Retrieval Augmented Generation (RAG)

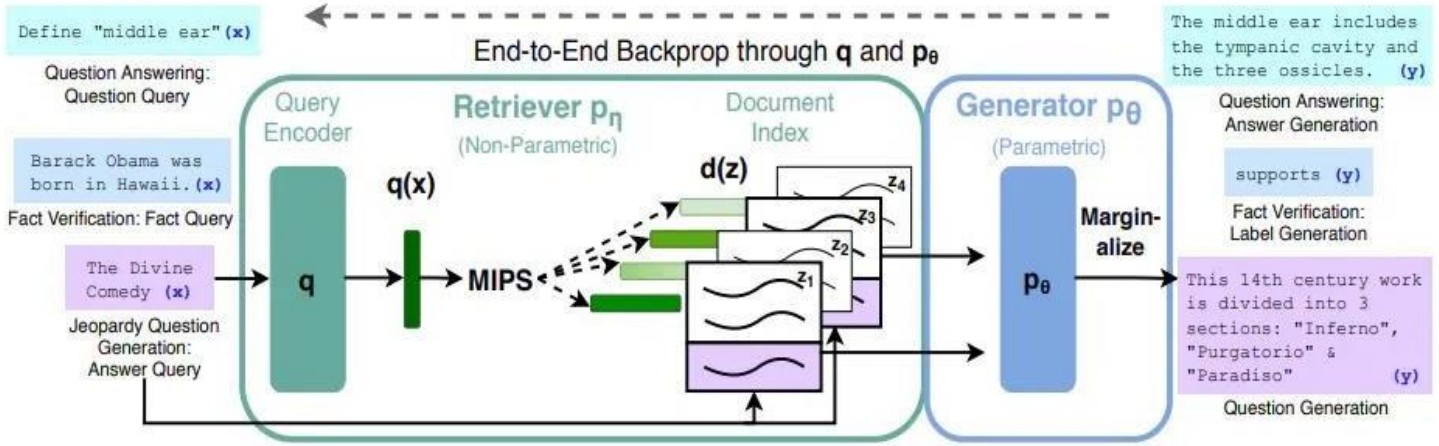


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder* + *Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

From Language Models to Assistants

1. **Zero-Shot (ZS) and Few-Shot (FS) In-Context Learning**

2. **Instruction finetuning**

3. **Reinforcement Learning from Human Feedback (RLHF)**

QA with Prompt Engineering

Instructions

Context

Question

Answer

You're an assistant trained to answer questions using the given context.

Context:
"The engine powering Grok is Grok-1, our frontier LLM, which we developed over the last four months. Grok-1 has gone through many iterations over this span of time.
After announcing xAI, we trained a prototype LLM (Grok-0) with 33 billion parameters. This early model approaches LLaMA 2 (70B) capabilities on standard LM benchmarks but uses only half of its training resources. In the last two months, we have made significant improvements in reasoning and coding capabilities leading up to Grok-1, a state-of-the-art language model that is significantly more powerful, achieving 63.2% on the HumanEval coding task and 73% on MMLU.
To understand the capability improvements we made with Grok-1, we have conducted a series of evaluations using a few standard machine learning benchmarks designed to measure math and reasoning abilities.

GSM8k: Middle school math word problems, (Cobbe et al. 2021), using the chain-of-thought prompt.
MMLU: Multidisciplinary multiple choice questions, (Hendrycks et al. 2021), provided 5-shot in-context examples.
HumanEval: Python code completion task, (Chen et al. 2021), zero-shot evaluated for pass@1.
MATH: Middle school and high school mathematics problems written in LaTeX, (Hendrycks et al. 2021), prompted with a fixed 4-shot prompt."

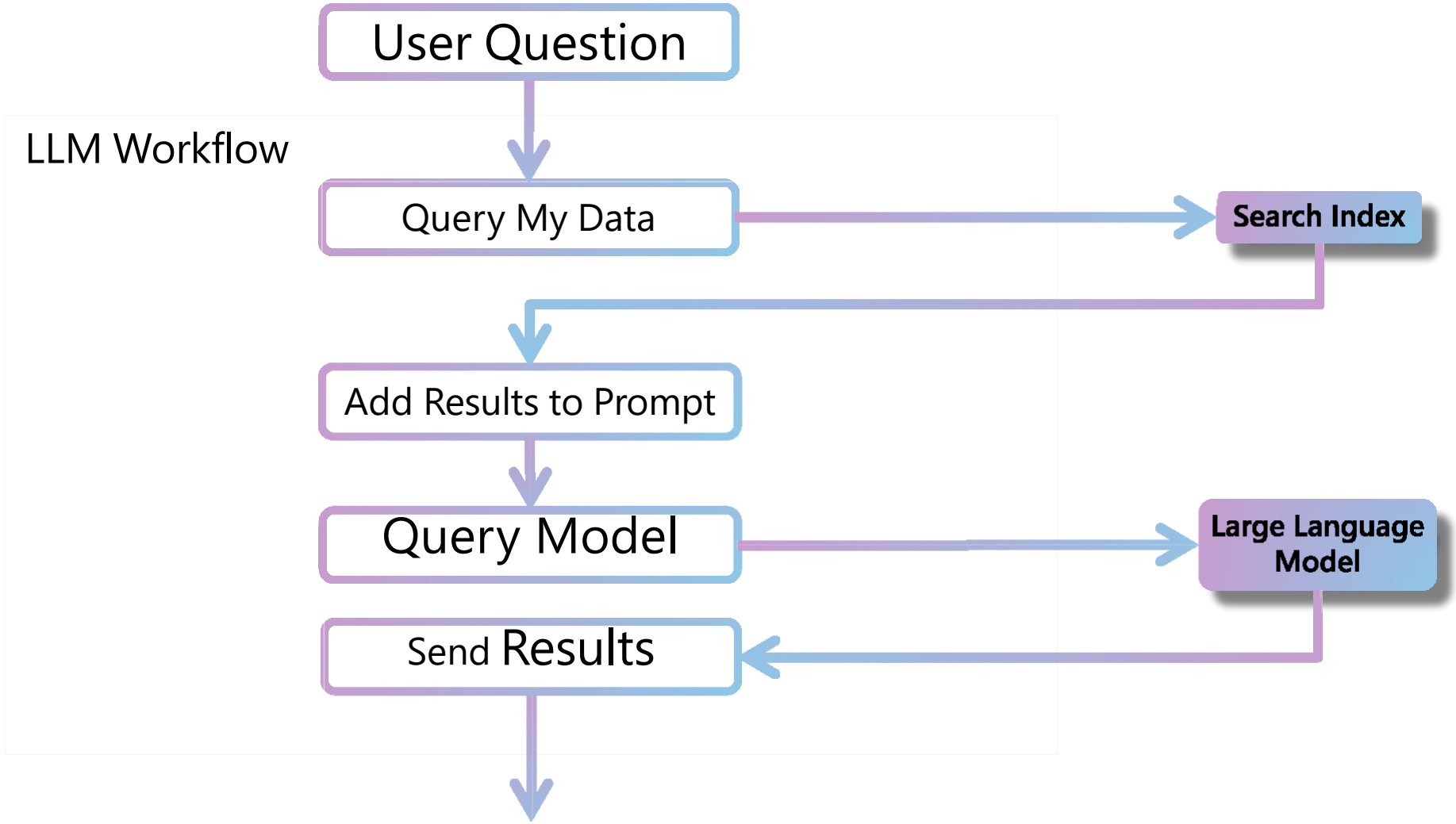
Answer the following question: "How many parameters are there in Grok-0?"

Grok-0, the prototype LLM mentioned in the provided context, is stated to have been trained with 33 billion parameters.

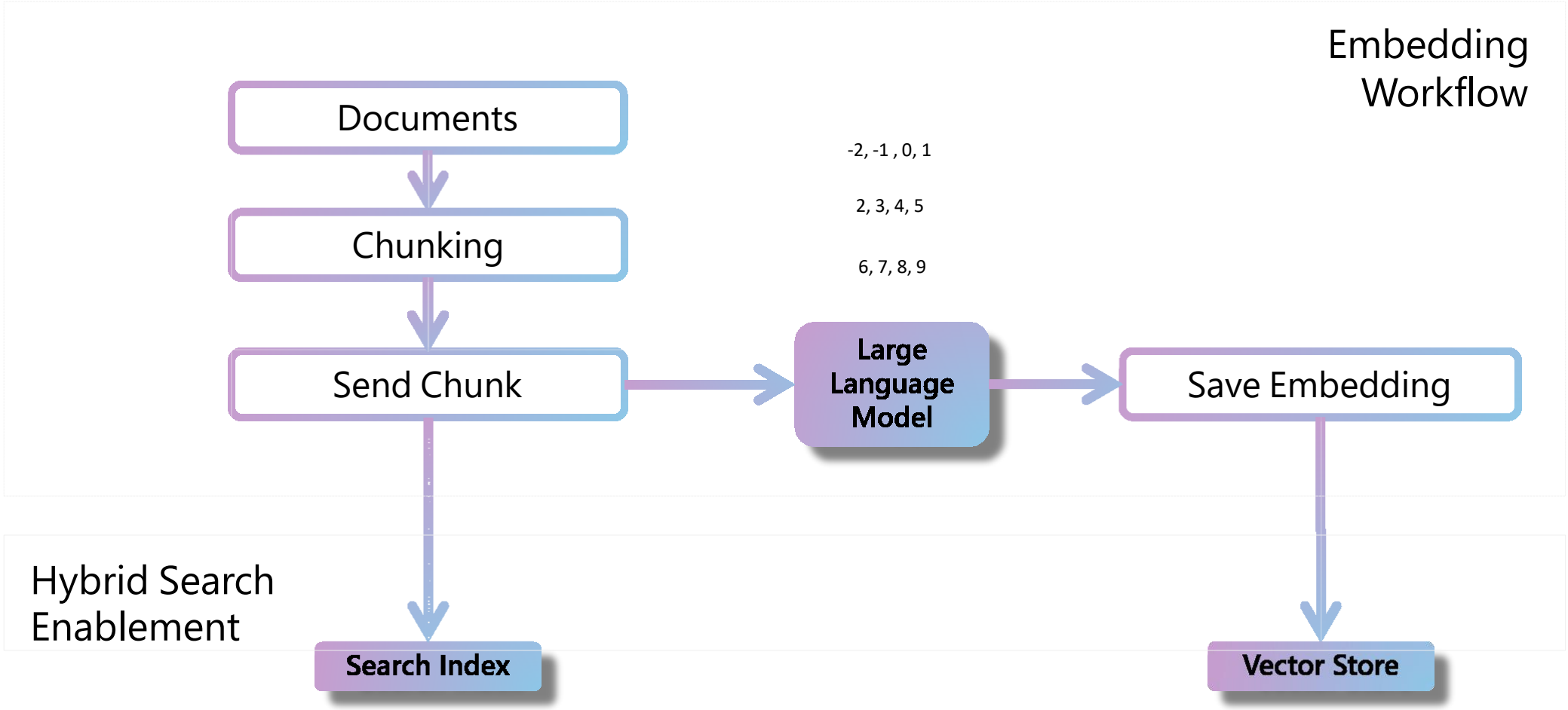
Prompt

What is RAG?

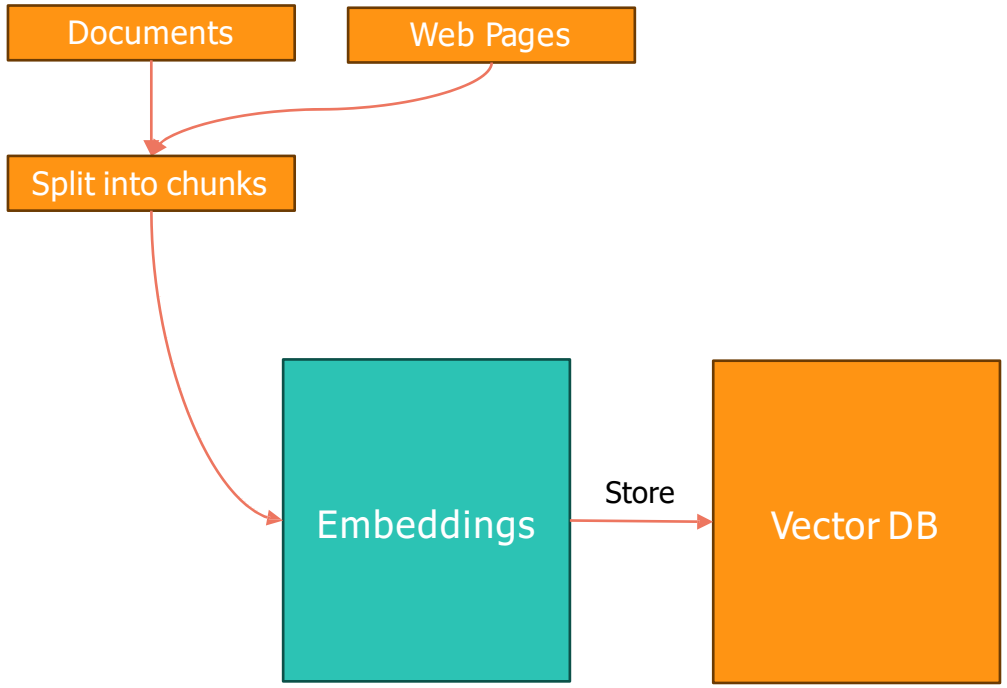
What is RAG?


User Question
LLM Workflow
Query My Data
Search Index
Add Results to Prompt
Query Model
Large Language Model
Send Results

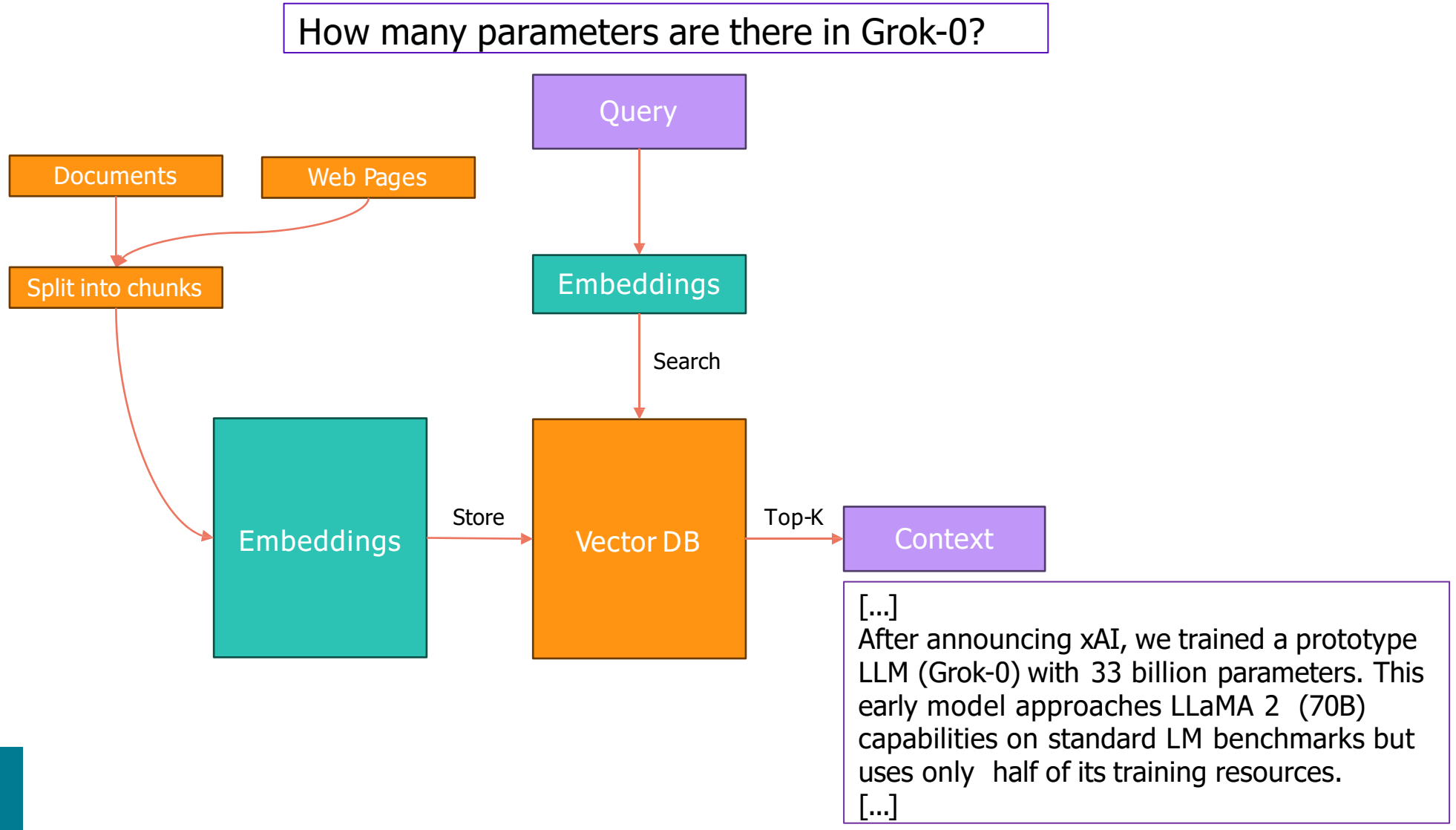
What is RAG?


Embedding Workflow
Documents
Chunking
Send Chunk
-2, -1 , 0, 1
2, 3, 4, 5
6, 7, 8, 9
Large Language Model
Save Embedding
Hybrid Search Enablement
Search Index
Vector Store

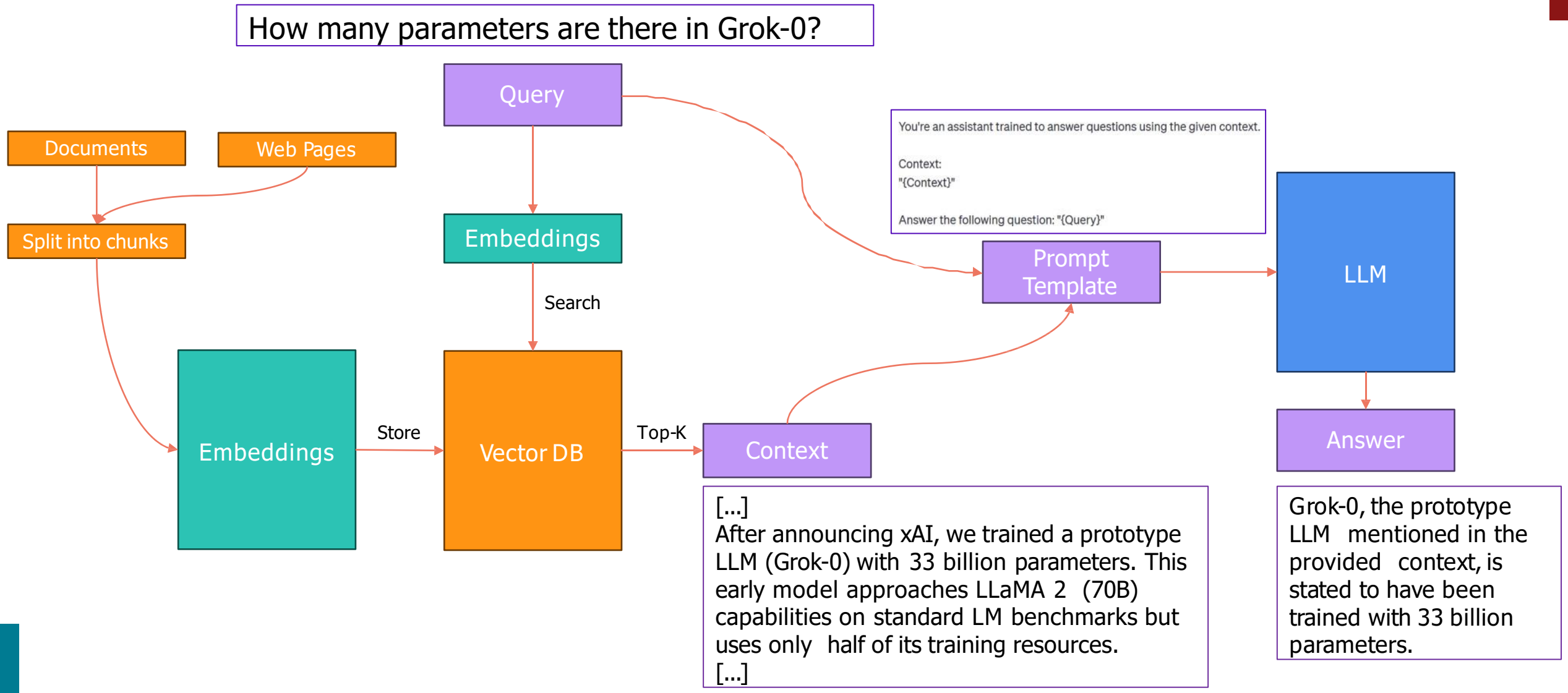
QA with Retrieval Augmented Generation



QA with Retrieval Augmented Generation

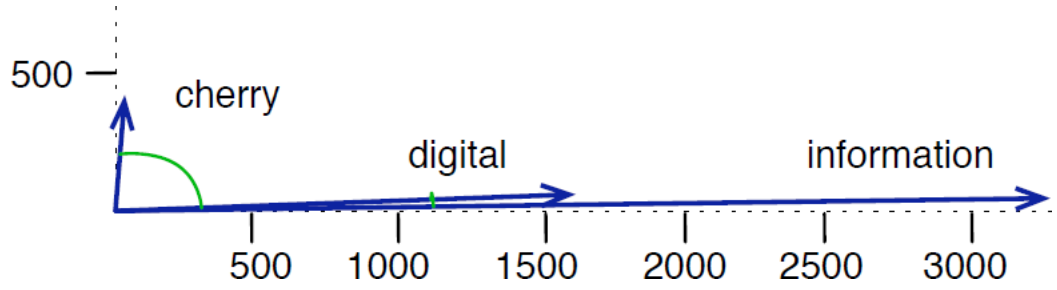


QA with Retrieval Augmented Generation



Why do we use vectors to represent words?

Given the words "**cherry**", "**digital**" and "**information**", if we represent the embedding vectors using only 2 dimensions (X, Y) and we plot them, we hope to see something like this: the angle between words with similar meaning is small, while the angle between words with different meaning is big. So, the embeddings "capture" the meaning of the words they represent by projecting them into a high-dimensional space.



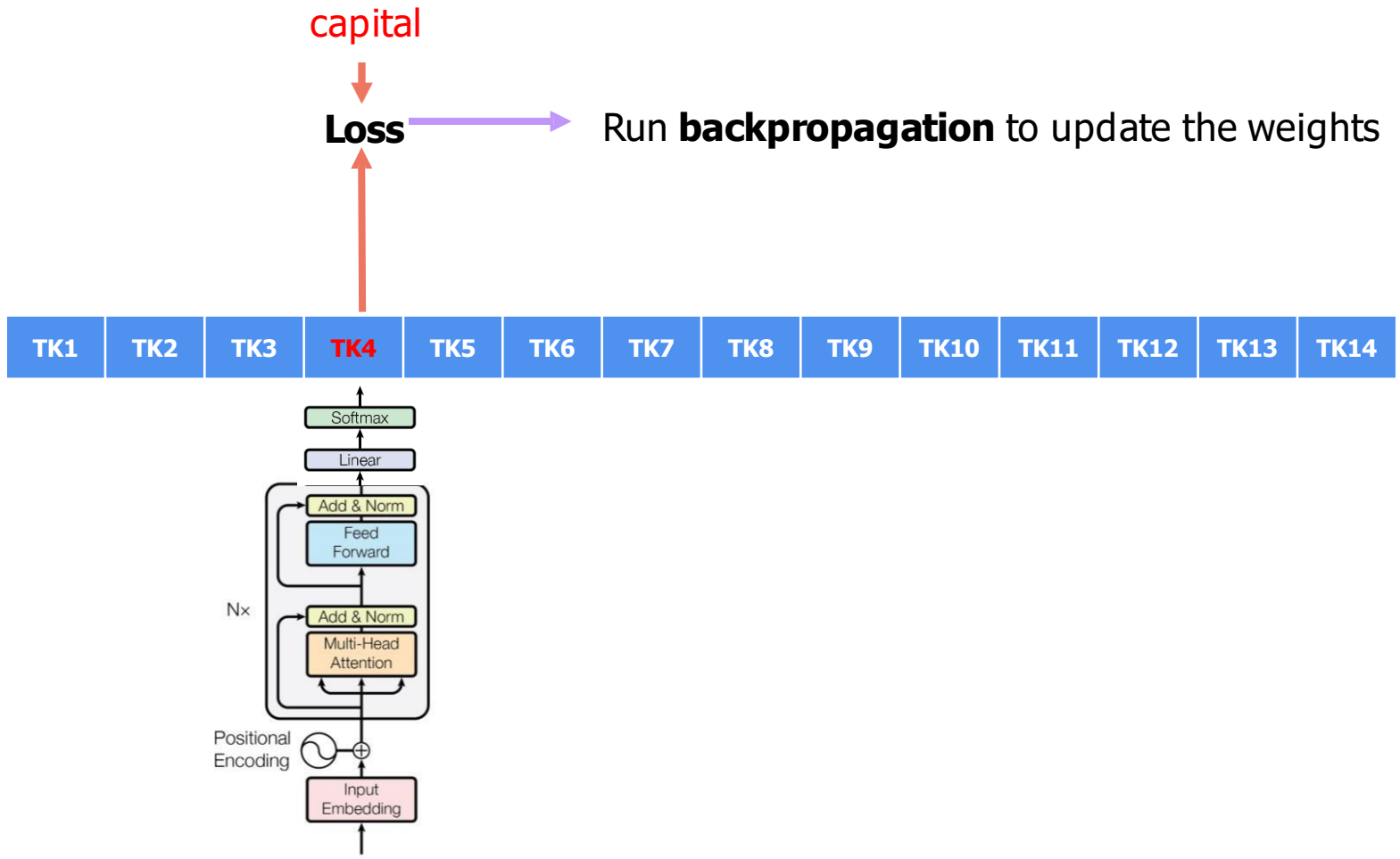
We commonly use the **cosine similarity**, which is based on the **dot product** between the two vectors.

Word embeddings: the ideas

- Words that are synonyms tend to occur in the same context (surrounded by the same words).
  - For example, the word "*teacher*" and "*professor*" usually occur surrounded by the words "*school*", "*university*", "*exam*", "*lecture*", "*course*", etc..
- The inverse can also be true: words that occur in the same context tend to have similar meanings. This is known as the **distributional hypothesis**.
- This means that to capture the meaning of the word, we also need to have access to its context (the words surrounding it).
- This is why we employ the Self-Attention mechanism in the Transformer model to capture contextual information for every token. The Self-Attention mechanism relates every token to all the other tokens in the sentence.

Embedding vectors in BERT

Target (1 token):



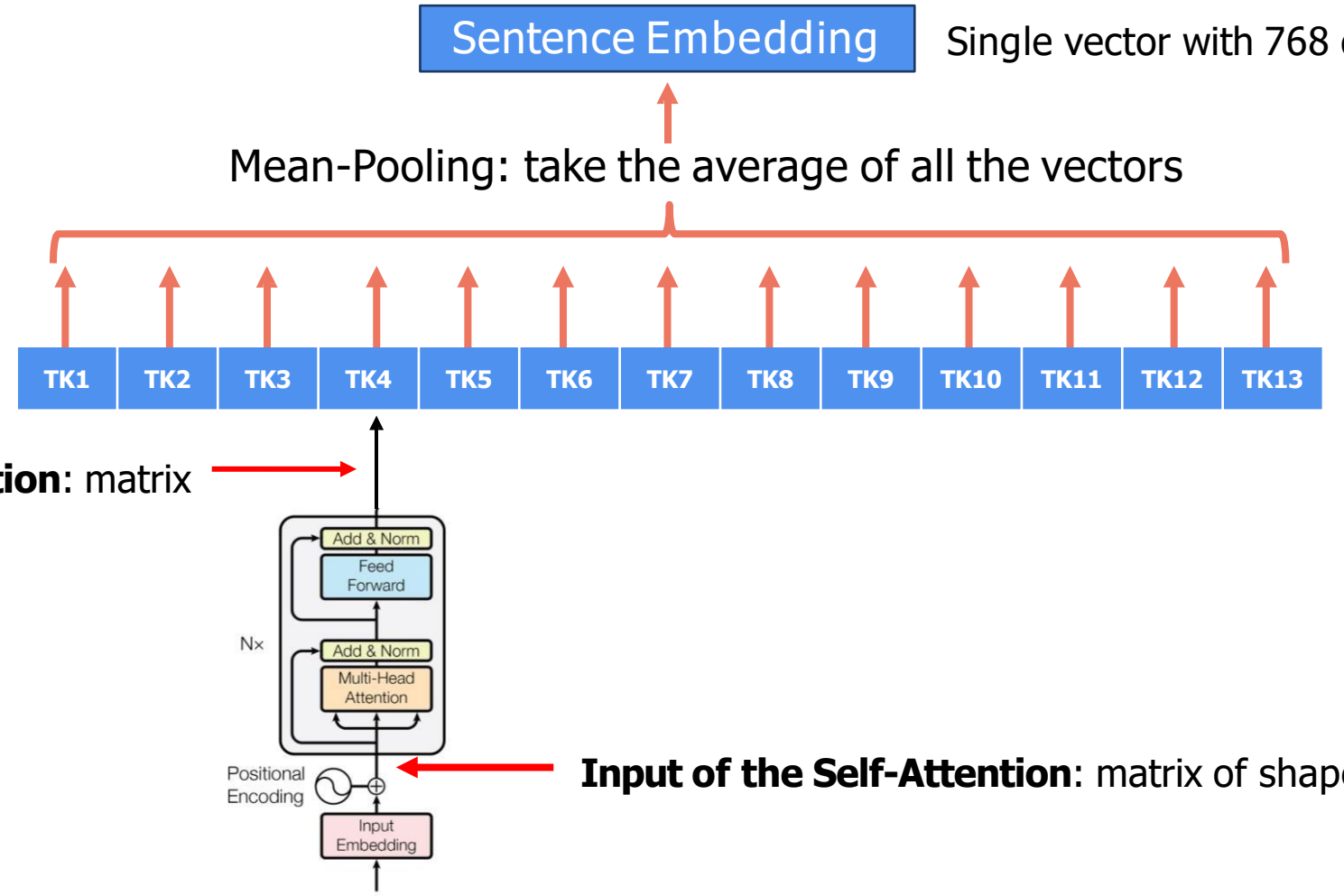
Input (14 tokens):

Rome is the [mask] of Italy, which is why it hosts many government buildings.

Sentence Embeddings

- We can use the Self-Attention mechanism also to capture the "meaning" of an entire sentence.
- We can use a pre-trained BERT model to produce embeddings of entire sentences. Let's see how

Sentence Embeddings with BERT



Sentence Embeddings: comparison

- How can we compare Sentence Embeddings to see if two sentences have similar "meaning"? We could use the cosine similarity, which measures the cosine of the angle between the two vectors. A small angle results in a high cosine similarity score.

$$\text{cosine similarity} = S_C(A, B) := \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \cdot \sqrt{\sum_{i=1}^{n} B_i^2}}$$

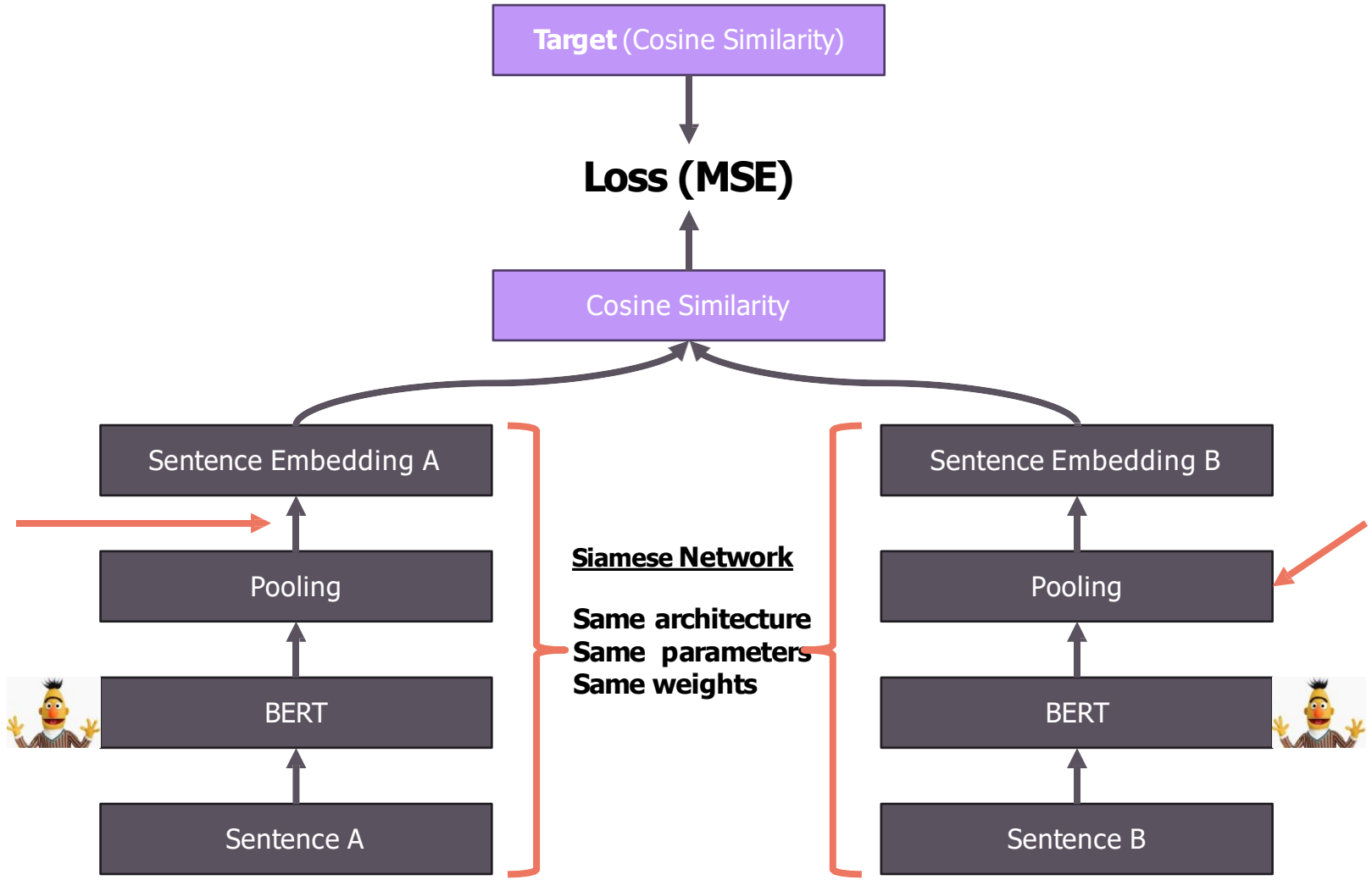
- **But there's a problem**: nobody told BERT that the embeddings it produces should be comparable with the cosine similarity, that is, two similar sentences should be represented by vectors pointing to the same direction in space. How can we teach BERT to produce embeddings that can be compared with a similarity function of our choice?

Sentence BERT

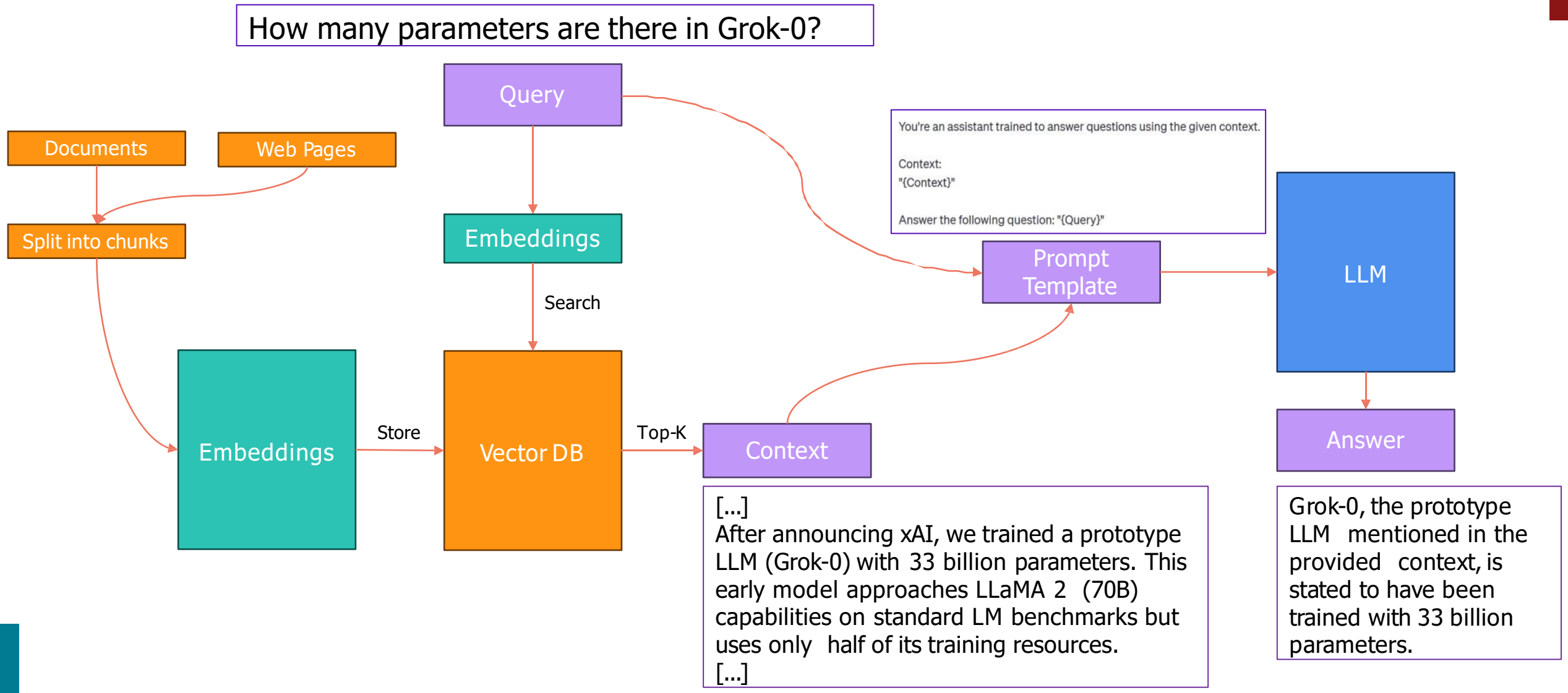
Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks

Nils Reimers and Iryna Gurevych
Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
www.ukp.tu-darmstadt.de

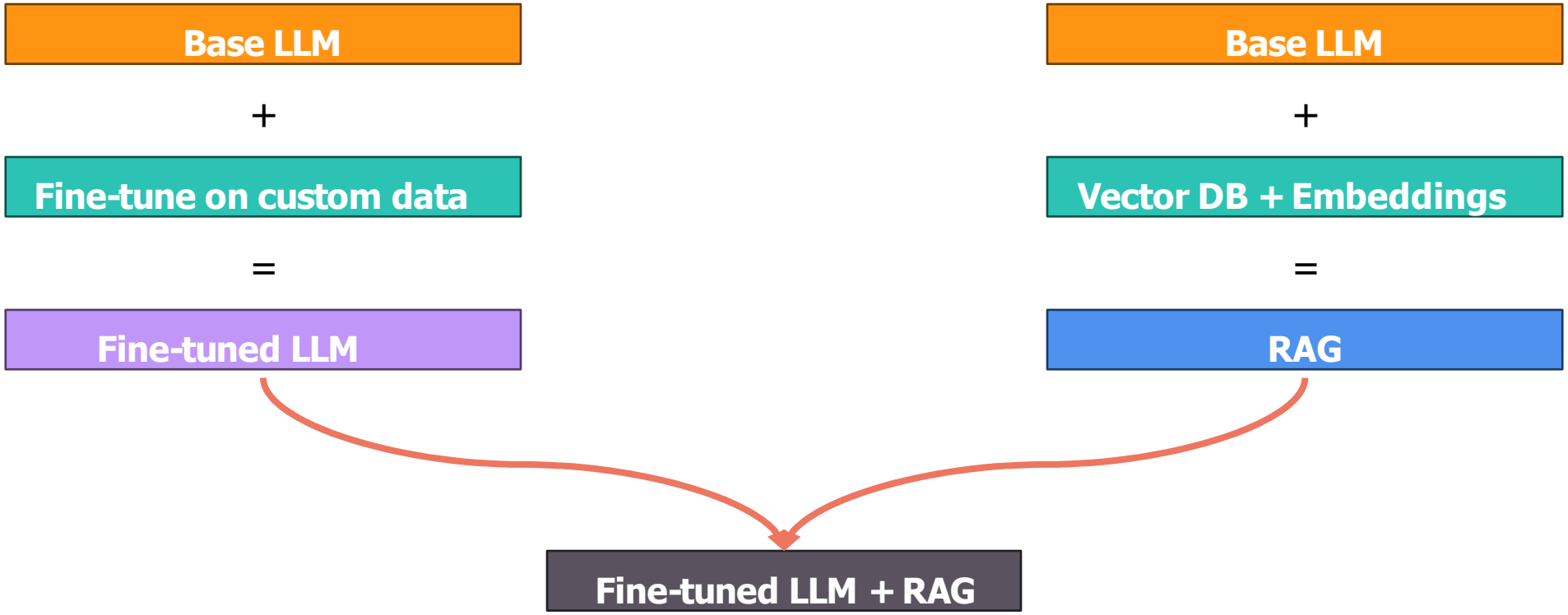
Sentence BERT: architecture



QA with Retrieval Augmented Generation



Strategies to teach new concepts to LLM



Umar Jamil – https://github.com/hkproj/retrieval-augmented-generation-notes