

# Watsonx.ai proof of experience education

Foundation models  
in watsonx.ai and  
model testing metrics

Felix Lee  
Principal, Learning Content Development  
Data and AI  
[felix@ca.ibm.com](mailto:felix@ca.ibm.com)





# Seller guidance and legal disclaimer

IBM and Business Partner  
Internal Use Only

Slides in this presentation marked as **"IBM and Business Partner Internal Use Only"** are for IBM and Business Partner use and should not be shared with clients or anyone else outside of IBM or the Business Partners' company.

© IBM Corporation 2023.  
**All Rights Reserved.**

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.












References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, or other results.

All client examples described are presented as illustrations of how those clients have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by client.

# Content

- Foundation models in watsonx.ai
- Other foundation models in the IBM portfolio
- Use cases for different watsonx.ai models
- Choosing model(s) for PoX
- Metrics for evaluating models

# Foundation models in watsonx.ai

  <b>flan-ul2-20b</b>  flan-ul2 is an encoder decoder model based on the T5 architecture and instruction-tuned using the Fine-tuned Language Net.  Provider: Google                      Source: Hugging Face	 <b>starcoder-15.5b</b>  The StarCoder models are 15.5B parameter models that can generate code from natural language descriptions.  Provider: BigCode                      Source: Hugging Face	 <b>mt0-xxl-13b</b>  An instruction-tuned iteration on mT5.  Provider: BigScience                      Source: Hugging Face	 <b>gpt-neox-20b</b>  A 20 billion parameter autoregressive language model trained on the Pile.  Provider: EleutherAI                      Source: Hugging Face
 <b>flan-t5-xxl-11b</b>  flan-t5-xxl is an 11 billion parameter model based on the Flan-T5 family.  Provider: Google                      Source: Hugging Face	 <b>granite-13b-chat-v1</b>  The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for generative tasks.  Provider: IBM                      Source: IBM	 <b>granite-13b-instruct-v1</b>  The Granite model series is a family of IBM-trained, dense decoder-only models, which are particularly well-suited for generative tasks.  Provider: IBM                      Source: IBM	 <b>mpt-7b-instruct2</b>  MPT-7B is a decoder-style transformer pretrained from scratch on 1T tokens of English text and code. This model was trained by IBM.  Provider: Mosaic, tuned by IBM                      Source: Hugging Face
 <b>llama-2-13b-chat</b>  Llama-2-13b-chat is an auto-regressive language model that uses an optimized transformer architecture.  Provider: Meta                      Source: Hugging Face	 <b>llama-2-70b-chat</b>  Llama-2-70b-chat is an auto-regressive language model that uses an optimized transformer architecture.  Provider: Meta                      Source: Hugging Face	<div>IBM delivers foundation models primarily in watsonx.ai.</div> <div>This diagram shows the available models as of November 2023. IBM continues to add to the list of both open-source models from Hugging Face, as well as IBM’s proprietary models.</div>	

# Foundation models in watsonx Code Assistant, Watson Studio

Foundation models are also delivered in other services and IBM products

- **watsonx Code Assistant**
  - granite.code.ansible (Ansible-tuned model)
  - granite.20b.code.cobol (Cobol2Java-tuned model)
- **Watson Studio**
  - slate – fine-tuned for entity extraction, relationship detection, and sentiment analysis
- **PoX implications**
  - IBM infuses its software with generative AI; it is not limited to just to watsonx.ai
  - Clients benefit from generative AI from various IBM services and watsonx.ai;  
adding foundation models in other IBM products simply takes the benefits of generative AI further, helping clients meet their AI-for-business objectives

# Why the plethora of smaller models?

## Larger models:

- Can be fairly good at many tasks, but still not at everything
- Very expensive to run - most vendors charge by tokens and core hours
- Very difficult and expensive to tune by clients (if at all possible)
- More “creative” in hallucinating
- Tend to be verbose – not always appropriate or desirable

## Smaller models

- Domain-specific models are smaller – focus on specific tasks, and tend not to perform as well outside of the domain
- Much cheaper to run
- Easier to prompt tune or fine-tune
- Less creative
- Less verbose – which may or may not be desirable
- Easier to govern

No ONE MODEL rules them all

IBM offers [Open, Trusted, Targeted](#) models that [Empower](#) clients

# Open-source versus IBM models

## Open-source models

- Large collection of innovative models
- Large models
- Various vendors provide different models with many built for special use cases (image, video, extraction, code, etc.)
- Can be black boxes with various issues:
  - Quality of training data (may include copyright, licensed, HAP contents)
  - Issues with data privacy, security

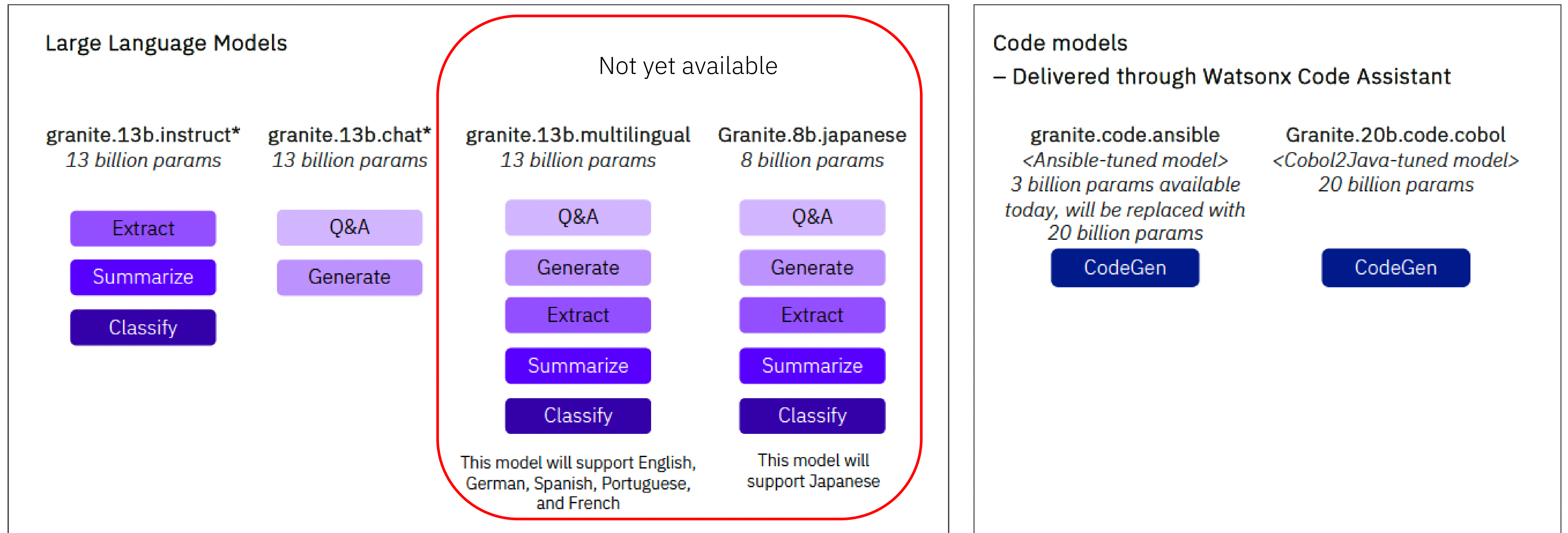
## IBM models

- Built on highly curated, filtered data, removing
  - Duplication
  - Copyright, licensed material
  - HAP content
- Users can use IBM models with confidence
- Easier to govern
- Built with IBM's enterprise data – ready to solve enterprise use cases
- Focus on specific use cases (Ansible, code translation, and more)
- Smaller in size (so far) – and is less costly to run



# IBM Granite model series

*Decoder-only, generative models*

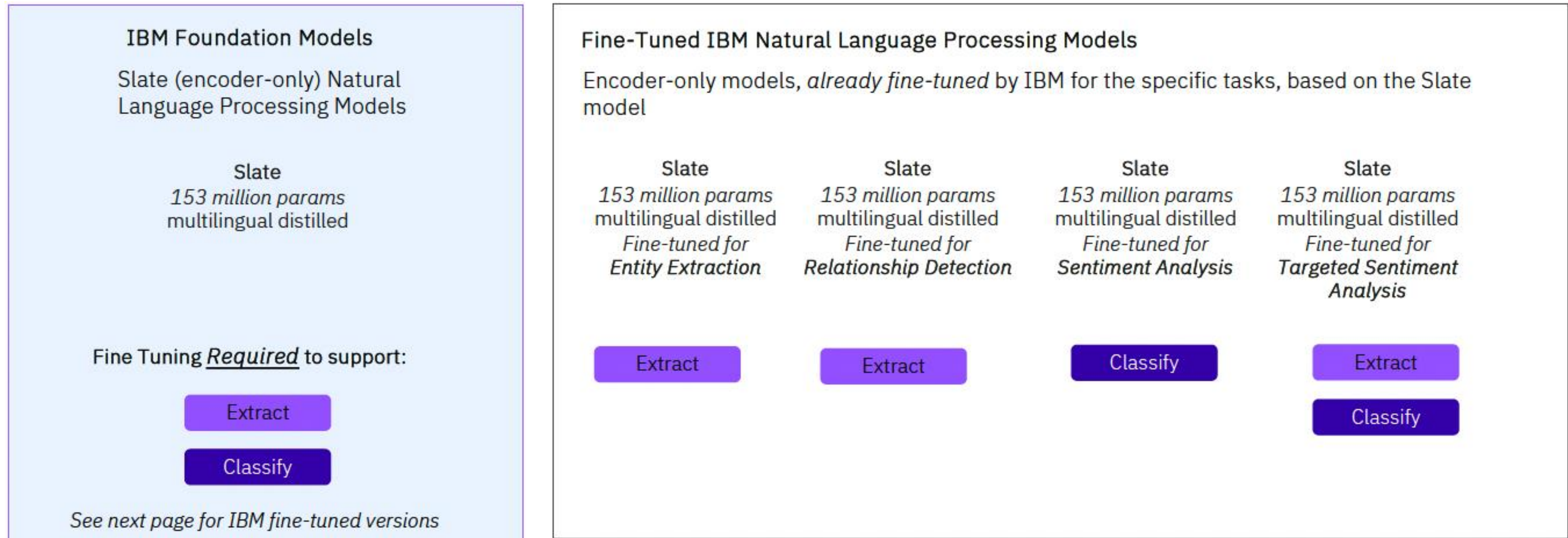


\* Granite models support all 5 tasks (Q&A, Generate, Extract, Summarize, and Classify).  
The .instruct model is designed to follow short instructions and return concise responses.  
The .chat model is designed for human/agent conversations and Q&A.



# IBM Slate model series


*Encoder-only, non-generative models*



- Slate model can be fine-tuned via notebooks and APIs
- Support batch inference via Notebook. No online inference. Batch inference is supported in both CPU and GPU Notebook environments (note the size of the model).

# Open-source and third-party models

*Encoder/decoder and decoder-only LLM (tuning not required for most tasks)*

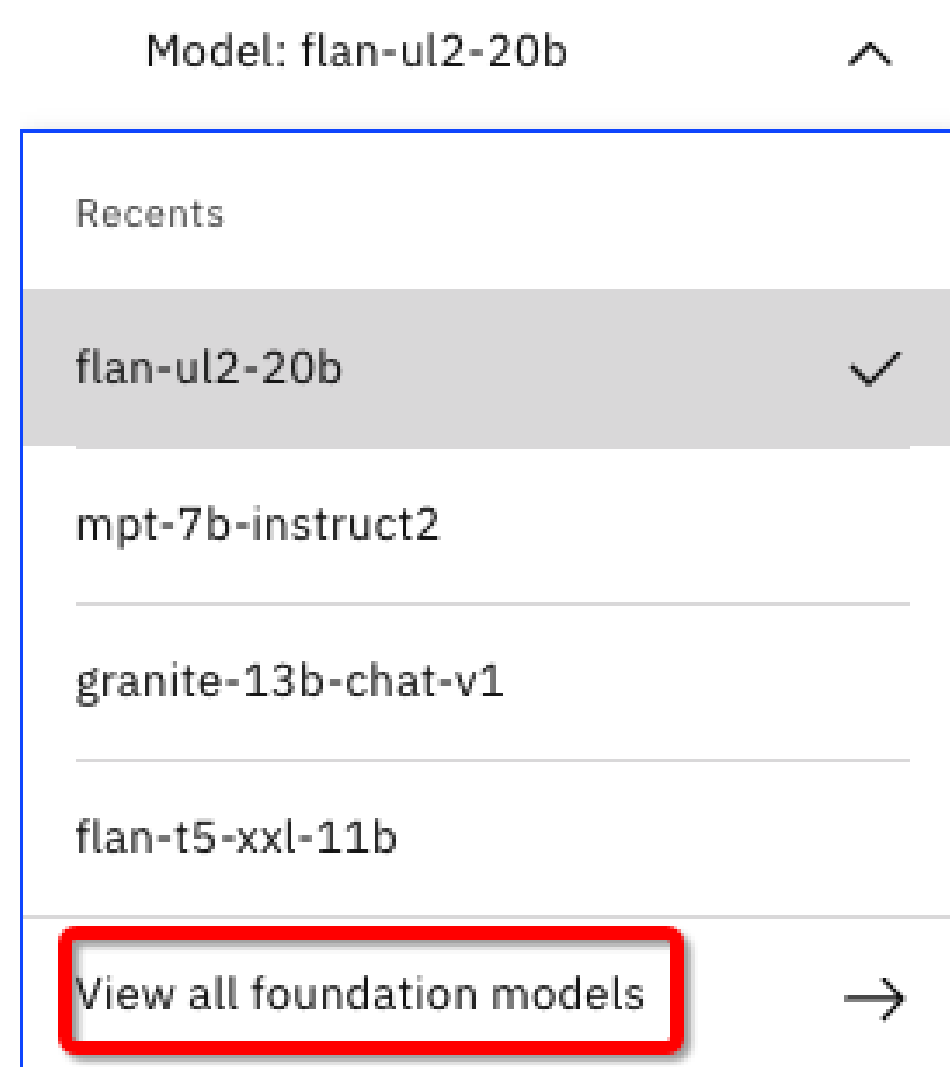
Open-Source Large Language Models					3 <sup>rd</sup> Party Large Language Models*	
						
flan-ul2-20b 20 billion params encoder/decoder	gpt-neox-20b 20 billion params decoder only	mt0-xxl-13b 13 billion params encoder/decoder	flan-t5-xxl-11b 11 billion params encoder/decoder	mpt-instruct2-7b 7 billion params decoder only	llama2-chat.70b 70 billion params decoder only	llama2-chat.13b 13 billion params decoder only
Q&A	Q&A	Q&A	Q&A	Q&A	Q&A	Q&A
Generate	Generate	Generate	Generate	Generate	Generate	Generate
Extract		Summarize	Summarize		Extract	Extract
Summarize		Classify	Classify		Summarize	Summarize
Classify					Classify	Classify
					Starcoder-15.5b 15.5 billion params decoder only <i>Task Specific Model</i>	
					CodeGen	

\*. Llama 2 and StarCoder have non-standard open-source terms with additional Acceptable Use Policies



# Choosing model(s) for PoX

- Select a model that is suitable for the use case
- On the watsonx.ai console, click the model pull down and click **View all foundation models**



- Click a model and read the details

Model information is not uniform. Look especially for **Intended use**. For example, gpt-neox-20b has this information:

## Intended use

GPT-NeoX-20B was developed primarily for research purposes. It learns an inner representation of the English language that can be used to extract features useful for downstream tasks. In addition to scientific uses, you may also further fine-tune and adapt GPT-NeoX-20B for deployment, as long as your use is in accordance with the Apache 2.0 license. This model works with the [Transformers Library](#). If you decide to use pre-trained GPT-NeoX-20B as a basis for your fine-tuned model, please note that you need to conduct your own risk and bias assessment.

This makes the model **not** a good PoX candidate.

# IBM Granite model information

Variant	Description / Intended Use	Pre-training Data Seen	MMLU (5-shot)
granite.13b.instruct	This variant is a Supervised Fine-Tuned (SFT) version of the base model to improve its instruction-following. It was tuned using a mix of FLAN and a mixture of other datasets (Dolly, HHRLHF, and IBM internal datasets, etc.). This model is intended as a starting point to help bootstrap further downstream alignment or task-specific tuning.	1000B Tokens	42.05
granite.13b.chat	This variant is a further-aligned version of the granite.13b.instruct variant. It was aligned using Contrastive Fine Tuning (CFT) for general to improve its harmlessness and the quality of its generation responses. This model should be used when looking to prompt-engineer out of the box, particularly when longer responses are desired. It also may be helpful as a starting point for further downstream fine-tuning.	1000B Tokens	42.07

## Intended Use

- Primary intended uses:
  - .chat / .instruct : The granite series of models are a family of IBM-trained decoder-only models used for text generation, summarization, question answering, classification, and extraction.
  - base : The base model will be primarily used to fine-tune downstream language tasks.
- Primary intended users:
  - The primary users are IBM enterprise clients looking to bolster their portfolios with enterprise-level generative AI models.
- Out-of-scope use cases:
  - The granite.13b models are not designed, tested, or supported, for code use cases of any kind.

The granite models are much better than the gpt-neox-20b model for PoX



# How to evaluate a foundation model?

## Leaderboard

[| Vote](#) | [| Blog](#) | [| GitHub](#) | [| Paper](#) | [| Dataset](#) | [| Twitter](#) | [| Discord](#) |

🏆 This leaderboard is based on the following three benchmarks.

- [Chatbot Arena](#) - a crowdsourced, randomized battle platform. We use 100K+ user votes to compute Elo ratings.
- [MT-Bench](#) - a set of challenging multi-turn questions. We use GPT-4 to grade the model responses.
- [MMLU](#) (5-shot) - a test to measure a model's multitask accuracy on 57 tasks.

📄 Code: The Arena Elo ratings are computed by this [notebook](#). The MT-bench scores (single-answer grading on a scale of 10) are computed by [fastchat.llm\\_judge](#). The MMLU scores are mostly computed by [InstructEval](#). Higher values are better for all benchmarks. Empty cells mean not available. Last updated: November, 2023.

Hugging Face puts out  
leaderboard reports on  
the most popular models.

Model <span>▲</span>	🌟 Arena Elo rating <span>▲</span>	📈 MT-bench (score) <span>▲</span>	MMLU <span>▲</span>	License <span>▲</span>
<a href="#">GPT-4</a>	1169	8.99	86.4	Proprietary
<a href="#">Claude-1</a>	1153	7.9	77	Proprietary
<a href="#">Claude-2</a>	1128	8.06	78.5	Proprietary
<a href="#">Claude-instant-1</a>	1109	7.85	73.4	Proprietary
<a href="#">GPT-3.5-turbo</a>	1109	7.94	70	Proprietary
<a href="#">WizardLM-70b-v1.0</a>	1096	7.71	63.7	Llama 2 Community
<a href="#">Vicuna-33B</a>	1095	7.12	59.2	Non-commercial
<a href="#">Llama-2-70b-chat</a>	1072	6.86	63	Llama 2 Community
<a href="#">OpenChat-3.5</a>	1066	7.81	64.3	Apache-2.0
<a href="#">WizardLM-13b-v1.2</a>	1051	7.2	52.7	Llama 2 Community
<a href="#">zephyr-7b-beta</a>	1044	7.34	61.4	MIT
<a href="#">MPT-30B-chat</a>	1038	6.39	50.4	CC-BY-NC-SA-4.0
...	...	...	...	...

General benchmarks  
are great, but they  
don't measure  
performance on  
your use cases,  
your prompting, &  
your fine-tuning!





# General process for model evaluation

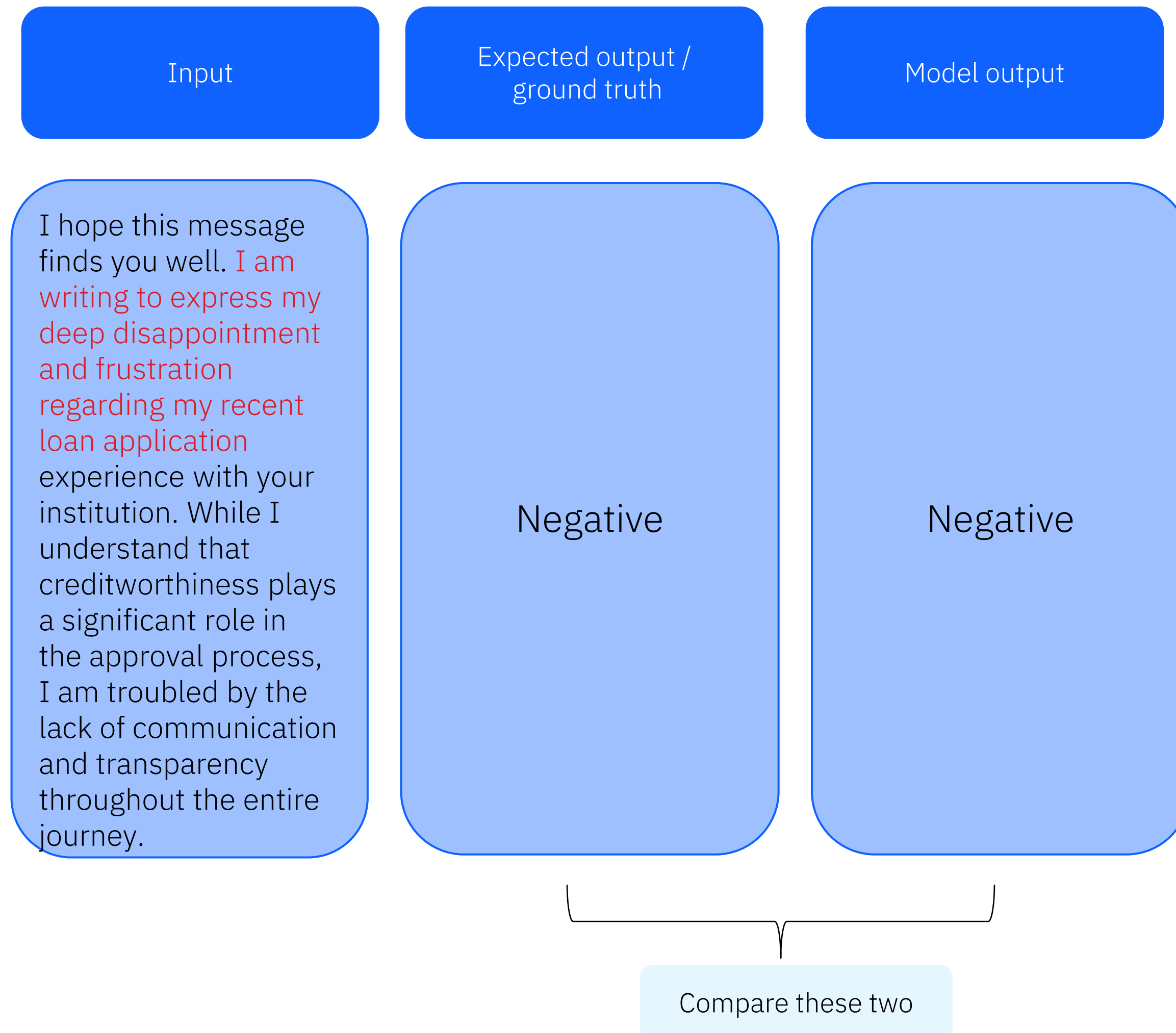
1. Collect a list of test records
2. Gather expected output (ground truth)
3. Generate model output
4. Compare model output with the expected output
5. Use a metric to measure the overall performance

} Freeform  
text



# Examples of metrics - accuracy

## *Text classification use case*



Read the customer review and classify the sentiment into **positive** or **negative**.

Model output = {"Negative", "Negative", "Negative"}

Expected output = {"Negative", "Negative", "Positive" }

=> Accuracy =  $2/3 = 66.67\%$

The accuracy metric is easy to understand and apply for outputs that are easy to compare as in classification.



# Other metrics commonly used

## ROUGE

- Recall-Oriented Understudy for Gisting Evaluation
- Evaluate **summarization**
- Compare the output summary against a human-produced summary
- More details on [ROUGE](#)

## BLEU

- Bilingual Evaluation Understudy
- Evaluate the **quality of translated** texts.
- Measure how close a model's output is to a set of good reference (human) translations.
- More details on [BLEU](#)

## Perplexity

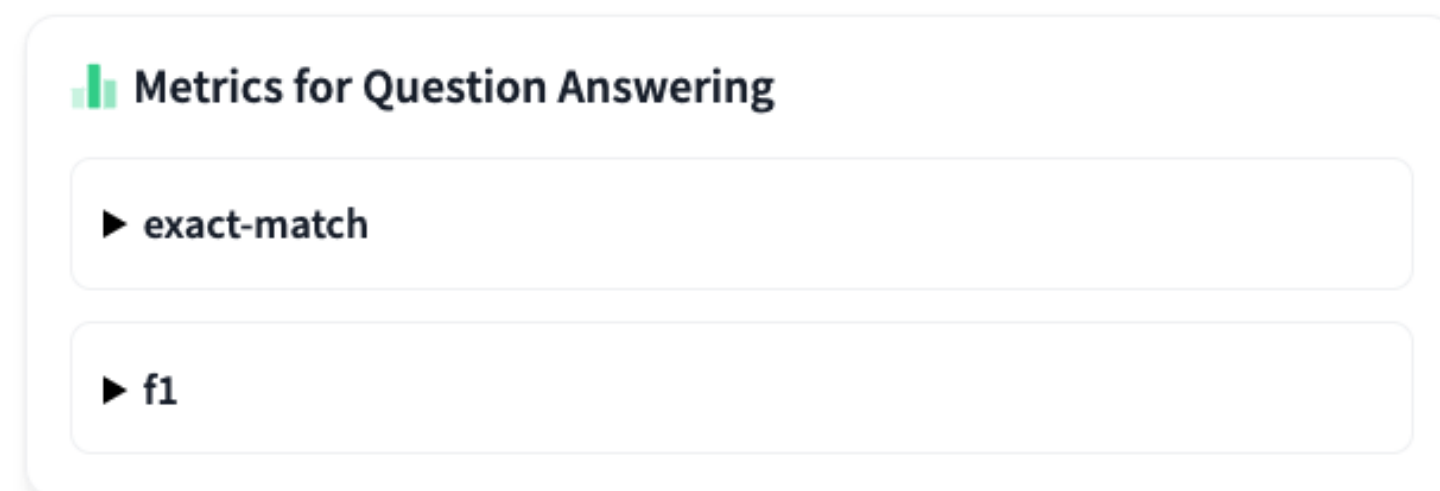
- Evaluate **text generation**
- Evaluate the probabilities assigned to the next word by the model.
- Lower perplexity indicates better performance
- More details on [Perplexity](#)

## F1

- Evaluate **question answering**
- The harmonic mean of the precision and recall value (both measure how frequently a model correctly identifies a true positive)
- More details on [F1](#)

# Selecting metrics in a PoX

- Important to have a way to measure the results of the PoX against client expectation
- Suggestion:
  1. Determine the type of use case in a PoX
  2. Go to <https://huggingface.co/tasks> and match to the closest use case (in the example to the right, Question Answering is highlighted).
  3. Click on the tile, and scroll down to see the set of metrics recommended for this task:



4. If you decide to use it, ensure you are familiar with how the metrics can be gathered.

