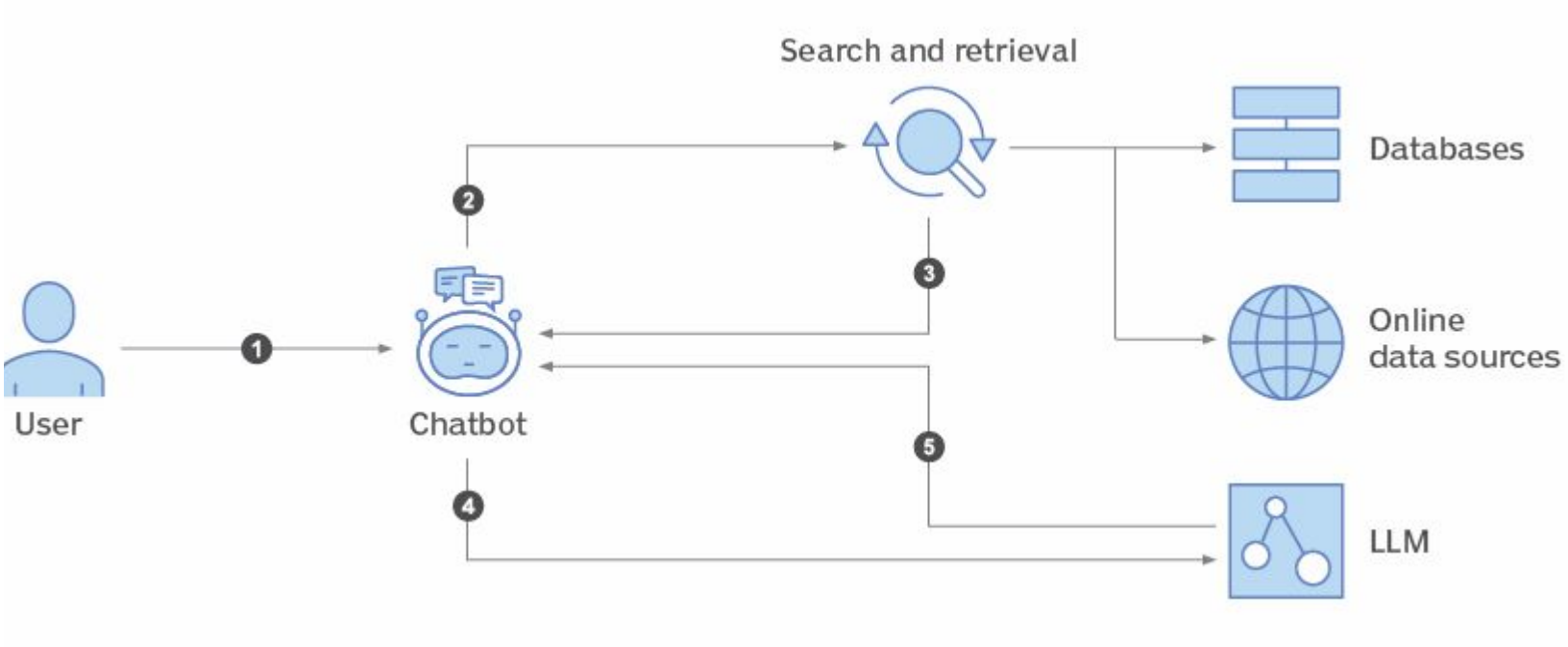
Submodularity in RAG settings

~Prayas Agrawal, 23m0757

RAG problem


Search and retrieval
Databases
Online
data sources
LLM
User
Chatbot
1
2
3
4
5

RAG problem

- Usually large retrieval corpus
- My aim is to apply submodular optimisation as a retrieval ranker
- Since corpus is large, opt for reranking approach

Primal-Dual relationship (Not standard terminology)

- Primal problem for countering RAG is having a retrieval corpus, and filtering out relevant documents before being passed as context to the language model. Common approach for filtering is having a **dedicated retriever model.**

- The dual of this problem is having the entire corpus (or a fraction of it) provided to the LM (**possibly** as context) and its the job of the language model to determine relevant documents from the very large corpus.
- Commonly falls under Long-Context learning

Related Work

- **Realm** - One of first papers implementing RAG
- **RAG** - Improves upon Realm
- **Unlimiformer** - Solves problem as “Dual”
- **Lin and Bilmes** - A class of submodular functions for document summarization.

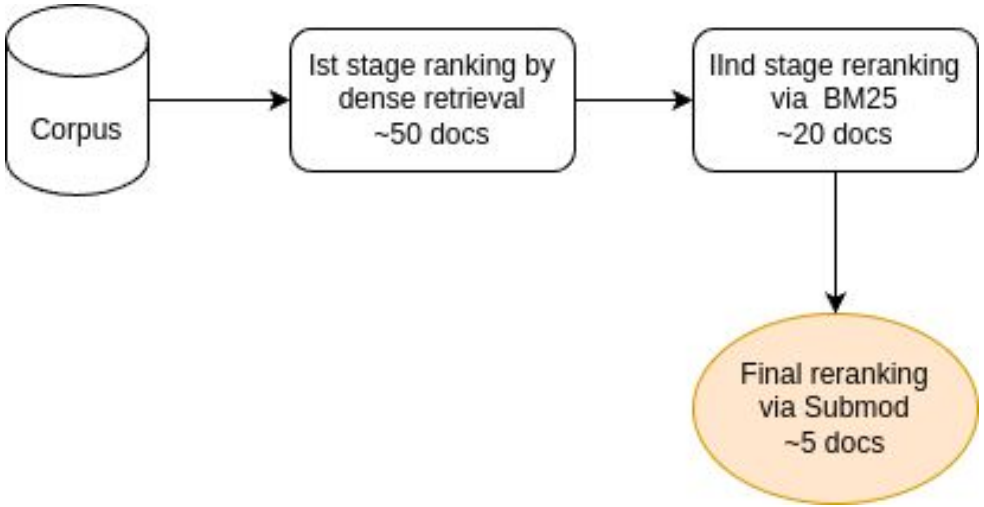
- Surprisingly, there seems to be **no work** exploring submod opt in RAG settings.

Dataset

- FEVER: Fact entailment and Verification
- Given a claim, retrieve evidence from wikipedia
- Output one of [Entailment, Neutral, Contradiction]
- 200k train instances
- 20k dev instances

Reranking approach

- Submod over Wikipedia(our retrieval corpus) too expensive
- Instead opt for reranking approach



Retriever and NLI Model

- Retriever: Evaluated both BM25 and a Dense retriever (MiniLM-L6)
- NLI Model: Deberta-v3-base
- Finetuned the NLI model on train set of FEVER

Submod functions (1)

- Facility Location MI, LogDet MI, GraphCut MI
- All evaluated over sparse and dense scores
- Similarity Matrix in sparse is BM25 similarity
- In dense its dot product similarity of embeddings
- Evaluate over two custom made functions as well

Submod functions (2), Query Relevant Similarity Measure:

- **Query Relevant Similarity Measure**
- For the MI versions, its sufficient to provide seperate, interdocument and cross query-document similarity matrices.
- However, FacilityLocationMI objective only considers ground set - input set and ground set - query set interactions and completely ignores input set - query set interactions.

Submod functions (2), Query Relevant Similarity Measure:

- LogDetMI can be tuned to favor query relevance over diversity.
-
$$logDetMI(A;Q) = \log\det(S_A) - \log\det\left(S_A - \eta^2 S_{A,Q} S_Q^{-1} S_{A,Q}^T\right)$$
-
- Second term controls the query relevance.

Submod functions (2), Query Relevant Similarity Measure:

- LogDetMI can be tuned to favor query relevance over diversity.
-
$$logDetMI(A;Q) = \log\det(S_A) - \log\det\left(S_A - \eta^2 S_{A,Q} S_Q^{-1} S_{A,Q}^T\right)$$
-
- Second term controls the query relevance.
- For **single** query, the relative importance of query and document sims is roughly: $sim(d_i, d_j) - c * \eta^2 * sim(q, d_i) * sim(q, d_j)$
-
- Above one is a difference of similarities, instead experiment with product of similarities

Submod functions (2), Query Relevant Similarity Measure:

- Directly model the document similarity matrix to include the query, not just retrieved documents.

$$prob(b|a) = softmax(sim(a,b);\ sim(a,v) \mid \forall v \in D)$$
$$s'_{ij}(d_i, d_j) = prob(d_i|q) * prob(d_j|q) * sim(d_i, d_j)$$
$$s_i = softmax(s'_i)$$

Submod functions (2), Query Relevant Similarity Measure:

- Directly model the document similarity matrix to include the query, not just retrieved documents.

$$prob(b|a) = softmax(sim(a,b);\ sim(a,v) \mid \forall v \in D)$$
$$s'_{ij}(d_i, d_j) = prob(d_i|q) * prob(d_j|q) * sim(d_i, d_j)$$
$$s_i = softmax(s'_i)$$

- Similarity is weighted by query relevance with respect to a pair of documents.
- Reflects the low score of a pair of docs which are highly similar but of little relevance to query.
-

Submod functions (2), Query Relevant Similarity Measure:

- Directly model the document similarity matrix to include the query, not just retrieved documents.

$$prob(b|a) = softmax(sim(a,b);\ sim(a,v) \mid \forall v \in D)$$
$$s'_{ij}(d_i, d_j) = prob(d_i|q) * prob(d_j|q) * sim(d_i, d_j)$$
$$s_i = softmax(s'_i)$$

- Similarity is weighted by query relevance with respect to a pair of documents.
- Reflects the low score of a pair of docs which are highly similar but of little relevance to query.
- Observe that this formulation neatly capture input set-query interactions as well, which was missing the MI variant of FacilityLocation
- Apply this directly to FL and LogDet **without** MI

Submod functions (3), Sparse Dense similarity:

- **Sparse Dense similarity**:
- Bag of words similarity models often outperform dense retrievals in situations where entity overlap highly correlates with the queries of the downstream task.

Submod functions (3), Sparse Dense similarity:

- **Sparse Dense similarity**:
- Bag of words similarity models often outperform dense retrievals in situations where entity overlap highly correlates with the queries of the downstream task.
- For this reason I experiment with a variant of dense and sparse retrieval which attempts to leverage the best of both worlds.

$$s_{ij} = \alpha * sparseProb(q, d_i, d_j) + (1 - \alpha) * denseProb(q, d_i, d_j)$$

Results

- Sparse ranking > dense reranking
- However, a Sparse-Dense similarity has a positive effect over just sparse retriever and dense retriever

Method	Accuracy
No Context	0.335
Dense FacilityLocationMI	0.491
Dense LogDetMI	0.522
Dense GraphCutMI	0.546
QR FacilityLocation	0.503
QR LogDet	0.531
Sparse FacilityLocationMI	0.512
Sparse LogDetMI	0.534
Sparse GraphCutMI	0.551
S-D FacilityLocationMI	0.519
S-D LogDetMI	0.541
S-D GraphCutMI	0.567
BART Large	0.64

Results

- Sparse ranking > dense reranking
- However, a Sparse-Dense similarity has a positive effect over just sparse retriever and dense retriever
- Confirms Hypothesis that both n-gram and semantic similarity are useful signals for the task

Method	Accuracy
No Context	0.335
Dense FacilityLocationMI	0.491
Dense LogDetMI	0.522
Dense GraphCutMI	0.546
QR FacilityLocation	0.503
QR LogDet	0.531
Sparse FacilityLocationMI	0.512
Sparse LogDetMI	0.534
Sparse GraphCutMI	0.551
S-D FacilityLocationMI	0.519
S-D LogDetMI	0.541
S-D GraphCutMI	0.567
BART Large	0.64

Results

- Sparse ranking > dense reranking
- However, a Sparse-Dense similarity has a positive effect over just sparse retriever and dense retriever
- Confirms Hypothesis that both n-gram and semantic similarity are useful signals for the task
- However to be noted is that alpha is 0.7 which implies we prefer sparse over dense by a large margin.

Method	Accuracy
No Context	0.335
Dense FacilityLocationMI	0.491
Dense LogDetMI	0.522
Dense GraphCutMI	0.546
QR FacilityLocation	0.503
QR LogDet	0.531
Sparse FacilityLocationMI	0.512
Sparse LogDetMI	0.534
Sparse GraphCutMI	0.551
S-D FacilityLocationMI	0.519
S-D LogDetMI	0.541
S-D GraphCutMI	0.567
BART Large	0.64

Results

- improvements with using QR which model cross interaction as multiplication rather than difference as in logDetMI

Method	Accuracy
No Context	0.335
Dense FacilityLocationMI	0.491
Dense LogDetMI	0.522
Dense GraphCutMI	0.546
QR FacilityLocation	0.503
QR LogDet	0.531
Sparse FacilityLocationMI	0.512
Sparse LogDetMI	0.534
Sparse GraphCutMI	0.551
S-D FacilityLocationMI	0.519
S-D LogDetMI	0.541
S-D GraphCutMI	0.567
BART Large	0.64

Results

- improvements with using QR which model cross interaction as multiplication rather than difference as in logDetMI
- Also improvements over FI-MI which ignores inputSet-query interactions completely.

Method	Accuracy
No Context	0.335
Dense FacilityLocationMI	0.491
Dense LogDetMI	0.522
Dense GraphCutMI	0.546
QR FacilityLocation	0.503
QR LogDet	0.551
Sparse FacilityLocationMI	0.512
Sparse LogDetMI	0.534
Sparse GraphCutMI	0.551
S-D FacilityLocationMI	0.519
S-D LogDetMI	0.541
S-D GraphCutMI	0.567
BART Large	0.64

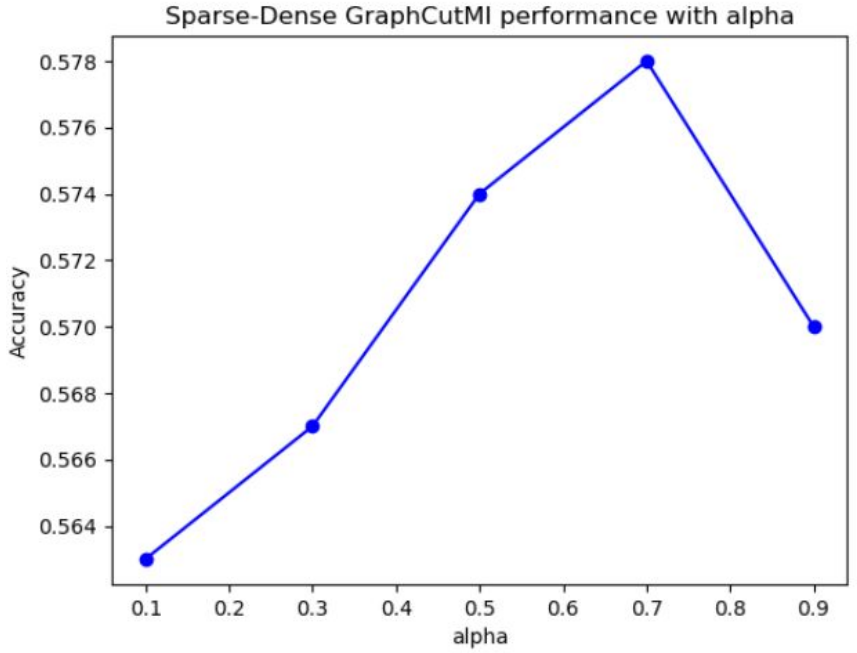
Results

- The best performing model is RAG BART with 0.64 label accuracy.
- However to be noted is that our model is 5 times smaller than BART.
- Future work may explore performance on larger models

Method	Accuracy
No Context	0.335
Dense FacilityLocationMI	0.491
Dense LogDetMI	0.522
Dense GraphCutMI	0.546
QR FacilityLocation	0.503
QR LogDet	0.551
Sparse FacilityLocationMI	0.512
Sparse LogDetMI	0.534
Sparse GraphCutMI	0.551
S-D FacilityLocationMI	0.519
S-D LogDetMI	0.541
S-D GraphCutMI	0.567
BART Large	0.64

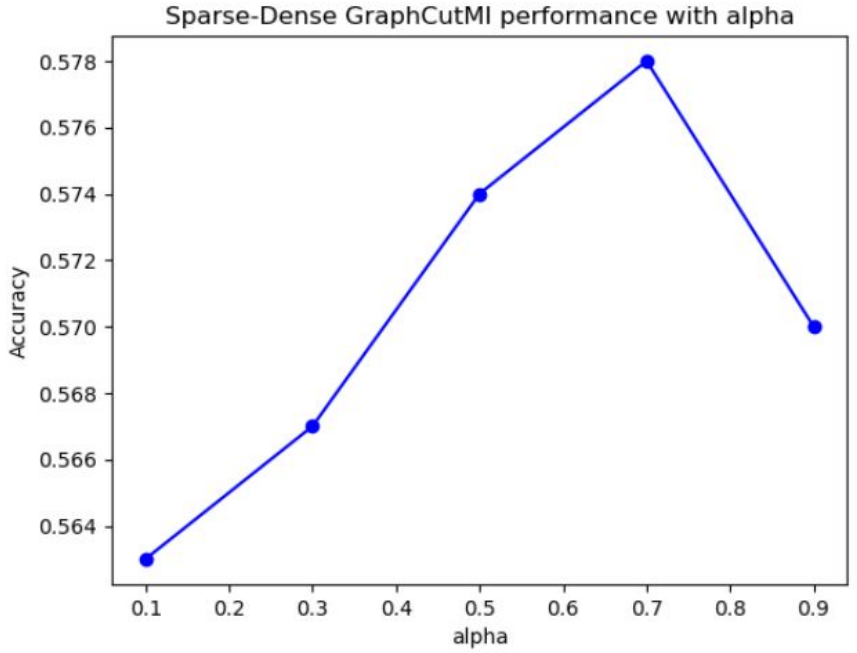
Tuning for sparse-dense parameter

- α in the Sparse-Dense setting.
- determined over 2000 examples of the dev dataset using Graph-CutMI.
- 0.7 performs best



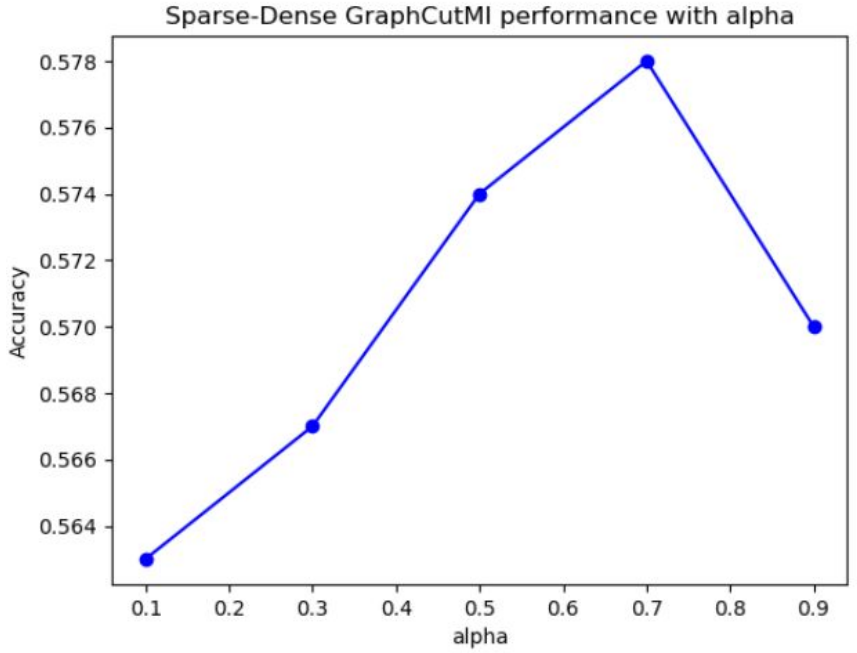
Tuning for sparse-dense parameter

- α in the Sparse-Dense setting.
- determined over 2000 examples of the dev dataset using Graph-CutMI.
- 0.7 performs best
- Thus n-gram overlap more advantageous than semantic signals,
- however pure n-gram signals are not better



Tuning for sparse-dense parameter

- α in the Sparse-Dense setting.
- determined over 2000 examples of the dev dataset using Graph-CutMI.
- 0.7 performs best
- Thus n-gram overlap more advantageous than semantic signals,
- however pure n-gram signals are not better
- implying downstream task still finds utility for semantic signals.



EXTRA: Dual formulation by Retrieval Scheduling

- This is a rough mathematical formulation of the retrieval scheduled RAG problem using submod optimisation.
- I dont provide results for it. Moreover, such a formulation seems to be **entirely new**, atleast in RAG literature.

EXTRA: Dual formulation by Retrieval Scheduling

- **Motivation and background**: Observe that the RAG problem poses no limitation on when to retrieve, and how to retrieve.
- Most papers retrieve at start once and append it as context.
- However its entirely possible to retrieve per token, and also even retrieve amidst forward pass of the transformer.

EXTRA: Dual formulation by Retrieval Scheduling

- **Motivation and background**: Observe that the RAG problem poses no limitation on when to retrieve, and how to retrieve.
- Most papers retrieve at start once and append it as context.
- However its entirely possible to retrieve per token, and also even retrieve amidst forward pass of the transformer.
- Ex: Whats the longest river of the seventh largest country in the world ?

EXTRA: Dual formulation by Retrieval Scheduling

- Unlimiformer, experiments with the dual variant of the RAG problem, commonly called as long context learning.
- The paper retrieves top-k encoder embeddings per head, per layer from the decoder of an encoder-decoder model.
- **No citation of the paper,** so far has yet considered retrieval scheduling, ie selectively choosing which heads to enforce retrieval upon.

EXTRA: Dual formulation by Retrieval Scheduling

- And the popular paper which does address this (**SelfRAG**), achieves this via data tuning
- Not mathematically motivated and structured
- Completely reliant on the whims of dataset

Ideal World and Method

- In an ideal world, given a query, a model **J** would output when to retrieve
- Thus, For a **N** layer decoder with **h** heads per layer, consider a **trainable** model **J**
- Model it is a MLP with inputs the input-query embeddings, layer, and head indices
- output the probability of head being active for retrieval

Ideal World and Submodular formulation

- For some metric, there are diminishing returns as the number of retrieval steps increase
- Also more number of retrievals can harm that metric
- Nonmonotone submodular
- **Sidenote**: Log likelihood **of LM** posses this anecdotally

Ideal World and Submodular formulation

For a head h_{ij} where i is the layer index, and j is the head index at the ith layer. Consider the set $\{h_{ij}\}$ of all the heads in a layer. Define the similarity between two heads (within same layer) as a measure of increase in model accuracy if the heads were active. $Acc_{active}(i, \{j, k\})$ denote the accuracy measure if heads j,k at ith layer were active for retrieval. Then, one such similarity measure could be:

$$s_i(j, k) = Acc_{active}(i, \{j, k\}) - Acc_{no\_retrieval} \tag{5}$$

Ideal World and Method

For a head h_{ij} where i is the layer index, and j is the head index at the ith layer. Consider the set $\{h_{ij}\}$ of all the heads in a layer. Define the similarity between two heads (within same layer) as a measure of increase in model accuracy if the heads were active. $Acc_{active}(i, \{j, k\})$ denote the accuracy measure if heads j,k at ith layer were active for retrieval. Then, one such similarity measure could be:

$$s_i(j, k) = Acc_{active}(i, \{j, k\}) - Acc_{no\_retrieval} \tag{5}$$

Thus, our first step is to create a training data. The training data is simply $(q_i, \{r_{jk}\})$, where for the ith query, we determine per layer j the ideal retriever heads k. These heads are obtained via submod optimisng the above formulation.

Ideal World and Method

Now to train the J network, we simply minimise the cross entroy loss over the ideal heads and heads what J thinks are appropriate.

Note however that calculating Acc_{active} can be expensive for the whole dataset, and for every pair of head, thus a reasonable option is to approximate it via markov samples over train data or even a subset selection problem itself. Also observe that the number of attention heads is usually small. Example: 32 for LLama70b, thus making the problem somewhat tractable.

Remarks

- J is the only trainable component
- At inference, we sample a multinomial from the J and threshold appropriately to reduce the number of retrieval steps, thus in effect pursuing retrieval scheduling.

Remarks

- J is the only trainable component
- At inference, we sample a multinomial from the J and threshold appropriately to reduce the number of retrieval steps, thus in effect pursuing retrieval scheduling.
- Retrieval scheduling in RAG settings (using a mathematical formulation) and modelling head indices directly as part of scheduling **submodular** objectives **seems to be new**.

Conclusion

- Explore submod in RAG settings and compare various functions
- Two custom functions
- Our small model competitive with larger ones
- Propose a entirely new retrieval scheduling framework for the **Dual Problem** (Long context learning vs RAG)

- Future Work: Bigger models, Bigger Datasets, and the above Retrieval scheduling formulation

Thank You