**,
including icons by **Flaticon**, infographics & images by **Freepik**

Competition Time

1

```
X = np.array([1, 2, 3, 4, 5]).reshape(-1, 1)
y = np.array([1, 2, 3, 4, 5])

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(y_test)

mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error: {mse}")
```

2

```
y_pred = model.predict(X_test)
accuracy = model.accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

3

```
X = np.array([[1, 1], [2, 2], [3, 3], [4, 4], [5, 5]])
y = np.array([0, 0, 1, 1, 1])
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.5)
model = KNeighborsClassifier(n_neighbors=6)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```