2695 Introduction to Machine Learning
Masters Program in Economics, Finance and Management

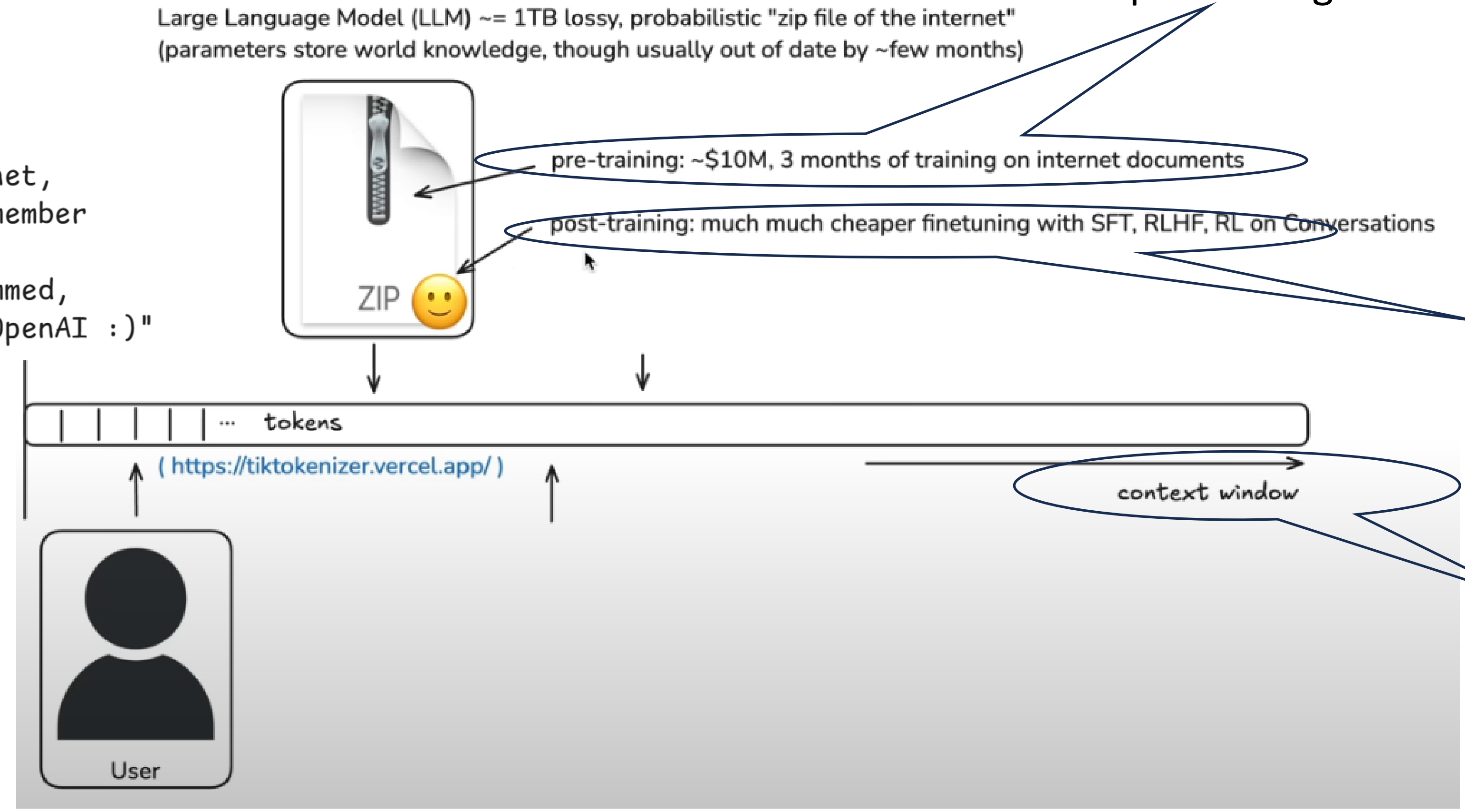
NOVA SCHOOL OF BUSINESS & ECONOMICS

USING LLMs

Using ChatGPT

knowledge from pretraining

```
"Hi I am ChatGPT.
I am a 1 terabyte zip file.
My knowledge comes from the internet,
which I read ~6 months ago and remember
only vaguely.
My winning personality was programmed,
by example, by human labelers at OpenAI :)"
```



Large Language Model (LLM) ~= 1TB lossy, probabilistic "zip file of the internet"
(parameters store world knowledge, though usually out of date by ~few months)

pre-training: ~$10M, 3 months of training on internet documents

post-training: much much cheaper finetuning with SFT, RLHF, RL on Conversations

personality from post training

new chat

working model

How I use LLMs, Andrej Karpathy

https://www.youtube.com/watch?v=EWvNQjAaOHw

On using LLMs

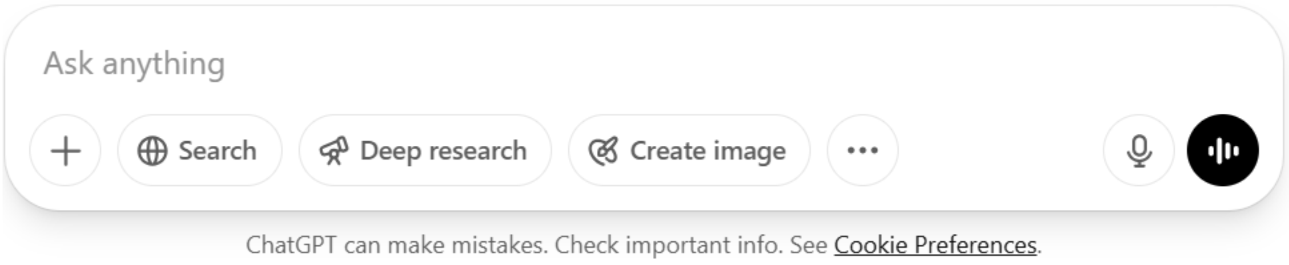
- What is “stored” in the model knowledge base is only what was contained in the pre-training data
- LLM “knows” better things mentioned frequently on the Internet
- Recent knowledge can be obtained with additional tools
- When switching topics, change to a new chat to erase the context window, tokens in the context window make the inference more expensive and slower, also “distracting”
- Be mindful of the model that you are using: smaller models, cheaper but less powerful
- reasoning/thinking models tuned with reinforcement learning: inference takes longer, suitable for more complex problems in math and code

How I use LLMs, Andrej Karpathy

https://www.youtube.com/watch?v=EWvNQjAaOHw

Tool Internet search

- When a model knows that it does not know, it searches the web, pull that information in the context window and uses it to answer
- Some models automatically detect when to use search, for some you have to state to use search
- Internet search + reasoning models= Deep research
  - typically, available on paid plans
  - model asks clarifying question
  - takes a couple of minutes
  - generates custom research paper
  - still no guarantees that there are no hallucinations

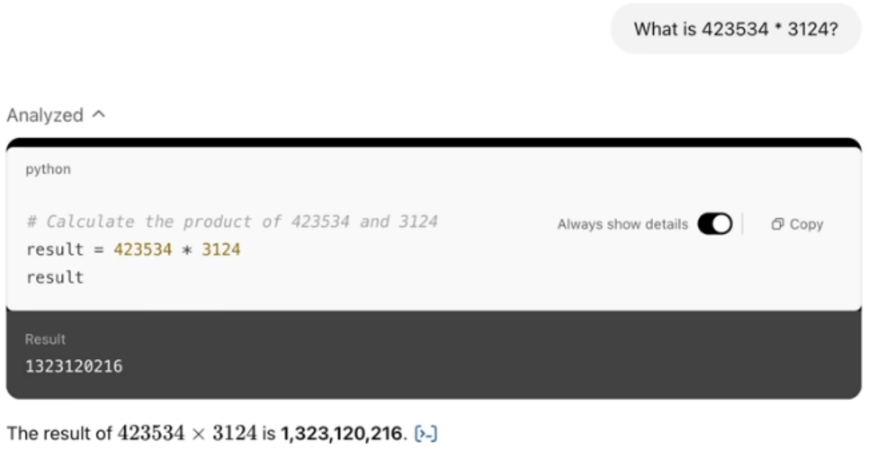


ChatGPT can make mistakes. Check important info. See Cookie Preferences.

How I use LLMs, Andrej Karpathy

https://www.youtube.com/watch?v=EWvNQjAaOHw

Different LLMs use different tools behind the scene



What is 423534 * 3124?

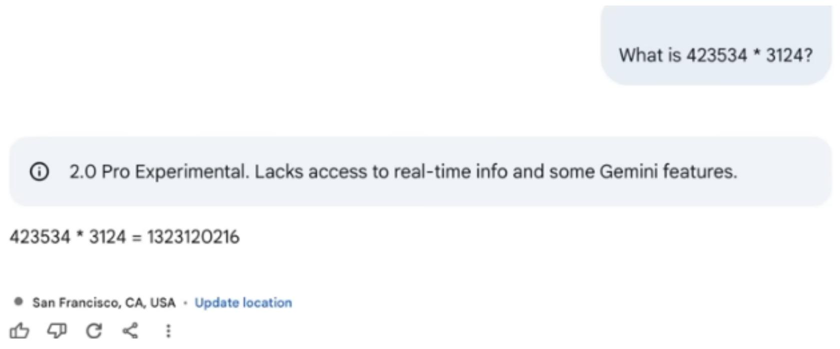
Analyzed

```python
# Calculate the product of 423534 and 3124
result = 423534 * 3124
result
```

Result
```
1323120216
```

The result of 423534 × 3124 is **1,323,120,216**.

ChatGPT writes python code to calculate



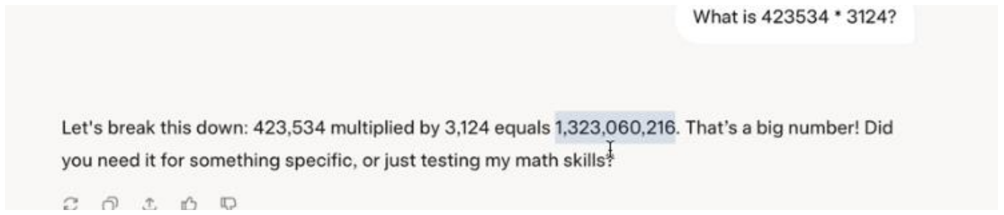
What is 423534 * 3124?

2.0 Pro Experimental. Lacks access to real-time info and some Gemini features.

423534 * 3124 = 1323120216

San Francisco, CA, USA · Update location

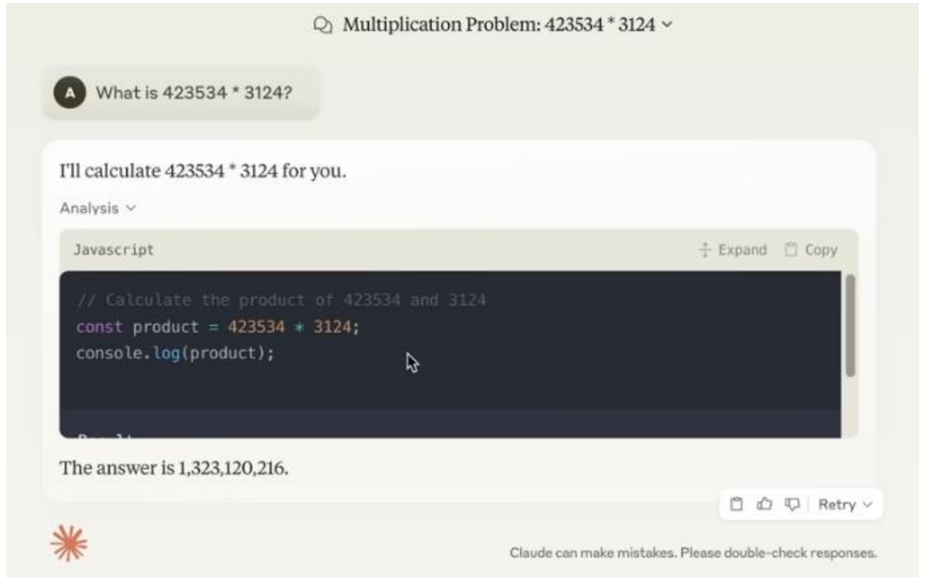
Gemini did not use any calculation tools, but knew this result



What is 423534 * 3124?

Let's break this down: 423,534 multiplied by 3,124 equals 1,323,060,216. That's a big number! Did you need it for something specific, or just testing my math skills?

Grok did not use any calculation tools, "remembers" the result wrongly



Multiplication Problem: 423534 * 3124

What is 423534 * 3124?

I'll calculate 423534 * 3124 for you.

Analysis

```javascript
// Calculate the product of 423534 and 3124
const product = 423534 * 3124;
console.log(product);
```

The answer is 1,323,120,216.

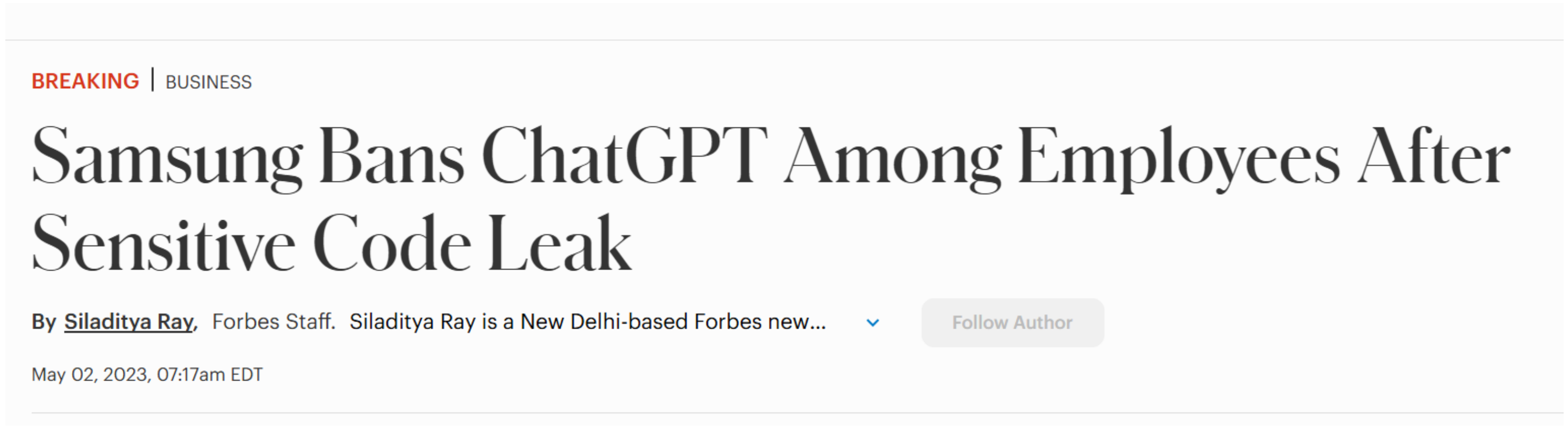
Claude can make mistakes. Please double-check responses.

Claude writes JS code to calculate

How I use LLMs, Andrej Karpathy
https://www.youtube.com/watch?v=EWvNQjAaOHw

Take care when using LLMs

- Avoid inputting personally identifiable information, trade secrets, or other sensitive data into public LLMs, as this data could potentially be stored, used for future training, or accessed by third parties.
- Evaluate and independently verify any factual claims, especially for important decisions. Do not treat LLM outputs as absolute truth.
- Review, edit, and validate all LLM outputs, especially those used for important tasks.
- Check the model's knowledge cut-off date. For current information, use LLMs integrated with search capabilities or consult up-to-date sources.
- Use LLMs as tools to augment human intelligence, not replace it entirely. Continue to develop and apply your own judgment and expertise.

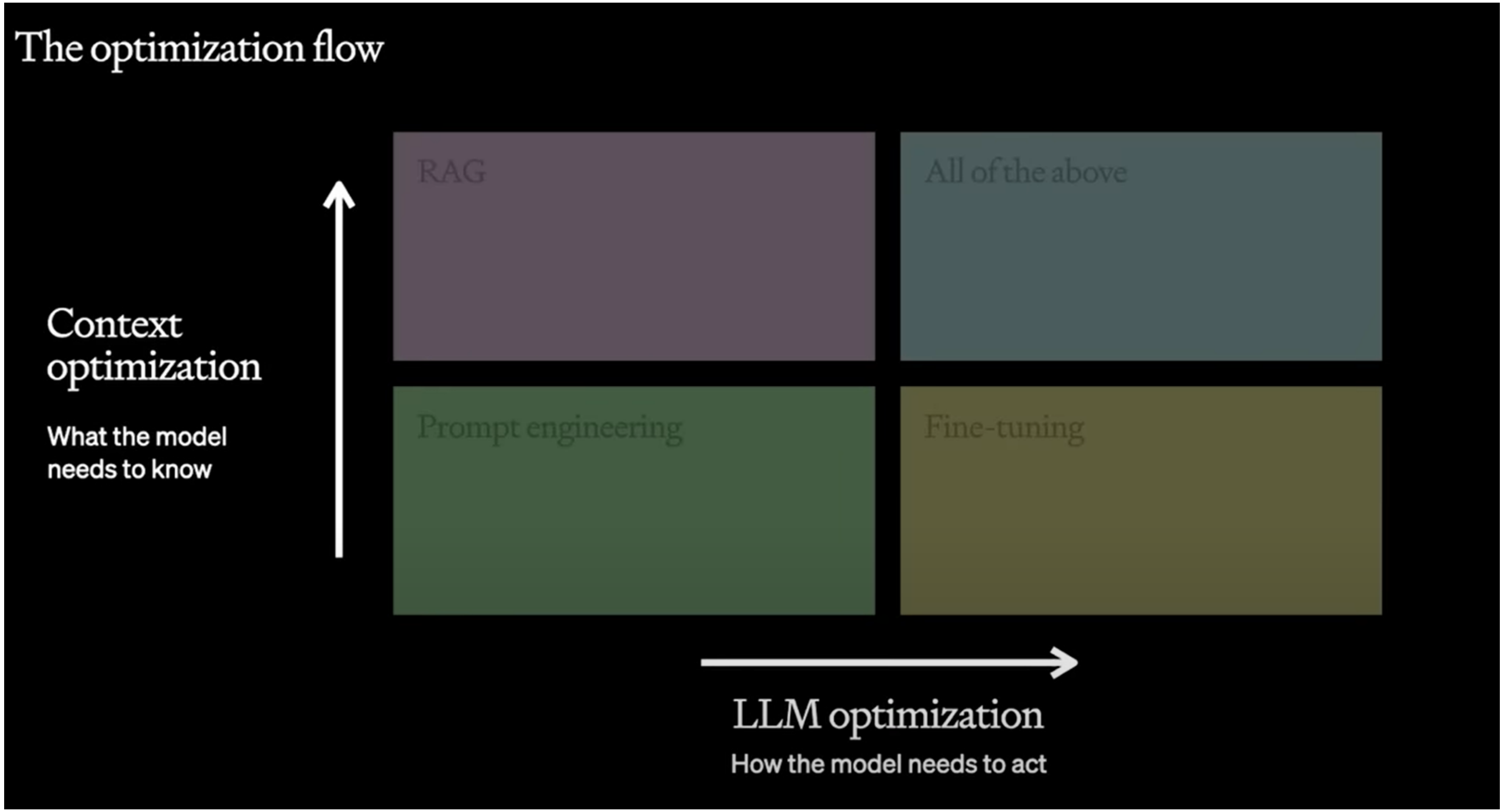

BREAKING | BUSINESS

Samsung Bans ChatGPT Among Employees After Sensitive Code Leak

By Siladitya Ray, Forbes Staff. Siladitya Ray is a New Delhi-based Forbes new...

May 02, 2023, 07:17am EDT

Techniques for maximizing LLM performance



https://www.youtube.com/watch?v=ahnGLM-RC1Y

Retrieval Augmented Generation (RAG)

- Technique for augmenting LLM knowledge with additional data: new data, private data.
- Helps with hallucinations, as the content is controlled.

RAG has two components:

- **Indexing done offline:**
  - ingest data from a source
  - break it into chunks
  - create embeddings of the chunks
  - store embedding in a vector store.

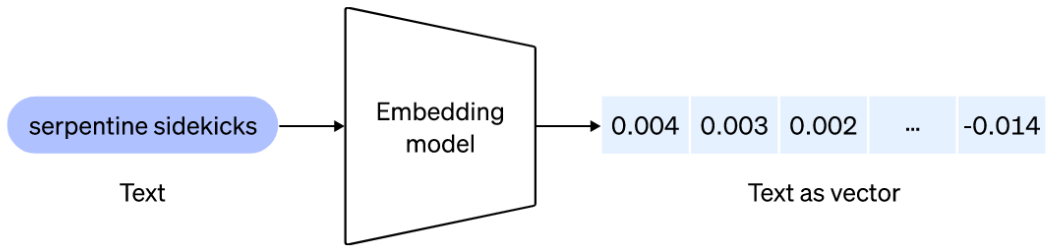
- **Retrieval and generation:**
  - take the user query at run time and create its embedding
  - retrieve the most relevant data from the source by comparing with the query embedding
  - concatenate this data with the original input prompt
  - send to the LLM fed to the text generator which produces the final output.

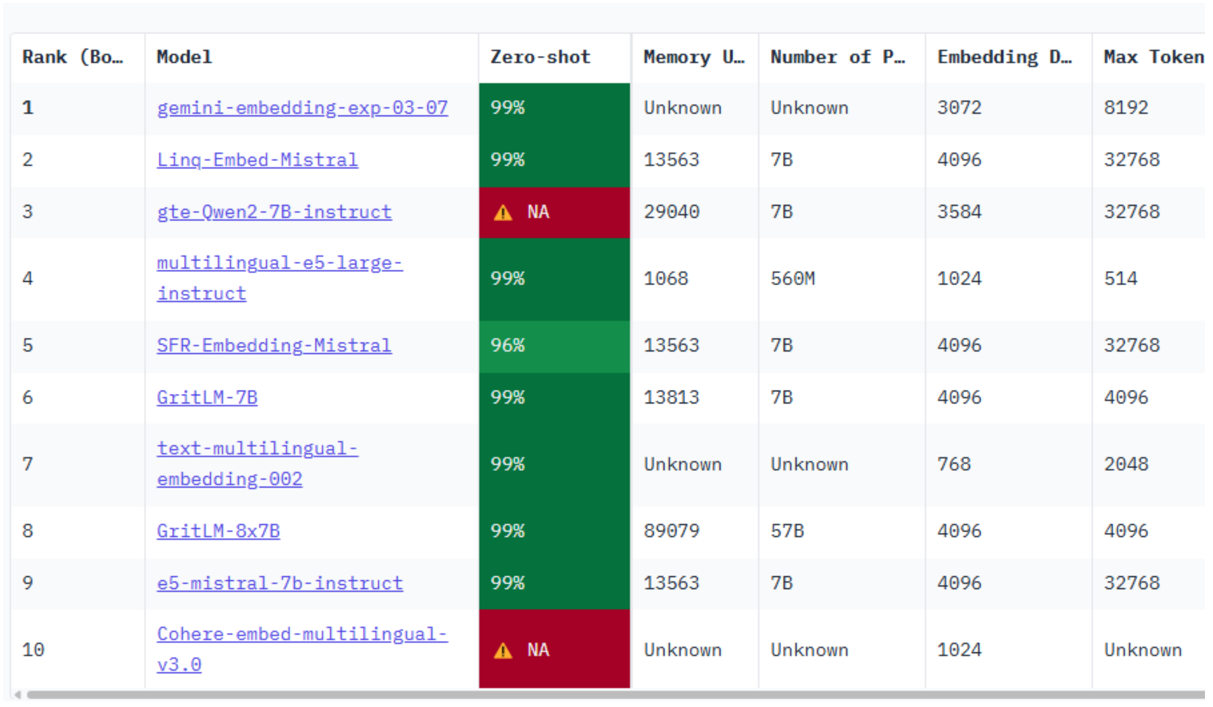
Sentence embedding

We can also represent sequences of words, or whole blocks of text as vectors.

Sentence Embeddings using Transformer models:

- Dense: Most values in the vector are non-zero.
- Fixed-Length: Uniform length, regardless of the sentence length
- Semantic richness: Similar meanings map to nearby vectors.
- Contextual: Meaning is derived from the surrounding words within the sentence.

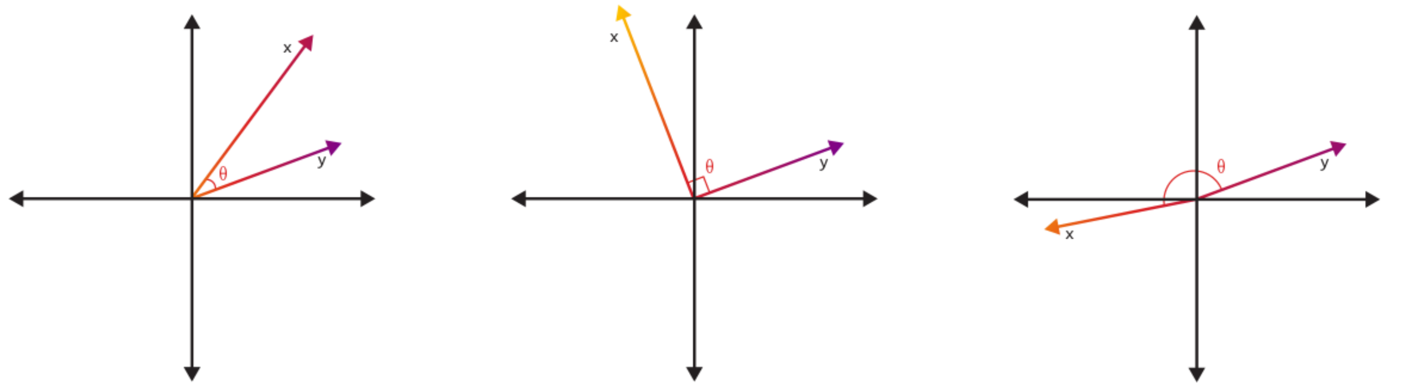




Rank (Bo…	Model	Zero-shot	Memory U…	Number of P…	Embedding D…	Max Token
1	gemini-embedding-exp-03-07	99%	Unknown	Unknown	3072	8192
2	Linq-Embed-Mistral	99%	13563	7B	4096	32768
3	gte-Qwen2-7B-instruct	⚠ NA	29040	7B	3584	32768
4	multilingual-e5-large-instruct	99%	1068	560M	1024	514
5	SFR-Embedding-Mistral	96%	13563	7B	4096	32768
6	GritLM-7B	99%	13813	7B	4096	4096
7	text-multilingual-embedding-002	99%	Unknown	Unknown	768	2048
8	GritLM-8x7B	99%	89079	57B	4096	4096
9	e5-mistral-7b-instruct	99%	13563	7B	4096	32768
10	Cohere-embed-multilingual-v3.0	⚠ NA	Unknown	Unknown	1024	Unknown

https://huggingface.co/spaces/mteb/leaderboard

Cosine similarity



Angle θ close to 0° Cos(θ) close to 1 → **Similar vectors**

Angle θ close to 90° Cos(θ) close to 0 → **Orthogonal vectors**

Angle θ close to 180° Cos(θ) close to -1 → **Opposite vectors**

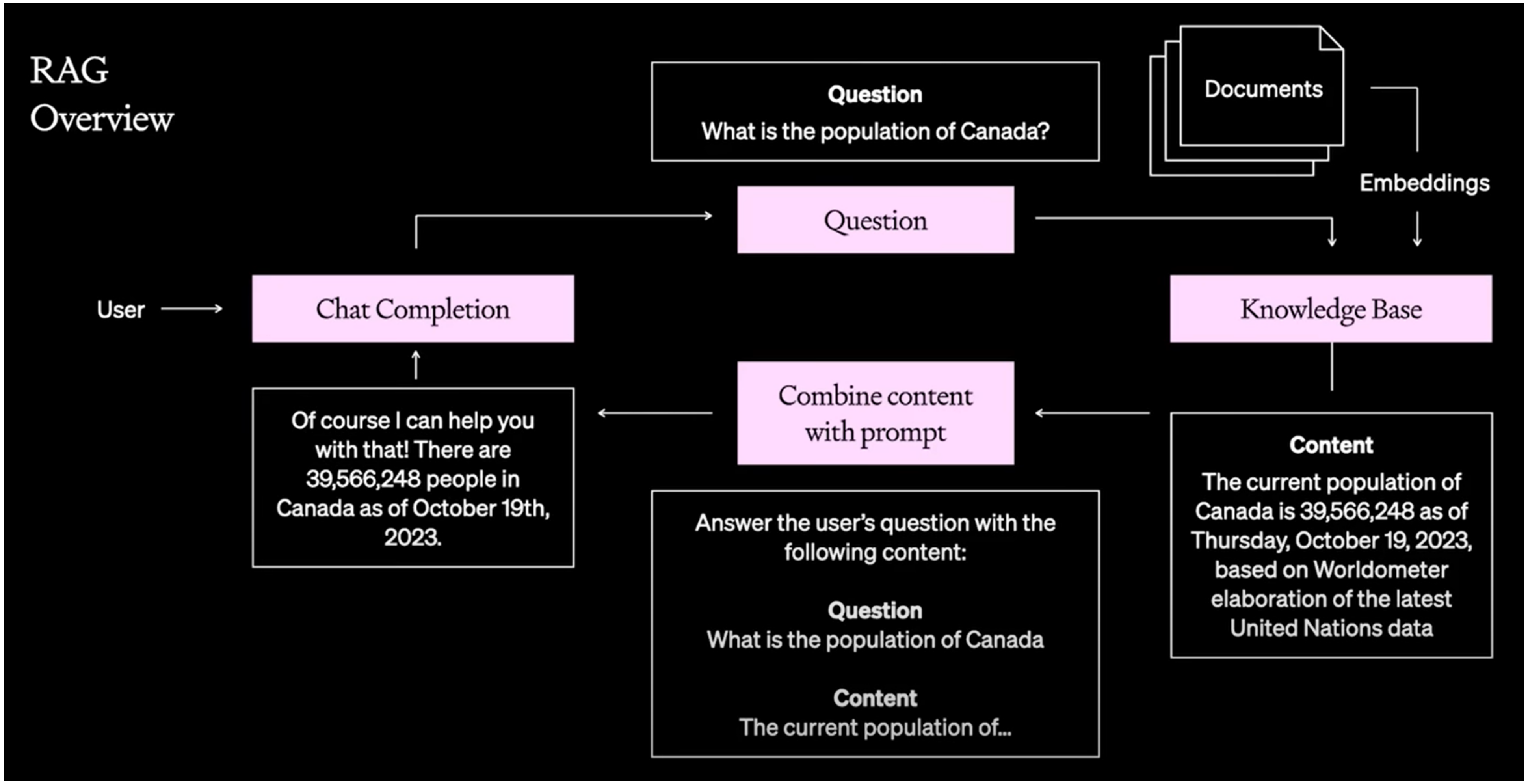
$$similarity(A,B) = cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

θ is the angle between the vectors,

$A \cdot B$ is dot product between A and B and calculated as
$A \cdot B = A^T B = \sum_{i=1}^{n} A_i B_i = A_1 B_1 + A_2 B_2 + \ldots + A_n B_n$,

$\|A\|$ represents the L2 norm or magnitude of the vector which is calculated as
$\|A\| = \sqrt{A_1^2 + A_1^2 \ldots A_1^n}$.

Retrieval Augmented Generation (RAG)

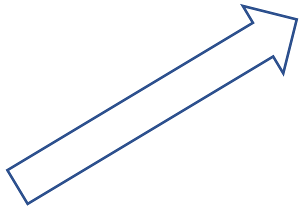

RAG
Overview
Question
What is the population of Canada?
Documents
Embeddings
Question
User
Chat Completion
Knowledge Base
Combine content
with prompt
Of course I can help you with that! There are 39,566,248 people in Canada as of October 19th, 2023.
Answer the user's question with the following content:
Question
What is the population of Canada
Content
The current population of...
Content
The current population of Canada is 39,566,248 as of Thursday, October 19, 2023, based on Worldometer elaboration of the latest United Nations data

https://www.youtube.com/watch?v=ahnGLM-RC1Y

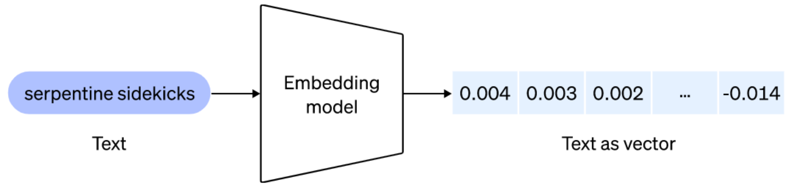
Step 1: Indexing

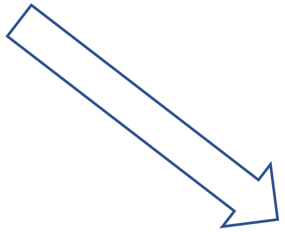
Documents

Break the text into *smaller* chunks, parameters:
chunk size,
chunk overlap

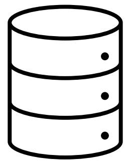


Send chunks of text

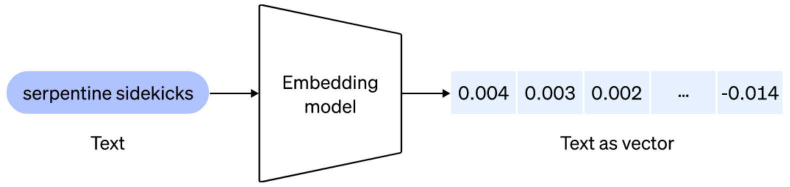




Store embeddings in embeddings data base (vector store)
(each chunk of text=embedding vector)



Steps 2: Retrieval



User enters a question

Question is converted to an embedding vector

Embedding of the question is compared to all the embeddings we have stored.
Top k Most similar embeddings are selected.

Question + Most similar text is sent to an LLM

Answer is shown to the user.

ADDRESSING COMMON ISSUES

Missing Values & Outliers

- **Investigate First: Don't remove rows with missing data or outliers blindly.**

- Missing value
  - why are they missing?
  - is there a pattern?
  - small Percentages add up: Removing rows with any missing value can significantly reduce your dataset if missing are spread across many columns.
  - in practice: once the model is used in practice, if your pipeline cannot handle missing data, any data point with a missing value cannot be scored
  - **leakage alert:** Impute *after* splitting data and use statistics *only* from the training set to fit your imputer. Use pipelines to handle this within cross-validation.

- Outliers
  - is it a data entry error, a measurement issue, or a true extreme value
  - outliers can represent genuine, important, and insightful data points
  - consider capping extreme values for models sensitive to outliers. Tree based models are robust to outliers

Encoding high cardinality features

- Features with very many unique values (approaching the number of samples) often act like identifiers and can lead to overfitting
- For features that are essentially unique IDs or offer no predictive value (names) after investigation, removal is justified. Always document your reasoning
- **Why one-hot encoding (dummy variables) is often a bad Idea for high-cardinality:**
  - Curse of Dimensionality: If a feature has hundreds or thousands of unique values (e.g., "ZIP Code“, "Product Name"), one-hot encoding will create an equal number of new binary (0/1) features.
  - This massively increases the number of columns (dimensionality) in the dataset.
  - Sparsity: Most of these new dummy features will be zero for any given sample
  - Increased Computational Cost: Training models on datasets with very high dimensions is computationally expensive and slow.
  - Overfitting Risk: With many features, models are more likely to find spurious correlations in the training data that don't generalize to unseen data. The model might memorize the training set instead of learning underlying patterns.
- Encoding Strategies:
  - grouping: Combine infrequent categories into an "Other" category.
  - target encoding: (Use with extreme caution within CV to prevent leakage).
  - domain knowledge: Can you derive a more meaningful, lower-cardinality feature?

When it is ok and not ok to remove a feature

- When it is ok to remove a feature
  - if we have features like user_id, customer_id, transaction_id, or serial numbers that are unique to each row. These usually offer no generalizable patterns for a model to learn
  - if a feature has almost the same value for nearly all data points, it provides no information to distinguish between samples
  - if we have very strong domain expertise that tells us a feature cannot logically influence the outcome we're trying to predict, it might be a candidate. However, careful with assumptions – sometimes data reveals unexpected relationships.
  - if a feature contains information that would *not* be available at the time of prediction in a real-world scenario
  - If we have two features that are perfectly or very highly correlated (e.g., “amount in euros" and “amount in dollars"), they are providing largely the same information

- When it is NOT ok to remove a feature
  - we think a feature is not that interesting (without strong domain knowledge), just by looking at the plots
  - a feature has some missing values

Train validation test split

- Train set: data from which the model learns
- Validation set: data on which to tune your model and make decisions about hyperparameters and model selection without "contaminating" the final test set.
- Test set: provides the final, unbiased evaluation of the fully trained and tuned model's performance. It simulates how the model would perform on new, unseen real-world data.

- Cross validation offers multiple validation sets: instead of a single train/validation split, data is divided in k folds, and the model is trained in iteration 1 on folds 1,2,34 and validated on fold 2, on iteration 1: trained on folds 1,2,3,5 and validated on fold 4 ...and so on, until each fold has served as the validation set once.
- If all you training and decision making (model selection, hyperparameter selection) is done within cross-validation, no need for additional validation set

Hyperparameter tuning

- Do not assume all default values are optimal for each dataset
- Understand what do parameters do and what range of parameter values make sense to explore (should you really waste computational resource testing random forest with trees of depth 1 or 2?)
- Do not define a large search space (high number of parameter combinations) and then do grid search --- takes too long, or use randomized search, but test only a couple of combinations (n_iter parameter)
- Do not do data leakage with CV: if your hyperparameter tuning involves a pipeline with preprocessing (e.g., scaling, imputation), **ensure these steps are fitted within each fold of the cross-validation**, not on the whole dataset
- Define a clear optimization metric based on the business problem
- Check the results of the top scoring models in cross validation (cv_results_), you can pass multiple metrics in the scoring parameter: examine the results, a model with a slightly lower score might be significantly simpler