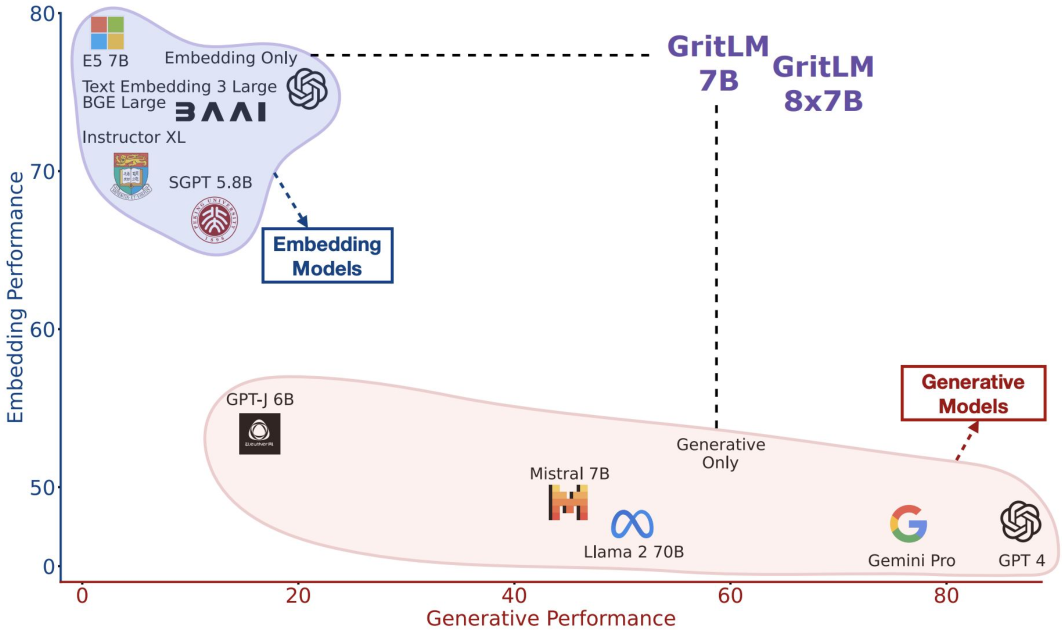
Generative Representational Instruction Tuning

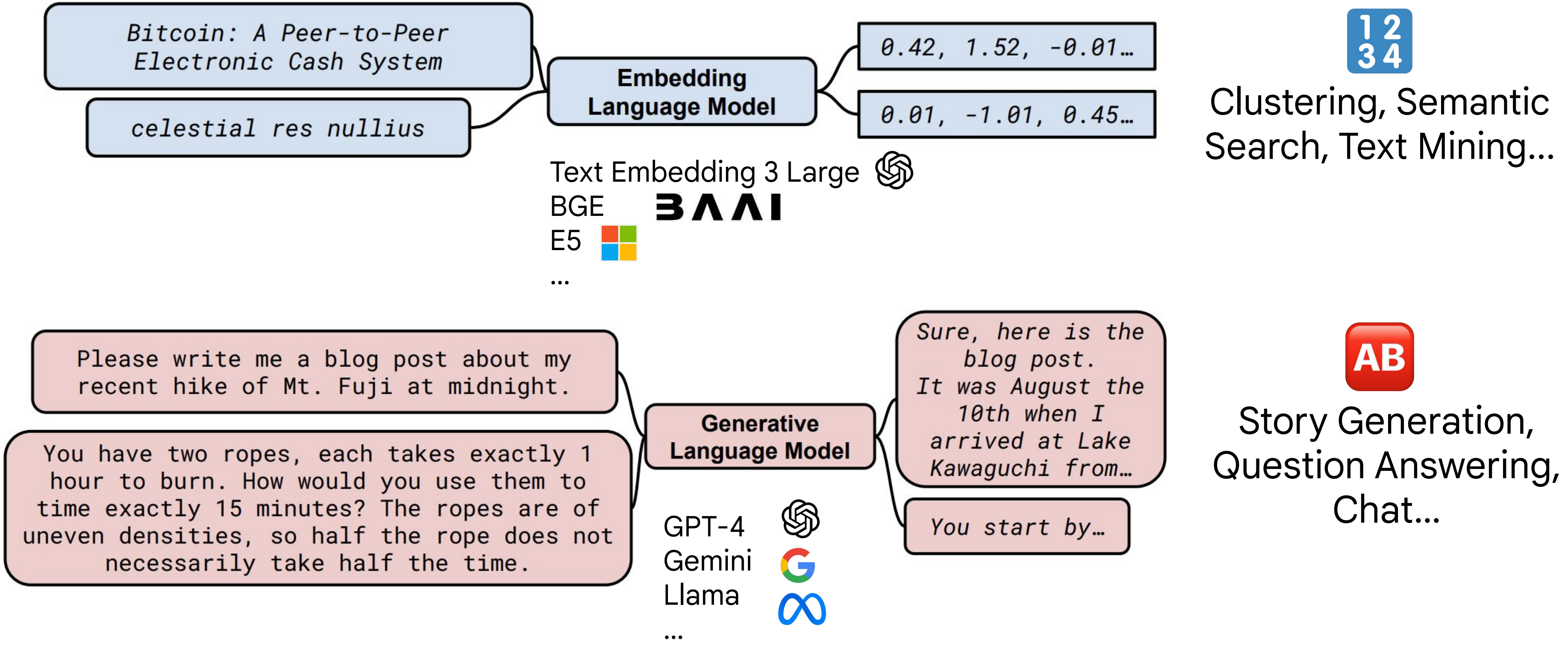
Niklas Muennighoff (Twitter: @Muennighoff)
Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, Douwe Kiela



Paper

This talk is being recorded.

Two types of language models



Advantages of combining them

Performance:
Get better on both?

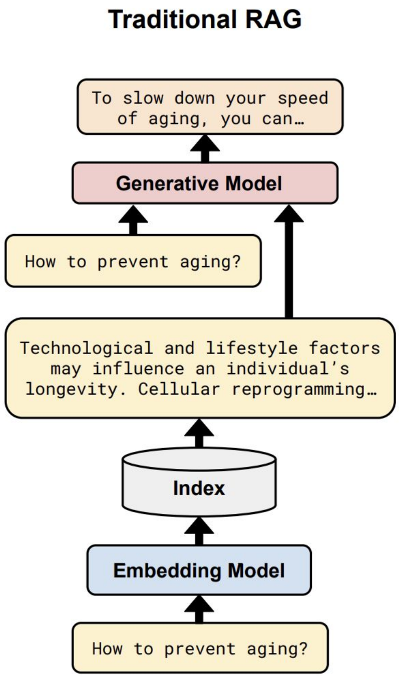
Embedding benchmarks: MTEB..

MTEB
Massive Text Embedding Benchmark

Generation benchmarks: AlpacaEval..

AlpacaFarm

Efficiency:
Speed-up joint use cases



Simplicity:
Unify endpoints

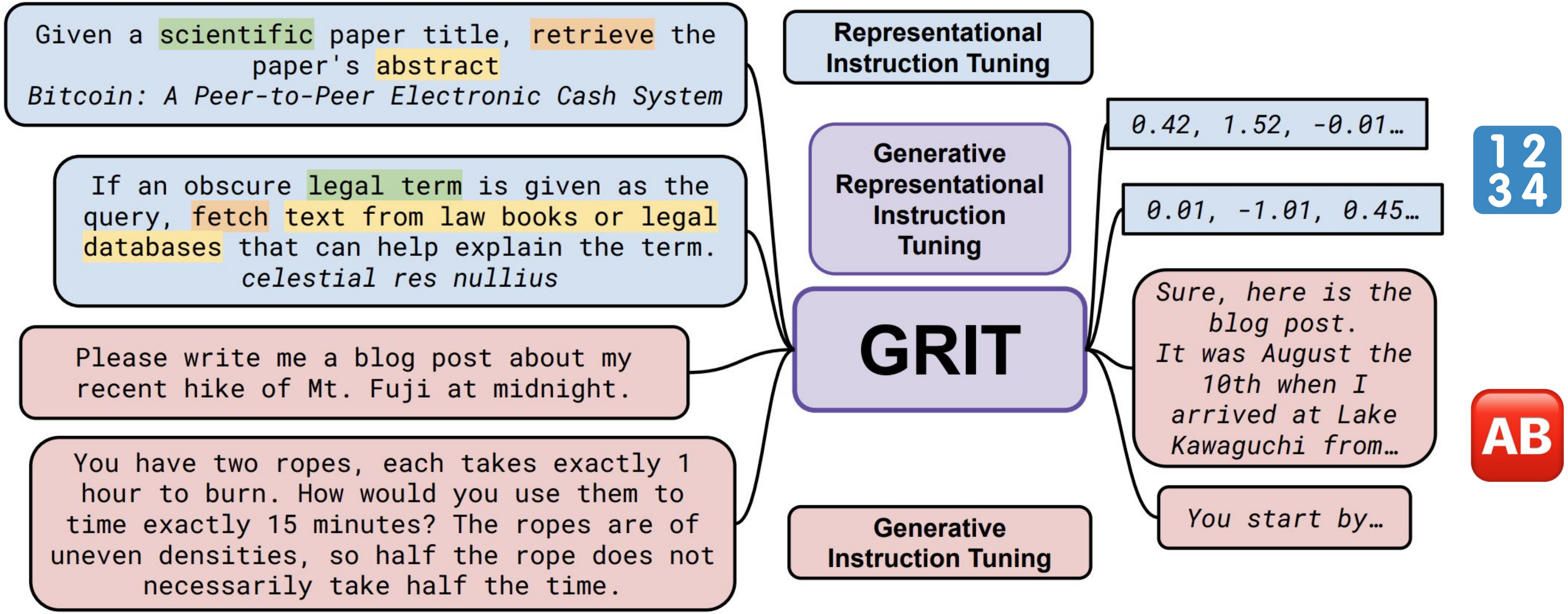
Embedding endpoint

```
curl https://api.openai.com/v1/embeddings \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "input": "Your text string goes here",
    "model": "text-embedding-3-small"
  }'
```

Generation endpoint

```
curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-3.5-turbo",
    "messages": [
      {
        "role": "system",
        "content": "You are a helpful assistant."
      },
      {
```

Unifying representation & generation

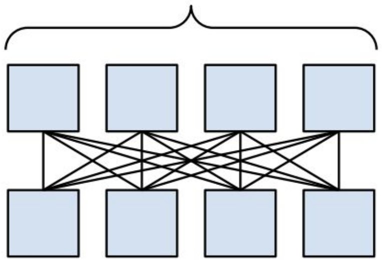


Embedding instructions ideally contain: Domain, intent & unit

Differing instructions, format & attention

Representation

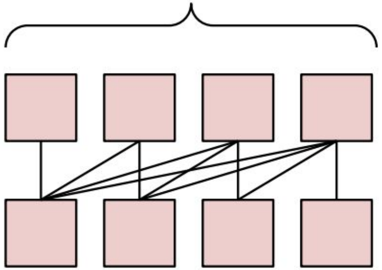
Mean Pooling



<s><|user|>
{instruction}
<|embed|>
{sample to represent}

Generation

Language Modeling Head



<s><|user|>
{instruction}
<|assistant|>
{response}</s>
<|user|>
...

Combining losses

Contrastive embedding loss with hard negatives

$$\mathcal{L}_{\text{Rep}} = -\frac{1}{M}\sum_{i=1}^{M} \log \frac{\exp(\tau \cdot \sigma(f_\theta(q^{(i)}), f_\theta(d^{(i)})))}{\sum_{j=1}^{M} \exp(\tau \cdot \sigma(f_\theta(q^{(i)}), f_\theta(d^{(j)})))}$$

Next token prediction loss for generation

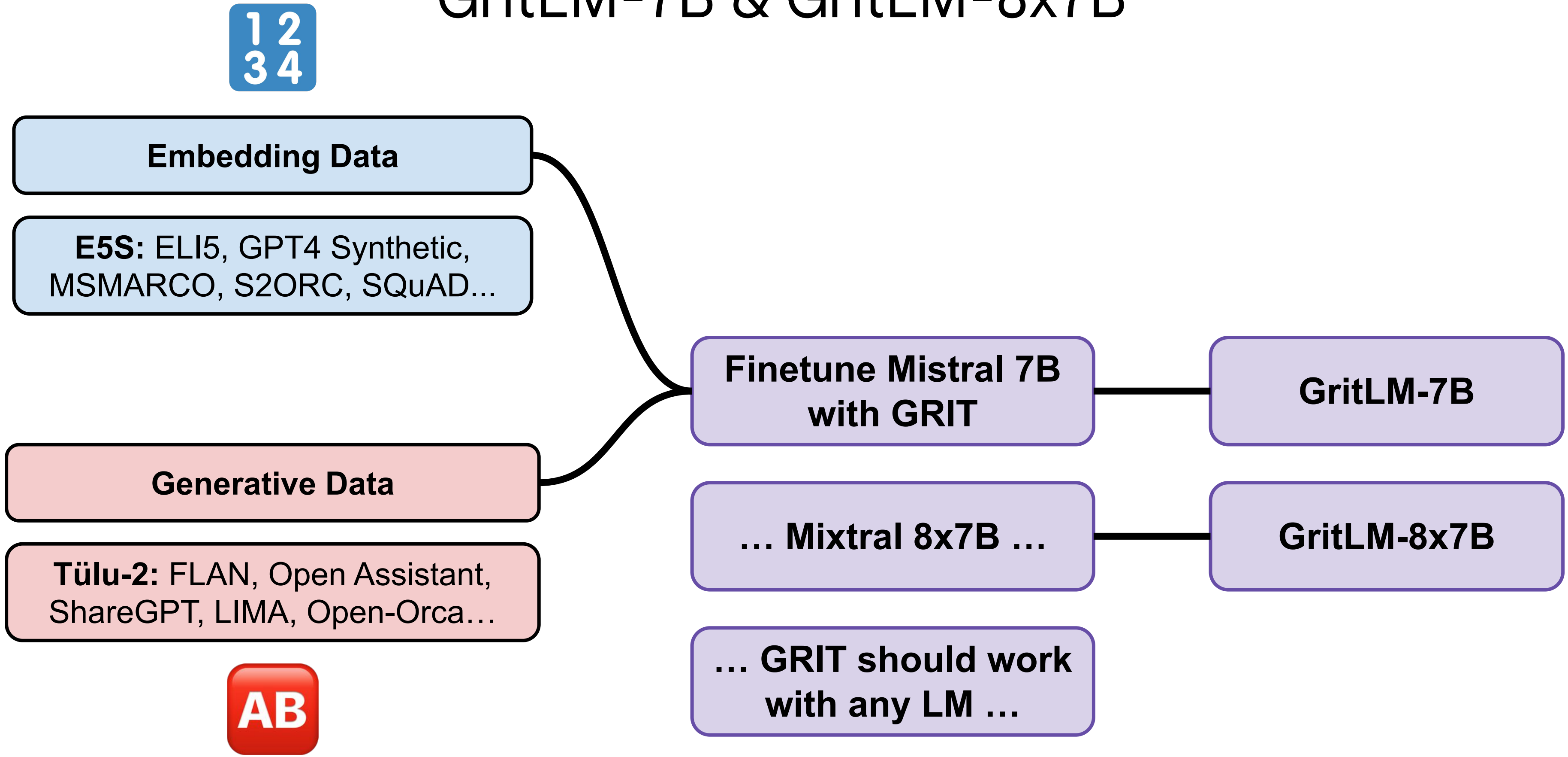
$$\mathcal{L}_{\text{Gen}} = -\frac{1}{N}\sum_{i=1}^{N} \log P(f_{\theta,\eta}(x^{(i)}) | f_{\theta,\eta}(x^{(<i)}))$$

Combining both with adjustable weights

$$\mathcal{L}_{\text{GRIT}} = \lambda_{\text{Rep}}\mathcal{L}_{\text{Rep}} + \lambda_{\text{Gen}}\mathcal{L}_{\text{Gen}}$$

🔢 🆎

GritLM-7B & GritLM-8x7B



GritLM-7B & GritLM-8x7B

Embedding Performance

Task (→) Metric (→) Dataset # (→)	CLF Acc. 12	Clust. V-Meas. 11	PairCLF AP 3	Rerank MAP 4	Retrieval nDCG 15	STS Spear. 10	Summ. Spear. 1	Avg. 56
Proprietary models♥								
OpenAI v3	75.5	49.0	85.7	59.2	55.4	81.7	29.9	64.6
Other Open Models♥								
Llama 2 70B	60.4	29.0	47.1	38.5	9.0	49.1	26.1	35.6
Mistral 7B	63.5	34.6	53.5	43.2	13.2	57.4	19.7	40.5
Mistral 7B Instruct	67.1	34.6	59.6	44.8	16.3	63.4	25.9	43.7
GPT-J 6B	66.2	39.0	60.6	48.9	19.8	60.9	26.3	45.2
SGPT BE 5.8B	68.1	40.3	82.0	56.6	50.3	78.1	31.5	58.9
Instructor XL 1.5B	73.1	44.7	86.6	57.3	49.3	83.1	**32.3**	61.8
BGE Large 0.34B	76.0	46.1	87.1	60.0	54.3	83.1	31.6	64.2
E5 Mistral 7B	78.5	50.3	**88.3**	60.2	56.9	**84.6**	31.4	66.6
GRITLM								
Gen.-only 7B	65.4	32.7	54.2	43.0	13.7	60.2	21.1	41.2
Emb.-only 7B	78.8	**51.1**	87.1	**60.7**	**57.5**	83.8	30.2	**66.8**
GRITLM 7B	**79.5**	50.6	87.2	60.5	57.4	83.4	30.4	**66.8**
GRITLM 8X7B	78.5	50.1	85.0	59.8	55.1	83.3	29.8	65.7

Generative Performance

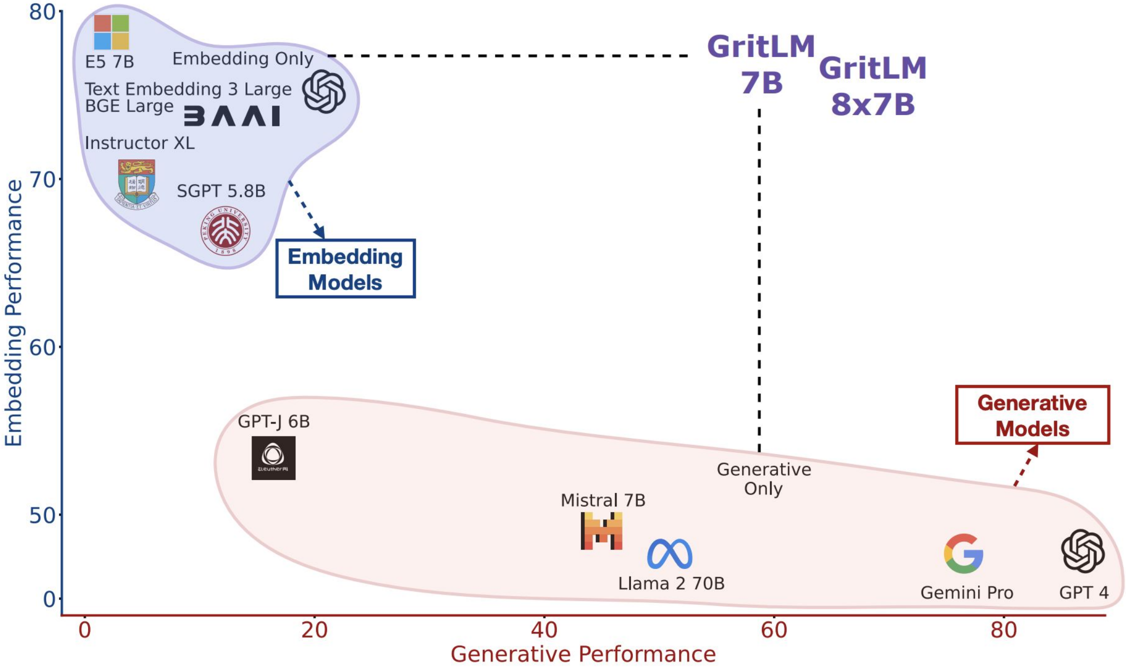
Dataset (→) Setup (→) Metric (→)	MMLU 0 FS EM	GSM8K 8 FS, CoT EM	BBH 3 FS, CoT EM	TyDi QA 1 FS, GP F1	HumanEval 0 FS pass@1	Alpaca 0 FS, 1.0 % Win	Avg.
Proprietary models♥							
GPT-4-0613	81.4	95.0	89.1	65.2	86.6†	91.2	84.8
Other Open Models♥							
Zephyr 7B β	58.6	28.0	44.9	23.7	28.5	85.8	44.9
Llama 2 70B	64.5	55.5	66.0	**62.6**	29.9♦	0.0	46.4
Llama 2 Chat 13B	53.2	9.0	40.3	32.1	19.6†	91.4	40.9
Llama 2 Chat 70B	60.9	59.0	49.0	44.4	34.3†	94.5	57.0
Tülu 2 13B	55.4	46.0	49.5	53.2	31.4	78.9	52.4
Tülu 2 70B	67.3	**73.0**	68.4	53.6	41.6	86.6	65.1
Mistral 7B Inst.	53.0	36.0	38.5	27.8	34.0	75.3	44.1
Mixtral 8x7B Inst.	**68.4**	65.0	55.9	24.3	**53.5**	**94.8**	60.3
GRITLM							
Emb.-only 7B	23.5	1.0	0.0	21.0	0.0	0.0	7.6
Gen.-only 7B	57.5	52.0	55.4	56.6	34.5	75.4	55.2
GRITLM 7B	57.6	57.5	54.8	55.4	32.8	74.8	55.5
GRITLM 8X7B	66.7	61.5	**70.2**	58.2	53.4	84.0	**65.7**

GritLM-7B & GritLM-8x7B

Embedding Performance

Generative Performance

Task (→) Metric (→) Dataset # (→)	CLF Acc. 12	Clust. V-Meas. 11
OpenAI v3	75.5	49.0
Llama 2 70B	60.4	29.0
Mistral 7B	63.5	34.6
Mistral 7B Instruct	67.1	34.6
GPT-J 6B	66.2	39.0
SGPT BE 5.8B	68.1	40.3
Instructor XL 1.5B	73.1	44.7
BGE Large 0.34B	76.0	46.1
E5 Mistral 7B	78.5	50.3
Gen.-only 7B	65.4	32.7
Emb.-only 7B	78.8	**51.1**
GRITLM 7B	**79.5**	50.6
GRITLM 8X7B	78.5	50.1

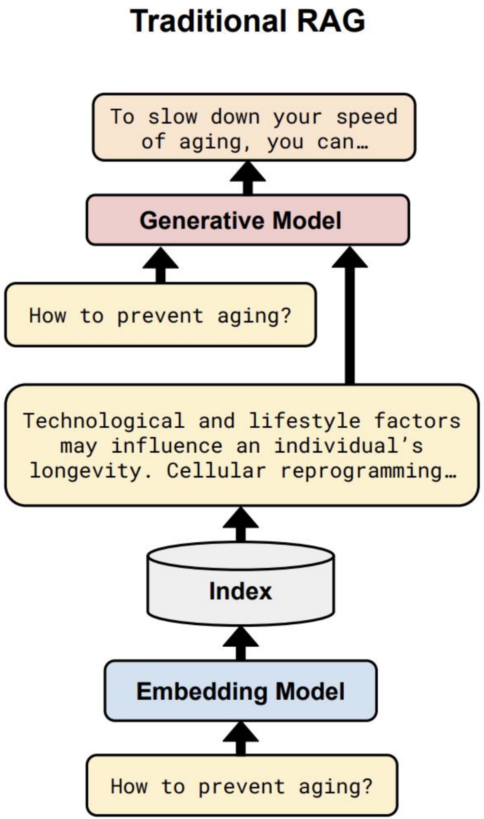


QA GP	HumanEval 0 FS pass@1	Alpaca 0 FS, 1.0 % Win	Avg.
	86.6†	91.2	84.8
	28.5	85.8	44.9
	29.9♦	0.0	46.4
	19.6†	91.4	40.9
	34.3†	94.5	57.0
	31.4	78.9	52.4
	41.6	86.6	65.1
	34.0	75.3	44.1
	53.5	**94.8**	60.3
	0.0	0.0	7.6
	34.5	75.4	55.2
	32.8	74.8	55.5
	53.4	84.0	**65.7**

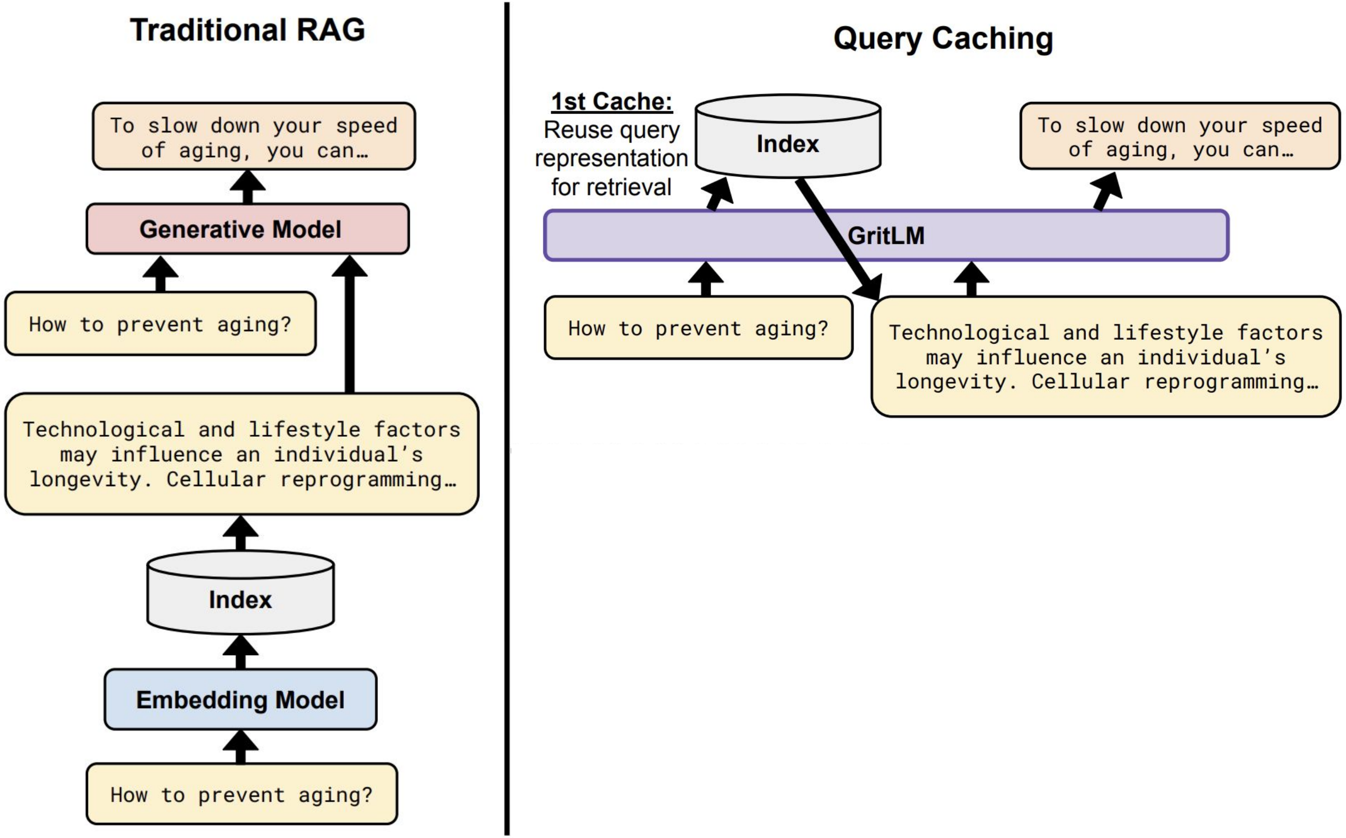
Questions thus far?

Next: RAG with GRIT

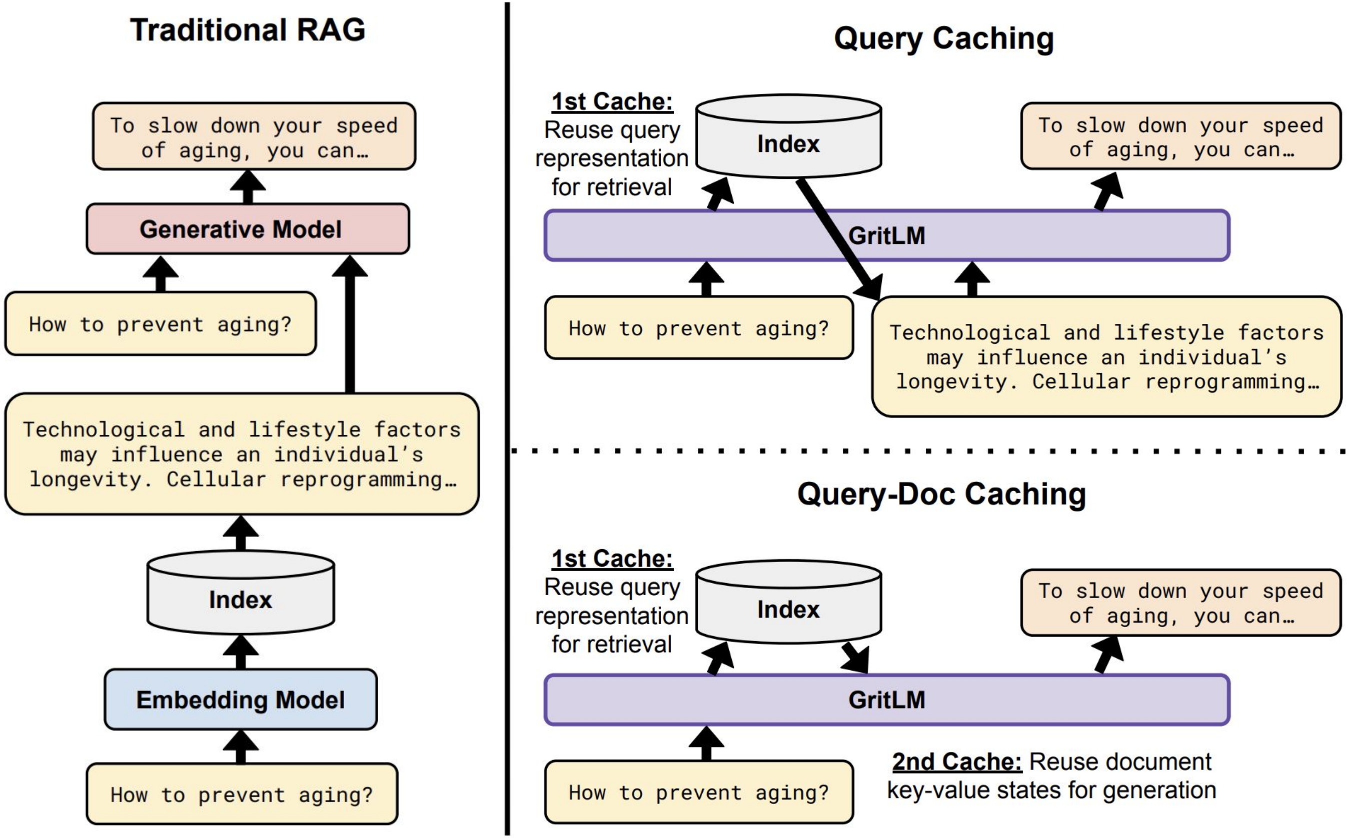
RAG with GRIT



RAG with GRIT



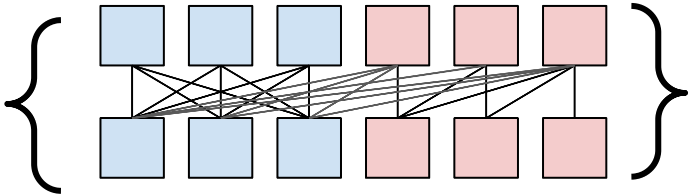
RAG with GRIT



Attention mismatch problem

1) Combining bidirectional & causal attention

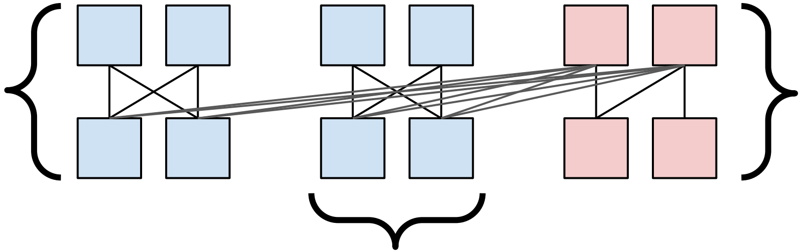
Embed query/doc bidirectionally & cache



Reuse bidirectionally attended cache for causal generation

2) Combining separately attended texts (only if caching both, query-doc/doc-query)

Embed query & cache



Reuse separately attended caches for causal generation

Embed doc & cache

Caching Performance

Query Caching Generation Prompt

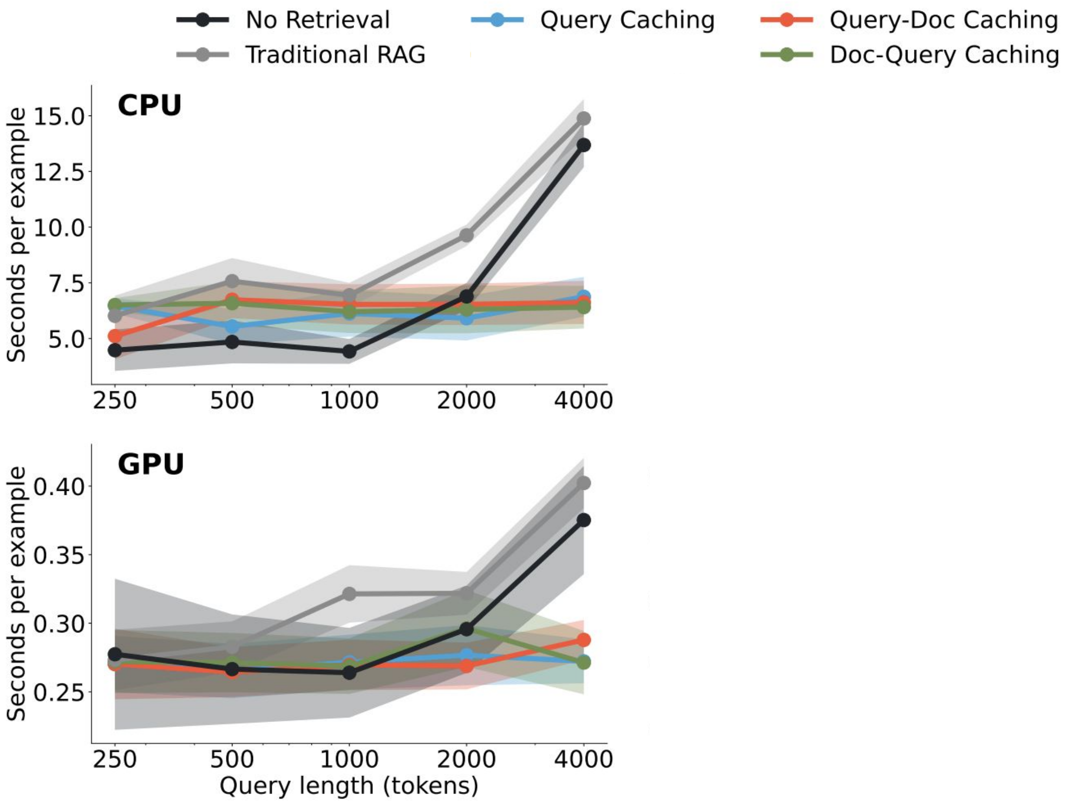
```
<|user|>
GRIT is…
Optionally using the prior
context answer the query
prior to it
<|assistant|>….
```

Doc Caching Generation Prompt

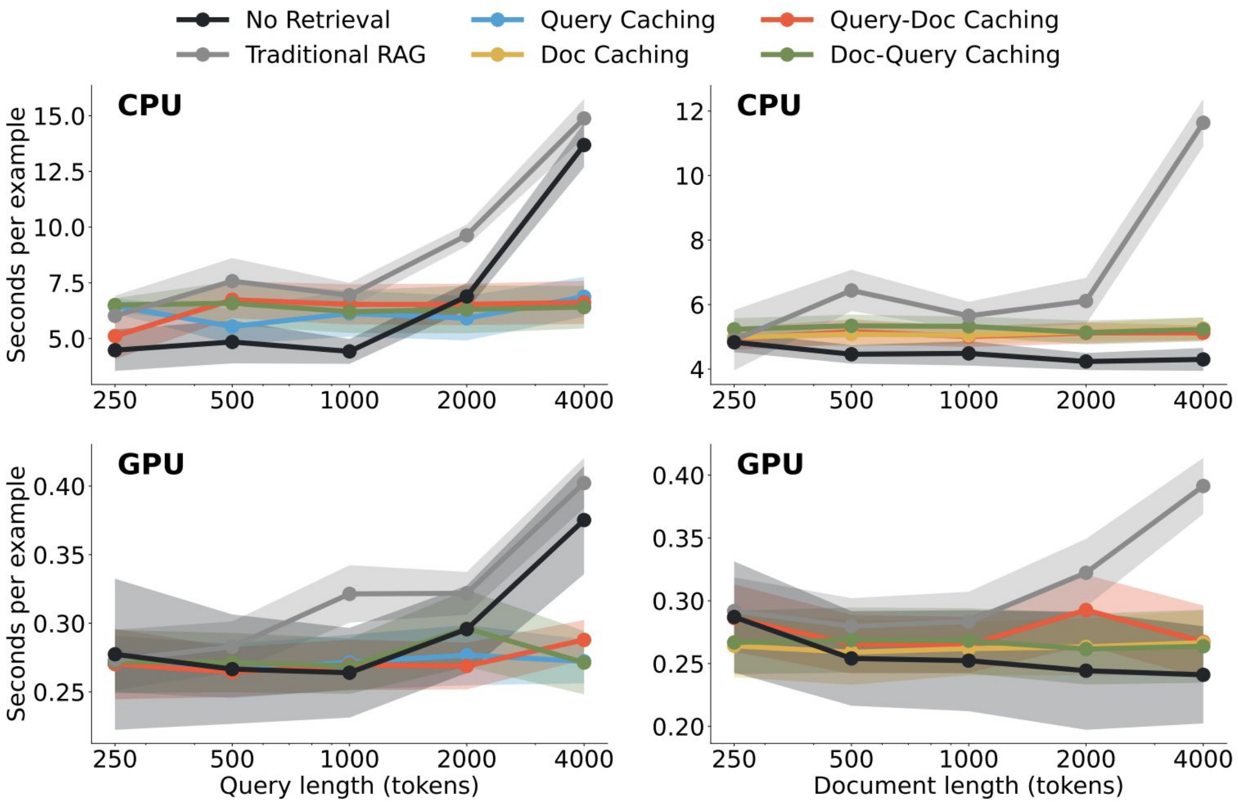
```
<|user|>
What is "GRIT"?
Answer the prior query while
optionally using the context
prior to it
<|assistant|>….
```

	Match (0-shot, ↑)
No RAG	21.00
RAG	30.50
Query Caching	25.46
Query-Doc Caching	21.63
RAG	30.47
Doc Caching	**33.38**
Doc-Query Caching	18.39

Scaling Query Length



Scaling Document Length



Questions thus far?

Next: GRIT Ablations

Ablations: Attention & Pooling

Attention Emb		Attention Gen		Pooling	Emb	Gen
Instruction	Sample	Instruction	Sample			
Embedding Only						
Causal				Wmean	60.0	-
Causal	Bidirectional			Mean	61.0	-
Bidirectional				Mean	61.8	-

Ablations: Attention & Pooling

Attention Emb		Attention Gen		Pooling	🔢 Emb	🆎 Gen
Instruction	Sample	Instruction	Sample			
Embedding Only						
Causal				Wmean	60.0	-
Causal	Bidirectional			Mean	61.0	-
Bidirectional				Mean	61.8	-
Generative Only						
		Causal			-	**55.2**
		Bidirectional	Causal		-	50.7

Ablations: Attention & Pooling

<table>
<tr><th colspan="2">Attention Emb</th><th colspan="2">Attention Gen</th><th rowspan="2">Pooling</th><th rowspan="2">🔢
Emb</th><th rowspan="2">🆎
Gen</th></tr>
<tr><th>Instruction</th><th>Sample</th><th>Instruction</th><th>Sample</th></tr>
<tr><td colspan="7">Embedding Only</td></tr>
<tr><td colspan="2">Causal</td><td colspan="2"></td><td>Wmean</td><td>60.0</td><td>-</td></tr>
<tr><td>Causal</td><td>Bidirectional</td><td colspan="2"></td><td>Mean</td><td>61.0</td><td>-</td></tr>
<tr><td colspan="2">Bidirectional</td><td colspan="2"></td><td>Mean</td><td>61.8</td><td>-</td></tr>
<tr><td colspan="7">Generative Only</td></tr>
<tr><td colspan="2"></td><td colspan="2">Causal</td><td></td><td>-</td><td>55.2</td></tr>
<tr><td colspan="2"></td><td>Bidirectional</td><td>Causal</td><td></td><td>-</td><td>50.7</td></tr>
<tr><td colspan="7">Unified</td></tr>
<tr><td colspan="2">Causal</td><td colspan="2">Causal</td><td>Last token</td><td>61.2</td><td>53.0</td></tr>
<tr><td colspan="2">Causal</td><td colspan="2">Causal</td><td>Wmean</td><td>62.8</td><td>52.8</td></tr>
<tr><td colspan="2">Bidirectional</td><td colspan="2">Causal</td><td>Mean</td><td>64.0</td><td>52.9</td></tr>
</table>

Ablations: Base Model

Variant	Emb	Gen
Mistral 7B	**54.6**	**22.4**
Llama 2 7B	48.2	20.8
GPT-J 6B	51.9	14.0

Finetuned with GRIT

Embedding Performance after GRIT ≠ Raw performance

Task (→)	CLF	Clust.	...	Avg.
Metric (→)	Acc.	V-Meas.		
Dataset # (→)	12	11		56
Llama 2 70B	60.4	29.0		35.6
Mistral 7B	63.5	34.6		40.5
Mistral 7B Instruct	67.1	34.6		43.7
GPT-J 6B	66.2	39.0		45.2

Only pretrained

Ablations: Base Dataset

Dataset	Emb
MEDI	64.0
MEDI2	64.7
E5	**66.0**

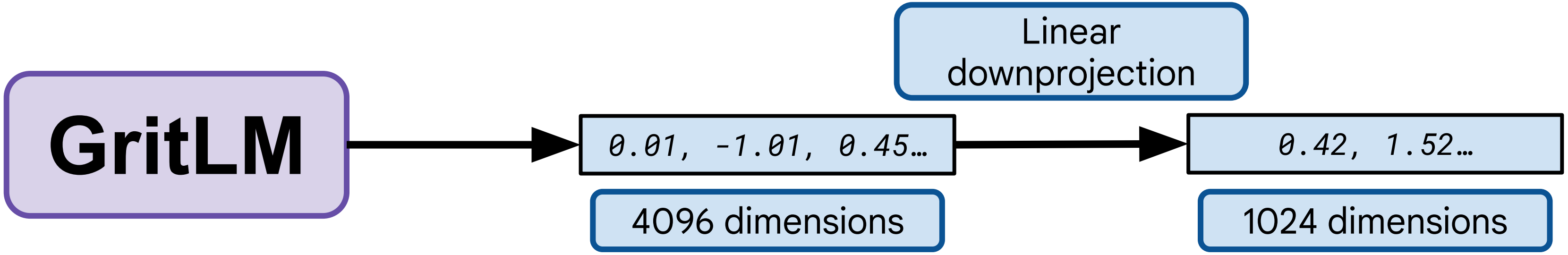
Includes GPT-4 generated samples

Dataset	Gen
Tülu 2	**55.2**
OASST	37.7
UltraChat	47.4

Collection of many instruction datasets

Ablations: Embedding Head

Variant	🔢 Emb	🆎 Gen
No head	**62.7**	**49.2**
-> 1024	62.1	48.0



Ablations: Batch Size (BS)

BS Emb:Gen	🔢 Emb	🆎 Gen
256:256	63.2	**53.4**
4096:256	**64.2**	53.3

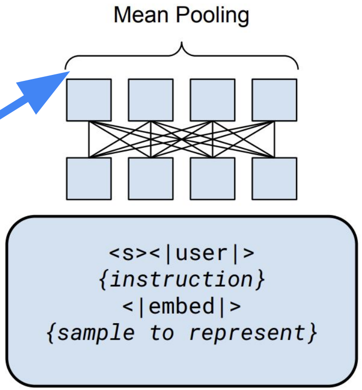
Increases M
➡️ more negatives

$$\mathcal{L}_{\text{Rep}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{\exp(\tau\cdot\sigma(f_\theta(q^{(i)}), f_\theta(d^{(i)})))}{\sum_{j=1}^{M}\exp(\tau\cdot\sigma(f_\theta(q^{(i)}), f_\theta(d^{(j)})))}$$

Ablations: Precision

Precision	Emb	Gen
FP32	66.3	52.4
BF16*	**66.5**	**55.0**

***Pooling & cosine similarity still in FP32 i.e. cast BF16->FP32 here**



Ablations: In-Batch-Negatives (IBN)

IBN origin	Emb	Gen
Any dataset	**66.0**	50.9
Same dataset	**66.0**	**51.1**

Massive boost on Retrieval though:

Retrieval nDCG
54.9
56.2

Ablations: Format

Format	Gen
Tülu 2	**55.2**
Zephyr β	49.0

Additional end-of-sequence token after user utterance

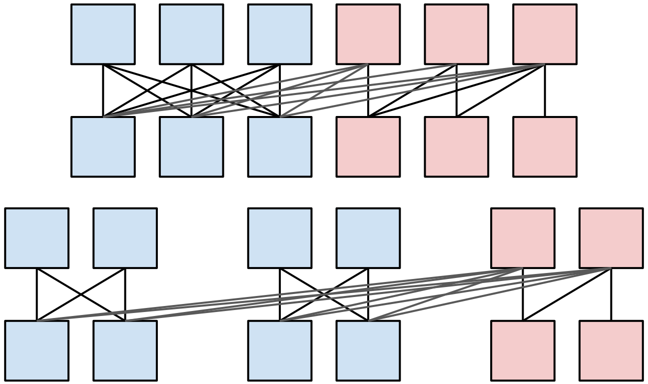
Ablations: Loss

Gen loss type	$\mathcal{L}_{\text{Rep}}/\mathcal{L}_{\text{Gen}}$	Emb	Gen
Token	2.4	66.1	54.4
Token	6.0	66.5	55.0
Mix (32 -> 8)	4.1	**66.7**	**55.4**

Ratio adjusted via weights:

$$\mathcal{L}_{\text{GRIT}} = \lambda_{\text{Rep}}\mathcal{L}_{\text{Rep}} + \lambda_{\text{Gen}}\mathcal{L}_{\text{Gen}}$$

Future Directions



1) Solving the attention mismatch issues

Simple finetuning may suffice?

2) GritLM Agents

Teaching the model to invoke its own embedding capabilities, maybe via finetuning

3) GRIT Doc Caching with multiple documents

Likely solved once the attention mismatch issue is solved

Thanks!

More about GRIT:

Paper

Open-source code:
ContextualAI/gritlm

Open-source models:
hf.co/GritLM 🤗

Niklas Muennighoff (Twitter: @Muennighoff)
Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, Douwe Kiela

Reranking with GritLM further boosts performance

Table 3: **Reranking (Rerank) using GritLM as both Bi- and Cross-Encoder.**

MTEB DS (↓)	No Rerank	Rerank top 10
ArguAna	63.24	**64.39**
ClimateFEVER	30.91	**31.85**
CQADupstack	49.42	**50.05**
DBPedia	46.60	**47.82**
FiQA2018	59.95	**60.39**
FEVER	82.74	**82.85**
HotpotQA	79.40	**80.46**
NFCorpus	40.89	**41.23**
NQ	70.30	**71.49**
MSMARCO	41.96	**42.47**
QuoraRetrieval	**89.47**	88.67
SCIDOCS	24.41	**24.54**
SciFact	79.17	**79.28**
TRECCOVID	74.80	**75.24**
Touche2020	27.93	**28.41**
Average	57.4	**57.9**

Few-shot embedding does not work

Table 4: **Few-shot embedding.** The 12 MTEB datasets ("DS") are grouped by the 7 main MTEB tasks in the same order as in Table 1.

Train DS (→)	E5S		MEDI2	
MTEB DS (↓)	0 FS	1 FS	0 FS	1 FS
Banking77	**88.5**	88.3	**88.1**	87.9
Emotion	**52.8**	51.0	**52.5**	51.9
IMDB	**95.0**	93.9	**94.3**	92.2
BiorxivS2S	**39.8**	39.4	**37.6**	37.4
SprintDup.	93.0	**94.9**	95.2	**95.7**
TwitterSem	**81.1**	77.9	**76.8**	73.9
TwitterURL	**87.4**	87.1	85.9	**86.1**
ArguAna	**63.2**	51.7	**53.5**	53.2
SCIDOCS	**24.4**	19.7	**25.5**	25.5
AskUbuntu	**67.3**	64.7	**66.6**	66.0
STS12	77.3	**78.0**	**76.6**	73.5
SummEval	**30.4**	29.5	29.1	**31.5**

GRIT + KTO

Table 5: **Aligning GRITLM with KTO after GRIT.** The upper table depicts embedding performance while the lower depicts generative performance.

Task (→) Metric (→) Dataset # (→)	CLF Acc. 12	Clust. V-Meas. 11	PairCLF AP 3	Rerank MAP 4	Retrieval nDCG 15	STS Spear. 10	Summ. Spear. 1	Avg. 56
GritLM-7B	79.5	50.6	87.2	60.5	57.4	83.4	30.4	66.8
GritLM-7B-KTO	79.6	50.1	87.1	60.5	57.1	83.5	30.5	66.7
GritLM-8x7B	78.5	50.1	85.0	59.8	55.1	83.3	29.8	65.7
GritLM-8x7B-KTO	78.7	50.0	84.4	59.4	54.1	82.5	30.8	65.2

Dataset (→) Setup (→) Metric (→)	MMLU 0 FS EM	GSM8K 8 FS, CoT EM	BBH 3 FS, CoT EM	TyDi QA 1 FS, GP F1	HumanEval 0 FS pass@1	Alpaca 0 FS, 1.0 % Win	Avg.
GritLM-7B	57.6	57.5	54.8	55.4	32.8	74.8	55.5
GritLM-7B-KTO	57.6	57.5	55.4	55.8	31.5	86.7	57.4
GritLM-8x7B	66.7	61.5	70.2	58.2	53.4	84.0	65.7
GritLM-8x7B-KTO	66.8	79.5	67.1	31.4	56.8	95.3	66.2