Improving Text Embeddings with Large Language Models

Liang Wang, Nan Yang, Xiaolong Huang,
Linjun Yang, Rangan Majumder, Furu Wei
Microsoft Corporation
{wangliang,nanya,xiaolhu,yang.linjun,ranganm,fuwei}@microsoft.com

Presented by Yasmine Belghmaidi and Kasey Lowe

Background

What are Text Embeddings?

Text embeddings are numerical vector representations that capture the semantic meaning of text

- Transform words, sentences, or documents into dense vectors (e.g., 768 or 4096 dimensions)
- Semantically similar texts have similar vectors (close in vector space)
- Enable machines to "understand" and compare text mathematically

Real World Applications

Google Search

- **Your query:** "best Italian restaurants nearby"
- **Behind the scenes:** Query converted to embedding vector → Matched against millions of indexed document embeddings → Returns most relevant results
- **Why it matters:** Understands semantic intent, not just keyword matching
    - "Fix my bike" matches "bicycle repair guide" even without shared words

Netflix Recommendations

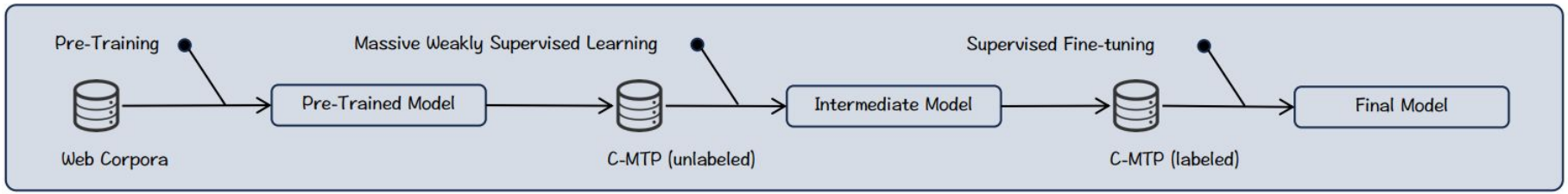
Customer Support Chatbots

Previous Methods

- Take weighted average of word embeddings
  - Misses contextual information of natural language
- Fine-tune BERT on natural language inference (NLI) datasets (Sentence-BERT, SimCSE)
  - BERT-style encoders have been surpassed by more modern LLMs and techniques
  - Limited context length

Previous Methods

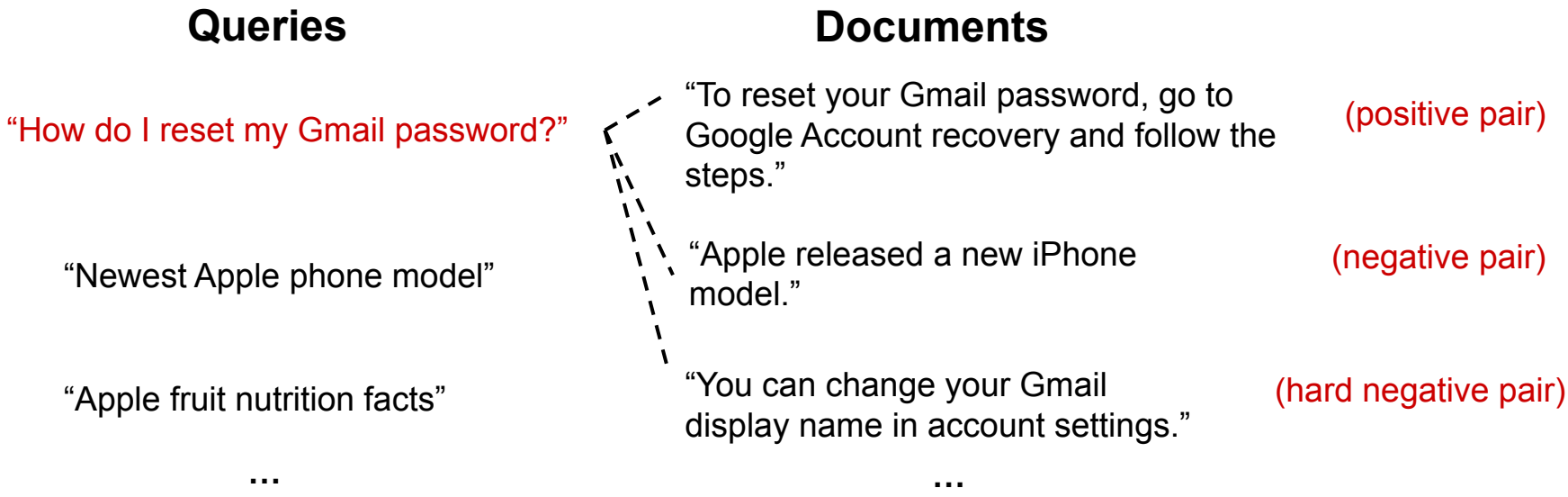
- Multi-stage approaches (E5, BGE, etc.)
  - Generally involve contrastive pre-training on unlabeled query/passage pairs and fine-tuning on labeled data
  - Rely on manually-collected datasets



Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighof. 2023. C-pack: Packaged resources to advance general chinese embedding. ArXiv preprint, abs/2309.07597

Contrastive Pre-Training

- Task: Given a query and a set of documents, find the document that best matches/responds to the query



Solution Overview

The Two-Step Method

Step 1: Task Brainstorming

- Use GPT-4 to generate diverse retrieval task definitions
- Covers wide range of scenarios (search, classification, recommendations, etc.)
- Creates ~150,000 unique task descriptions

Step 2: Example Generation

- For each task, GPT-4 generates concrete training examples
- Each example contains: query + positive document + hard negative document
- Produces ~500,000 high-quality synthetic examples across 93 languages

Step 3: Model Training

- Fine-tune a powerful LLM (Mistral-7B) on the synthetic data
- Simple, single-stage training (< 1,000 steps)
- Optional: Add real labeled datasets for additional boost

Data Generation - Step 1

Brainstorm a list of potentially useful text retrieval tasks.
Here are a few examples for your reference:
- Provided a scientific claim as query, retrieve documents that help verify or refute the claim.
- Search for documents that answers a FAQ-style query on children's nutrition.

Please adhere to the following guidelines:
- Specify what the query is, and what the desired documents are.
- Each retrieval task should cover a wide range of queries, and should not be too specific.

Your output should always be a python list of strings only, with about 20 elements, and each element corresponds to a distinct retrieval task in one sentence. Do not explain yourself or output anything else. Be creative!

["Retrieve company's financial reports for a given stock ticker symbol.",
"Given a book name as a query, retrieve reviews, ratings and summaries of that book.",
"Search for scientific research papers supporting a medical diagnosis for a specified disease."
... *(omitted for space)*]

Data Generation - Step 2

You have been assigned a retrieval task: *{task}*
Your mission is to write one text retrieval example for this task in JSON format. The JSON object must contain the following keys:
- **"user_query"**: a string, a random user search query specified by the retrieval task.
- **"positive_document"**: a string, a relevant document for the user query.
- **"hard_negative_document"**: a string, a hard negative document that only appears relevant to the query.

Please adhere to the following guidelines:
- The "user_query" should be *{query_type}*, *{query_length}*, *{clarity}*, and diverse in topic.
- All documents should be at least *{num_words}* words long.
- Both the query and documents should be in *{language}*.

... (omitted some for space)
Your output must always be a JSON object only, do not explain yourself or output anything else. Be creative!

{**"user_query":** "How to use Microsoft Power BI for data analysis",
"positive_document": "Microsoft Power BI is a sophisticated tool that requires time and practice to master. In this tutorial, we'll show you how to navigate Power BI ... *(omitted)* ",
"hard_negative_document": "Excel is an incredibly powerful tool for managing and analyzing large amounts of data. Our tutorial series focuses on how you...*(omitted)*" }

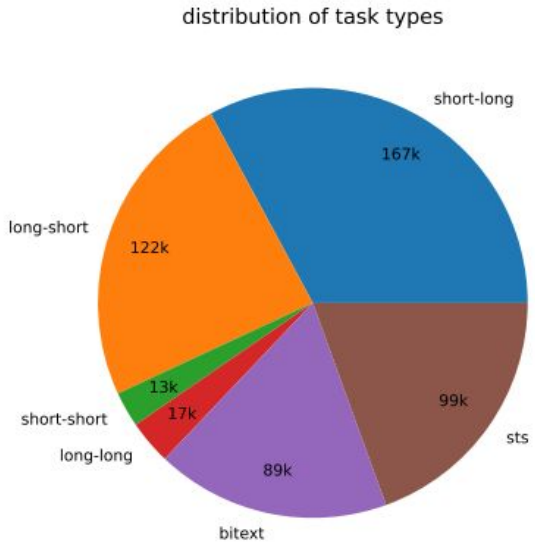
Types of tasks

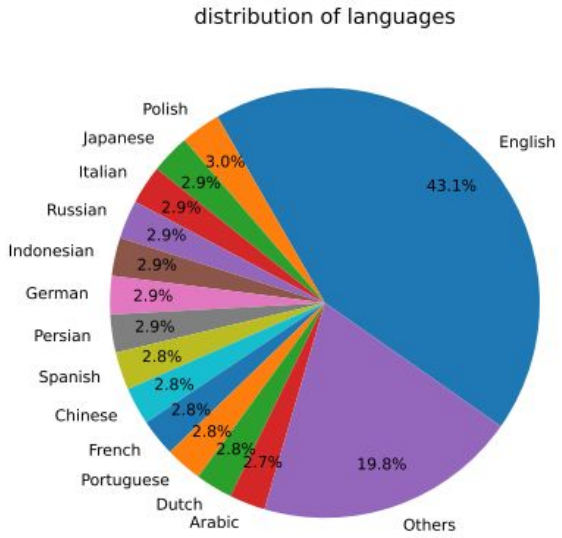
- **Symmetric tasks:** query and document have very similar semantic meanings but different surface forms
  - Monolingual semantic textual similarity (STS): evaluating semantic similarity of two texts
  - Bitext retrieval: finding semantically similar sentences in texts with different languages
- **Asymmetric tasks:** query and document are semantically related but not direct paraphrases of each other

	Short document	Long document
Short query	Short-short Ex. find movie title based on quote	Short-long Ex. find articles to answer search engine query
Long query	Long-short Ex. label customer complaint severity	Long-long Ex. given legal brief, find documents presenting similar arguments

Synthetic Data

- ~500k examples with ~150k unique instructions





Training

Base Model Selection : **Mistral-7B** (7 billion parameters, decoder-only LLM)

- Extensively pre-trained on trillions of tokens
- Already has strong semantic representations

Adding Instructions to Queries

q_inst = Instruct: {task_definition} And Query: {user_query}

Example :

Instruct: Given a book name as query, retrieve reviews, ratings and summaries of that book

Query: 1984 book review

Training

Embedding Extraction

1. Append **[EOS]** token to text
2. Pass through Mistral-7B
3. Extract **last-token hidden state** from final layer
4. → Produces **4096-dimensional embedding vector**

Why last token? In decoder models, it has "seen" all previous tokens

Goal: Pull positive pairs close, push negative pairs apart

Training Objective: InfoNCE Loss

$$\min \quad \mathbb{L} = -\log \frac{\phi(q^+_{\text{inst}}, d^+)}{\phi(q^+_{\text{inst}}, d^+) + \sum_{n_i \in \mathbb{N}} (\phi(q^+_{\text{inst}}, n_i))} \quad \phi(q, d) = \exp(\frac{1}{\tau} \cos(\mathbf{h}_q, \mathbf{h}_d))$$

Hyperparameter Influence

Datasets	Class.	Clust.	PairClass.	Rerank	Retr.	STS	Summ.	Avg
$E5_{mistral\text{-}7b}$	78.3	49.9	87.1	**59.5**	52.2	81.2	**32.7**	64.5
w/ LLaMA-2 7b init.	76.2	48.1	85.1	58.9	49.6	81.2	30.8	$62.9^{-1.6}$
w/ msmarco data only	71.6	47.1	86.1	58.8	54.4	79.5	31.7	$62.7^{-1.8}$
pooling type								
w/ mean pool	77.0	48.9	86.1	59.2	52.4	**81.4**	30.8	$64.1^{-0.4}$
w/ weighted mean	77.0	49.0	86.1	59.2	52.0	**81.4**	30.2	$64.0^{-0.5}$
LoRA rank								
w/ r=8	**78.4**	**50.3**	87.1	59.3	**53.0**	81.0	31.7	$\mathbf{64.8}^{+0.3}$
w/ r=32	**78.4**	**50.3**	**87.4**	**59.5**	52.2	81.2	30.6	$64.6^{+0.1}$
instruction type								
w/o instruction	72.3	47.1	82.6	56.3	48.2	76.7	30.7	$60.3^{-4.2}$
w/ task type prefix	71.1	46.5	79.7	54.0	52.7	73.8	30.0	$60.3^{-4.2}$

Table 5: Results on the MTEB benchmark with various hyperparameters. The first row corresponds to the default setting, which employs last-token pooling, LoRA rank 16, and natural language instructions. Unless otherwise stated, all models are trained on the synthetic and MS-MARCO passage ranking data.

Key Training Insights

1. **Instructions are Critical**
   - With instructions: **64.5 MTEB**
   - Without instructions: **60.3 MTEB** (-4.2 points!)
   - Natural language instructions inform the model about task context

2. **LoRA Enables Efficient Training**
   - Train only 0.6% of model parameters
   - Fast convergence, low memory

Key Training Insights

3. No Contrastive Pre-training Needed

- Traditional methods (E5, BGE): Pre-train on billions of pairs (weeks)
- This approach: Direct fine-tuning (18 hours)
- **Why?** Mistral already has excellent representations from LLM pre-training

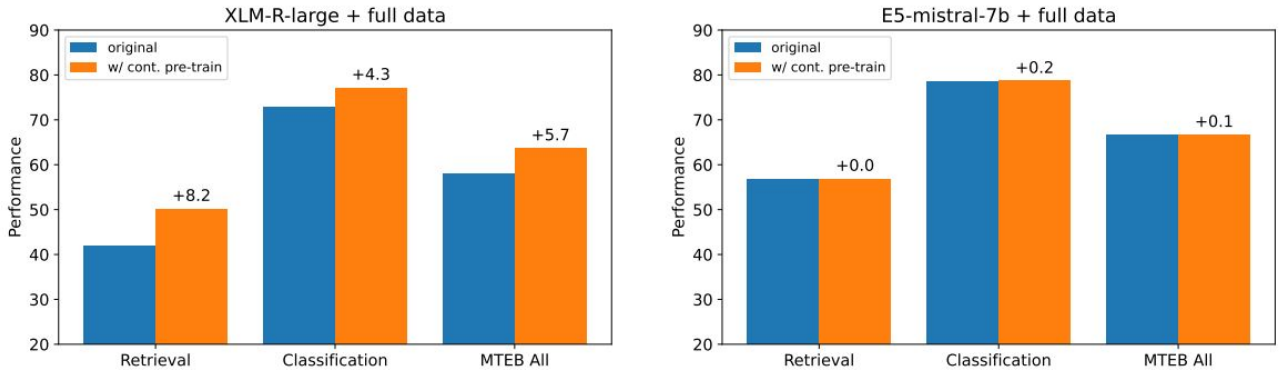


Figure 3: Effects of contrastive pre-training. Detailed numbers are in Appendix Table 7.

Experiment

Three Training Configurations:

- **Synthetic Only:** 500k examples
- **Synthetic + MSMARCO:** ~1M examples
- **Full Data (SOTA):** ~1.8M examples

These configurations were evaluated on the MTEB against many other previous solutions.

Performance

# of datasets →	Class. 12	Clust. 11	PairClass. 3	Rerank 4	Retr. 15	STS 10	Summ. 1	Avg 56
Unsupervised Models								
Glove (Pennington et al., 2014)	57.3	27.7	70.9	43.3	21.6	61.9	28.9	42.0
$\text{SimCSE}_{\text{bert-unsup}}$ (Gao et al., 2021)	62.5	29.0	70.3	46.5	20.3	74.3	31.2	45.5
Supervised Models								
$\text{SimCSE}_{\text{bert-sup}}$ (Gao et al., 2021)	67.3	33.4	73.7	47.5	21.8	79.1	23.3	48.7
Contriever (Izacard et al., 2021)	66.7	41.1	82.5	53.1	41.9	76.5	30.4	56.0
GTR_{xxl} (Ni et al., 2022b)	67.4	42.4	86.1	56.7	48.5	78.4	30.6	59.0
$\text{Sentence-T5}_{\text{xxl}}$ (Ni et al., 2022a)	73.4	43.7	85.1	56.4	42.2	82.6	30.1	59.5
$\text{E5}_{\text{large-v2}}$ (Wang et al., 2022b)	75.2	44.5	86.0	56.6	50.6	82.1	30.2	62.3
$\text{GTE}_{\text{large}}$ (Li et al., 2023)	73.3	46.8	85.0	59.1	52.2	83.4	31.7	63.1
$\text{BGE}_{\text{large-en-v1.5}}$ (Xiao et al., 2023)	76.0	46.1	87.1	60.0	54.3	83.1	31.6	64.2
Ours								
$\text{E5}_{\text{mistral-7b}}$ + full data	**78.5**	50.3	**88.3**	**60.2**	**56.9**	**84.6**	31.4	**66.6**
w/ synthetic data only	78.2	**50.5**	86.0	59.0	46.9	81.2	31.9	63.1
w/ synthetic + msmarco	78.3	49.9	87.1	59.5	52.2	81.2	**32.7**	64.5

Table 1: Results on the MTEB benchmark (Muennighoff et al., 2023) (56 datasets in the English subset). The numbers are averaged for each category. Please refer to Table 17 for the scores per dataset.

MTEB Leaderboard Performance

Model	BEIR	MTEB
OpenAI text-embedding-3-large	55.4	64.6
Cohere-embed-english-v3.0	55.0	64.5
voyage-lite-01-instruct	55.6	64.5
UAE-Large-V1	54.7	64.6
$E5_{\text{mistral-7b}}$ + full data	**56.9**	**66.6**

Table 3: Comparison with commercial models and the model that tops the MTEB leaderboard (as of 2023-12-22) (Li and Li, 2023). "BEIR" is the average nDCG@10 score over 15 public datasets in the BEIR benchmark (Thakur et al., 2021). "MTEB" is the average score over 56 datasets in the English subset of the MTEB benchmark (Muennighoff et al., 2023). For the commercial models listed here, little details are available on their model architectures and training data.

Long Context Experiment

- Previous embedding models (and datasets) are largely limited to shorter contexts
- To evaluate performance on longer contexts, the authors developed a novel **personalized passkey retrieval** task

Query: what is the pass key for Malayah Graves?

Doc1: <prefix filler> Malayah Graves's pass key is 123. Remember it. 123 is the pass key for Malayah Graves. <suffix filler>
Doc2: <prefix filler> Cesar McLean's pass key is 456. Remember it. 456 is the pass key for Cesar McLean. <suffix filler>
.....

<prefix filler> and <suffix filler> are repeats of “The grass is green. The sky is blue. The sun is yellow. Here we go. There and back again.”

Long Context Performance

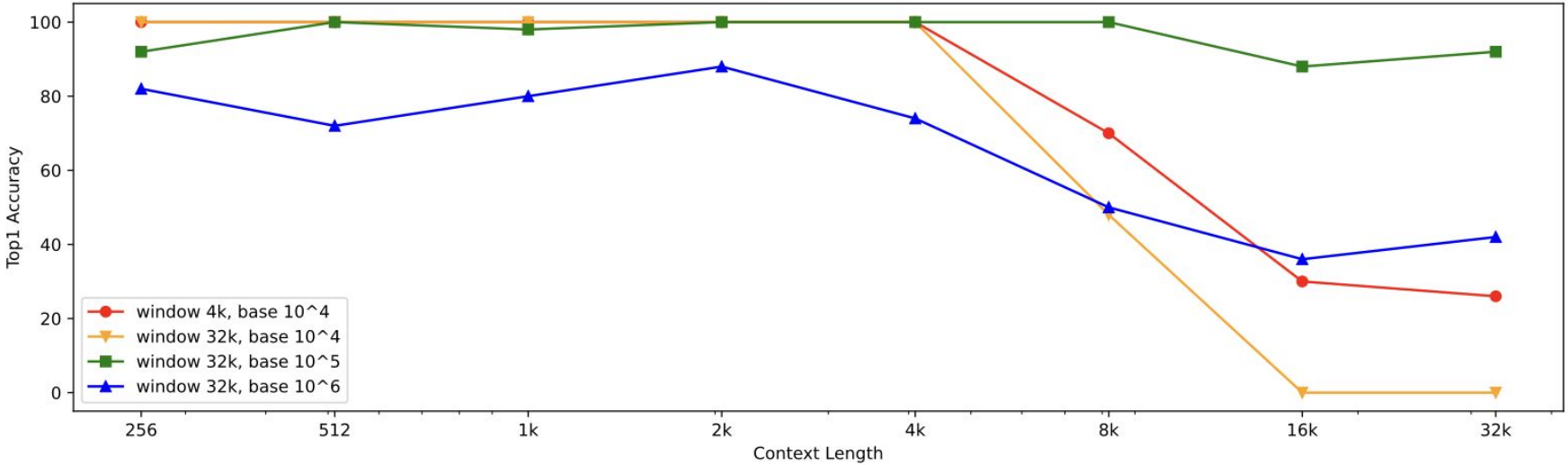


Figure 5: Accuracy of personalized passkey retrieval as a function of input context length. For each context length, we randomly generate 50 queries and compute the top-1 accuracy.

Multilingual Retrieval Performance

	High-resource Languages				Low-resource Languages			
	en	fr	es	ru	te	hi	bn	sw
BM25 (Zhang et al., 2023b)	35.1	18.3	31.9	33.4	49.4	45.8	50.8	38.3
mDPR (Zhang et al., 2023b)	39.4	43.5	47.8	40.7	35.6	38.3	44.3	29.9
$mE5_{base}$ (Wang et al., 2024)	51.2	49.7	51.5	61.5	75.2	58.4	70.2	71.1
$mE5_{large}$ (Wang et al., 2024)	52.9	54.5	**52.9**	67.4	**84.6**	**62.0**	**75.9**	**74.9**
$E5_{mistral\text{-}7b}$ + full data	**57.3**	**55.2**	52.2	**67.7**	73.9	52.1	70.3	68.4

Table 2: nDCG@10 on the dev set of the MIRACL dataset for both high-resource and low-resource languages. We select the 4 high-resource languages and the 4 low-resource languages according to the number of candidate documents. The numbers for BM25 and mDPR come from Zhang et al. (2023b). For the complete results on all 16 languages, please see Table 6.

Limitations and future work

- Synthetic data is limited by the LLM used to generate it and may encode any biases the in LLM
- LLM-based text embedding methods are more computationally intensive than BERT-based methods
- Manual prompt engineering is needed to generate synthetic data

Conclusion

- **Proves a New Paradigm**
  - LLM-generated data can **replace** expensive human annotation
- **Simplifies embedding process**
  - No multi-stage pre-training needed—just generate data and fine-tune

Key Message :

This paper shows that in the era of LLMs, the bottleneck is no longer data collection, but creative prompt engineering.