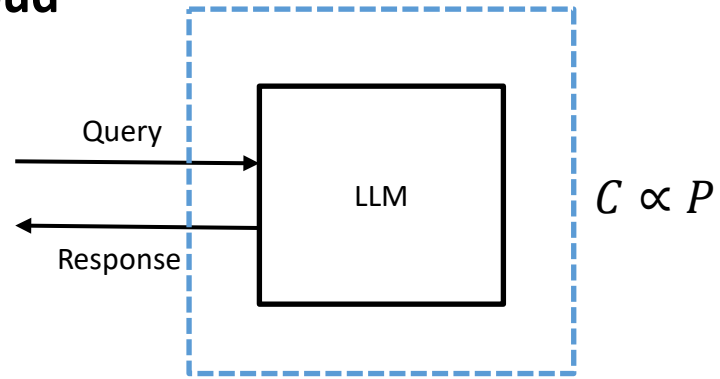
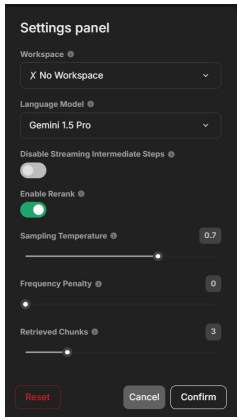
LLM Usage in AFSC/EN

- Current LLM services like ChatGPT and AskSage provide a chat-type interface for users to submit queries to the underlying LLM model
- This model usually needs to be chosen by the user and remains fixed during the chat-session
- In some cases, the model chosen by the user, may be overly complex and costly for the query submitted
- **The ability to dynamically move between LLMs and versions of any given LLM that best fits the need of the query is not currently present and should be an item to consider for AFSC/EN as it builds up its own LLM capabilities in the cloud**

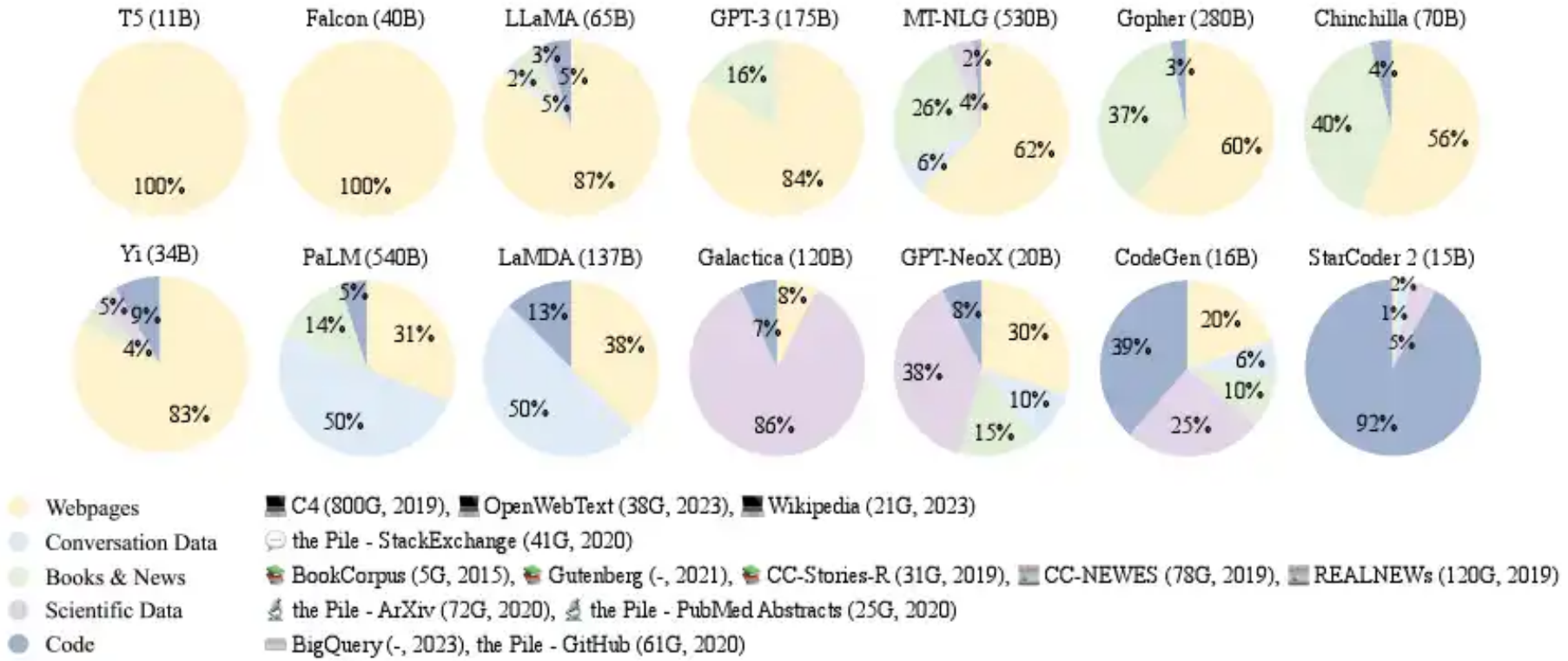


$C \propto P$



LLM Usage in AFSC/EN

- LLM models vary both in architecture and in underlying training-data, making some better suited to tackle specific types of queries, e.g. StarCoder2 trained mostly on code-data may be a more effective resource when utilizing LLMs in code-generation tasks (Zhao et al. 2023)



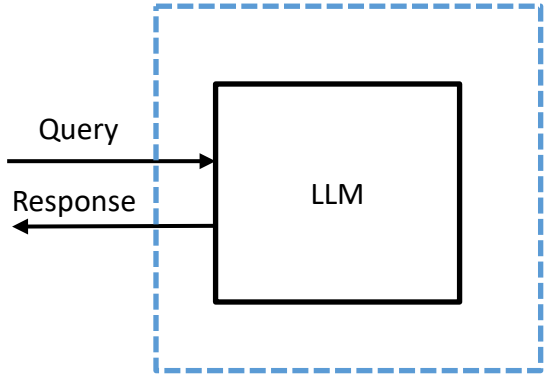
LLM Usage in AFSC/EN

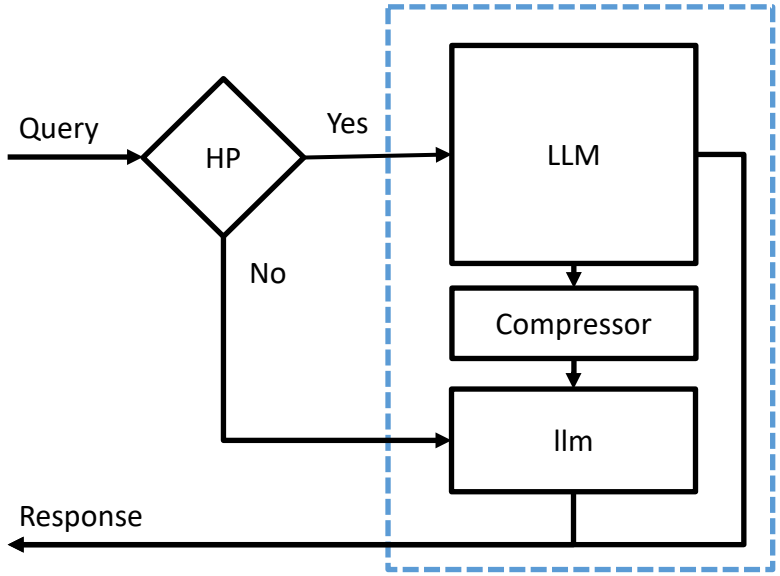
- Additionally, there are compression techniques that may reduce the size of the model without substantially affecting the model's performance. These techniques include:

Compression Technique	Description	Origins	Cons
Pruning	Removes weights in neural networks, reducing the computational costs.	LeCun et al 1989	Frameworks and hardware platforms may not support the resulting sparse models
Quantization	Lowers precision types, e.g. from 32-bit floats to 8-bit integers, reducing memory footprint	Courbariaux et al 2015	Training such models requires specialized knowledge and compression is lower than effective pruning
Distillation	Trains a 'student' model from the larger LLM	Hinton et al 2015	Student models may not generalize well beyond a specific domain

Proposed Scope of Study

- Purpose of the study is to review the feasibility of moving into a dynamic query-allocation between a full model and its compressed counterparts, depending on the nature of the query
- This can lead to opex savings down the line as we implement LLM capabilities in the cloud by reducing the average size of the LLM responding to queries





$C \propto P$

$C \propto P \cdot \varepsilon + p \cdot (1 - \varepsilon)$ $\log \frac{P}{p} \geq 1$

Previous Results

- Previous studies show relatively small drops in performance with an order of magnitude drop in model size (Shen et al. 2019)
  - E.g. SST-2 metric for accuracy in sentimental analysis results for quantization of BERT model from 415B parameters to <49B (Q-BERTMP 2/3 MP) with accuracy dropping to only **92.08%** from the full model's **93.00%**
- These studies also indicate that the compression method affects the resulting model's performance, i.e. DirectQ and Q-BERTMP 2/3 MP have an **8 percentage-points** difference:

(a) SST-2

Method	w-bits	e-bits	Acc	Size	Size-w/o-e
Baseline	32	32	93.00	415.4	324.5
Q-BERT	8	8	92.88	103.9	81.2
DirectQ	4	8	85.67	63.4	40.6
Q-BERT	4	8	**92.66**	63.4	40.6
DirectQ	3	8	82.86	53.2	30.5
Q-BERT	3	8	**92.54**	53.2	30.5
Q-BERTMP	2/4 MP	8	**92.55**	53.2	30.5
DirectQ	2	8	80.62	43.1	20.4
Q-BERT	2	8	**84.63**	43.1	20.4
Q-BERTMP	2/3 MP	8	**92.08**	**48.1**	**25.4**

Proposed Scope of Study

- One key use case in AFSC/EN is the code generation/debugging abilities of LLMs
- Metrics to assess an LLM's performance on this particular task are based on the Bilingual Translation Understudy, aka BLEU, which measures the quality of machine-generated translations with reference human-generated ones
  - Code-BLEU accounts for differences between natural language and code, mainly three: i) limited vocabulary, ii) structure (tree vs. sequential), and iii) unambiguous semantics (Ren et al. 2020)

$$\text{CodeBLEU} = \alpha \cdot \text{BLEU} + \beta \cdot \text{BLEU}_{\text{weight}} + \gamma \cdot \text{Match}_{\text{ast}} + \delta \cdot \text{Match}_{\text{df}}$$

Proposed Scope of Study

- Initial use case, with suggested model and compression technique studies:

Use-Case	Model	Model Size (B)	Compression Method	Resulting Model Size (B)	Metric
Code-generation	LLaMA 3.3	70	Pruning	*x*	$CodeBLEU$ (or other relevant metric)
			Quantization	*y*	
			Knowledge Distillation	*z*	

References

- Shen, S., et al. “Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT.” arXiv preprint arXiv: 1909.05840, 2019.
- Shuo, R. et al. “CodeBLEU: a Method for Automatic Evaluation of Code Synthesis.” arXiv preprint arXiv: 2009.10297, 2020
- Wang, J. et al. “Understanding User Experience in Large Language Model Interactions.” arXiv preprint arXiv: 2401.08329, 2025.
- Zhao, W., et al. “A Survey of Large Language Models.” arXiv preprint arXiv:2303.18223v15, 2024