Retrieval-Augmented Generation (RAG)

Outline

- Intro to Large Language Models
- RAG pipeline
- Embedding vectors
  - Sentence BERT
- Vector DB
  - Algorithms (HNSW)

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.

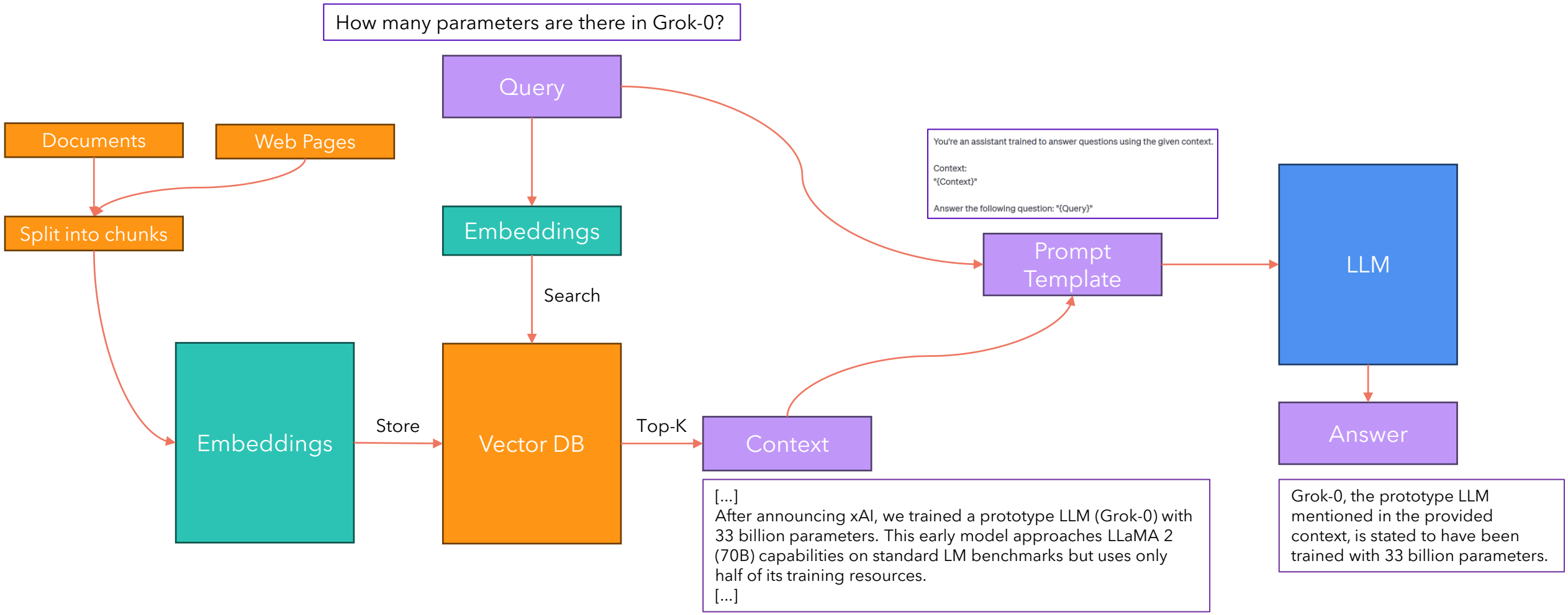
GSM8k: Middle school math word problems, (Cobbe et al. 2021), using the chain-of-thought prompt.
MMLU: Multidisciplinary multiple choice questions, (Hendrycks et al. 2021), provided 5-shot in-context examples.
HumanEval: Python code completion task, (Chen et al. 2021), zero-shot evaluated for pass@1.
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?"

Prompt

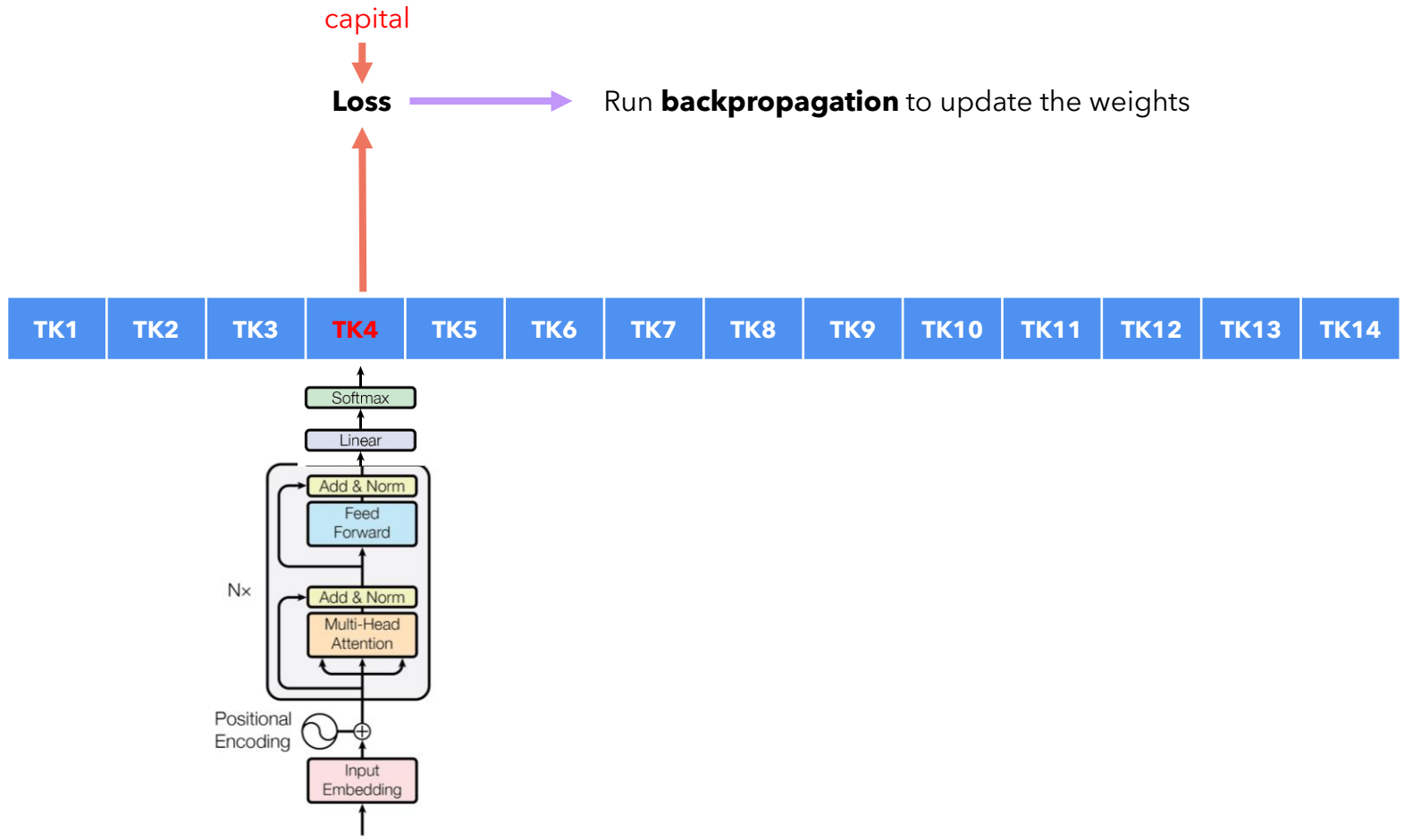
Grok-0, the prototype LLM mentioned in the provided context, is stated to have been trained with 33 billion parameters.

QA with Retrieval Augmented Generation



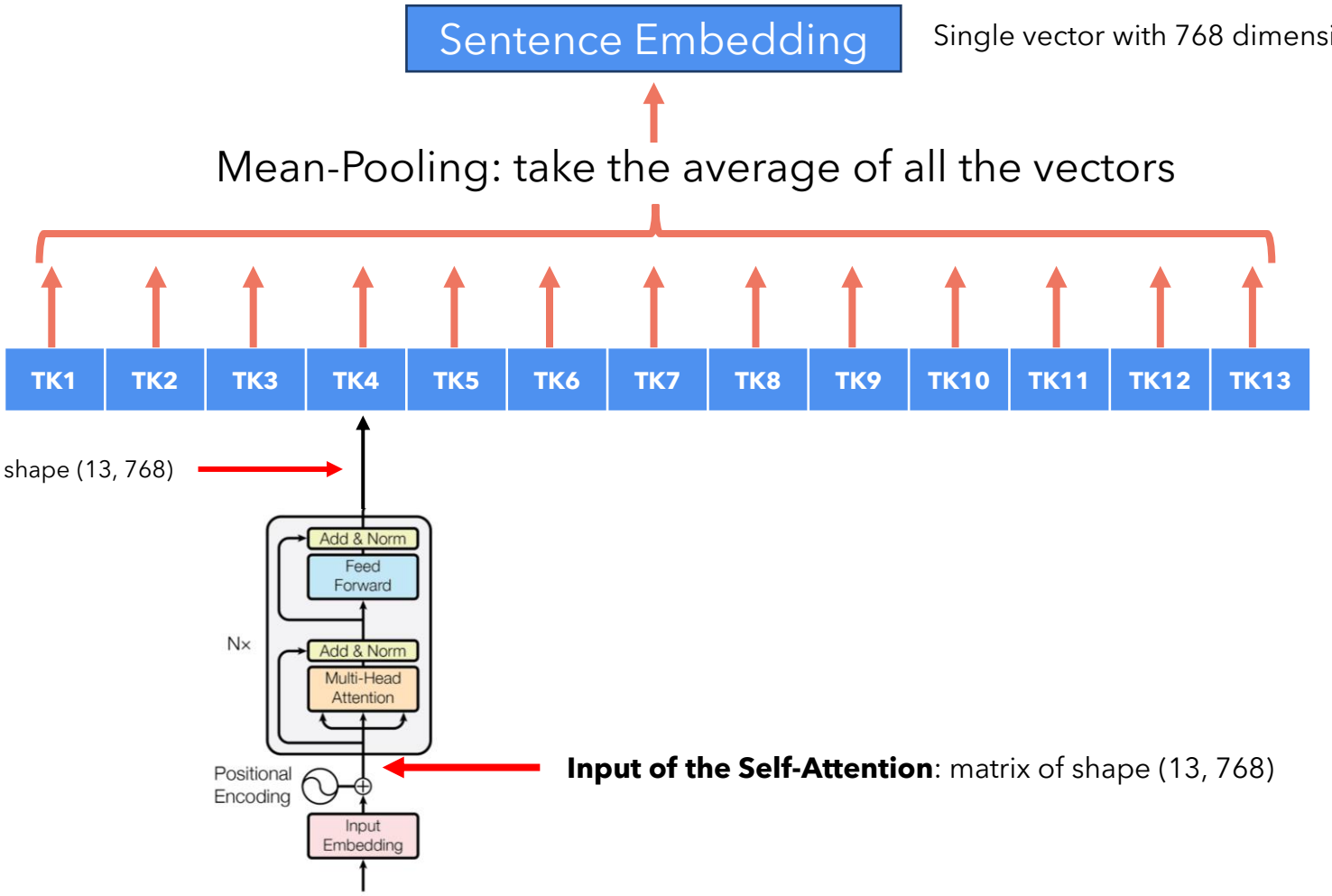
How do we train embedding vectors in BERT?

Target (1 token):



Input (14 tokens): Rome is the [mask] of Italy, which is why it hosts many government buildings.

Sentence Embeddings with BERT



Output (13 tokens):

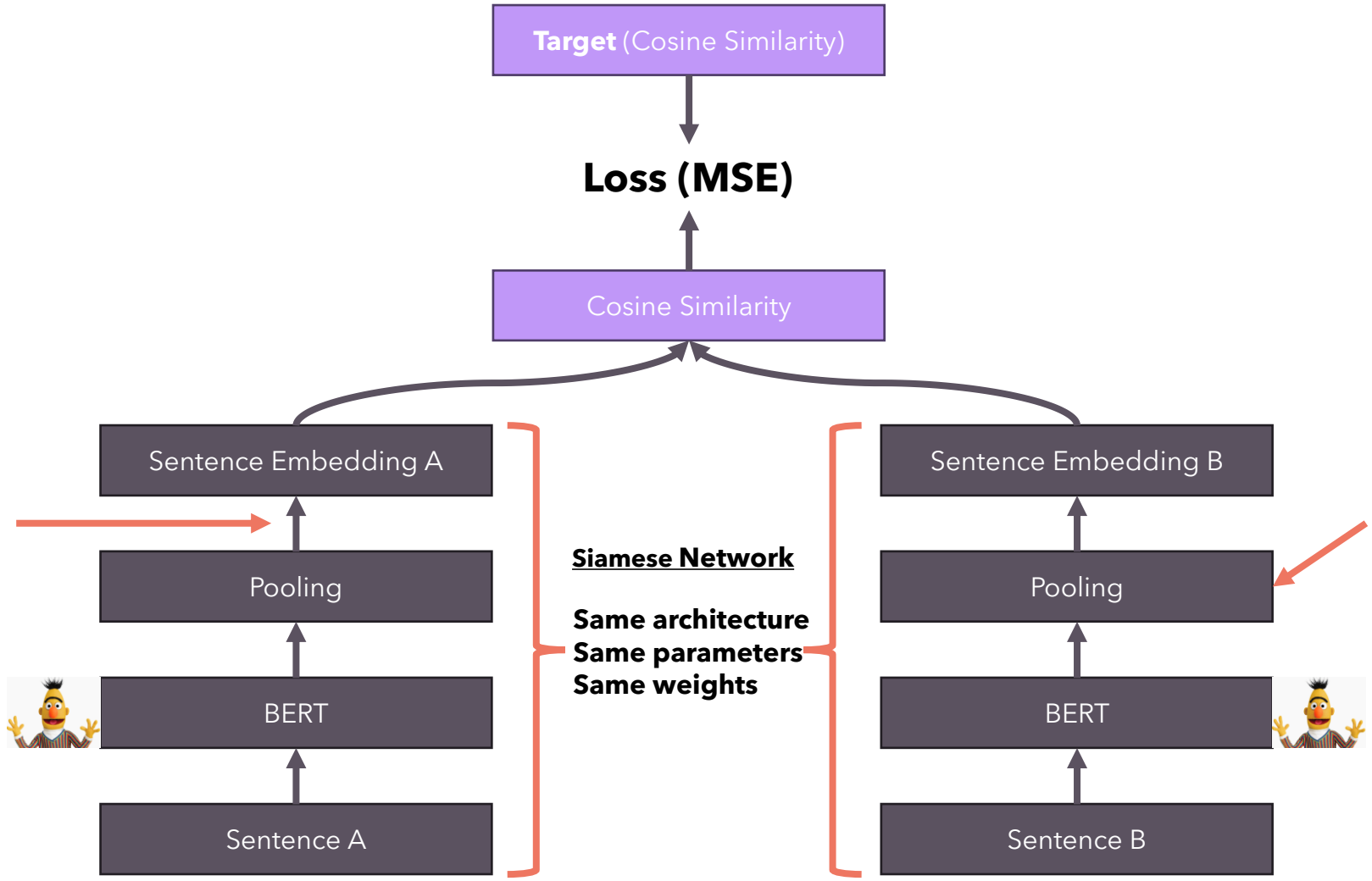
Input (13 tokens): Our professor always gives us lots of assignments to do in the weekend.

Introducing 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



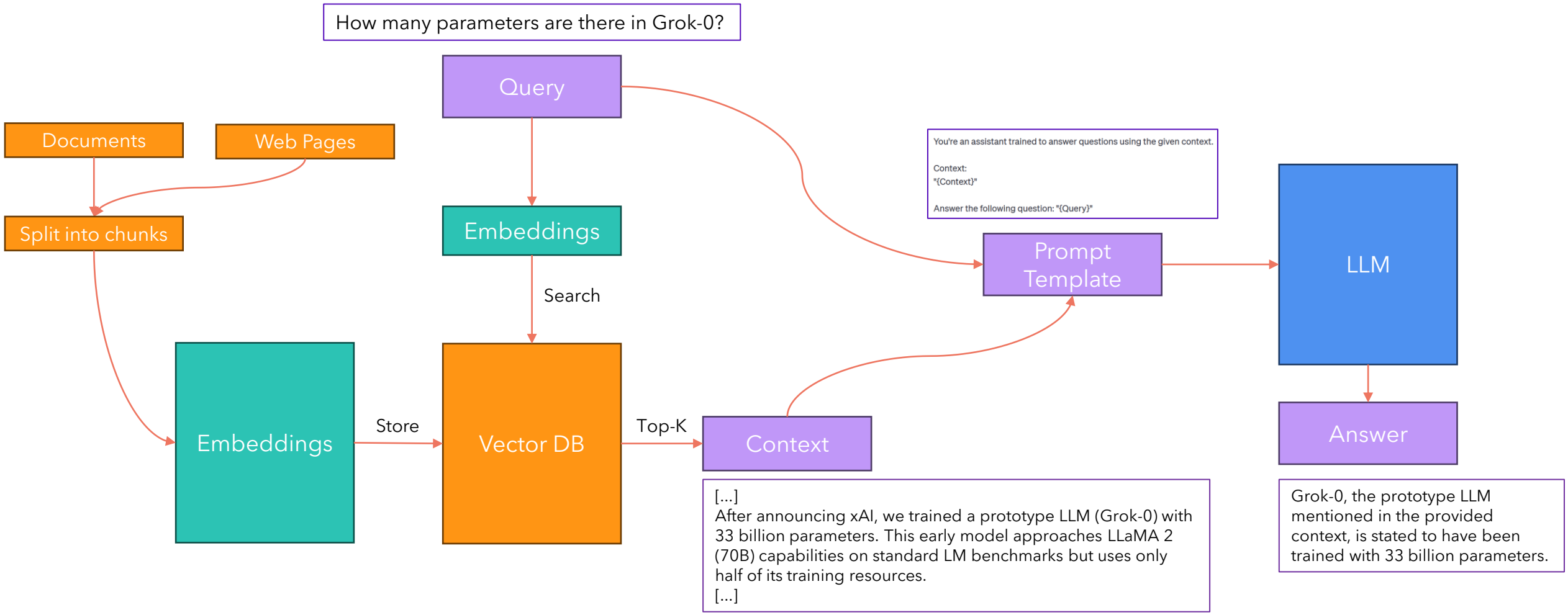
We can apply a linear layer to reduce the size of the embedding vector, for example from 768 to 512

We can use mean-pooling, max-pooling or just use the [cls] token as sentence embedding.

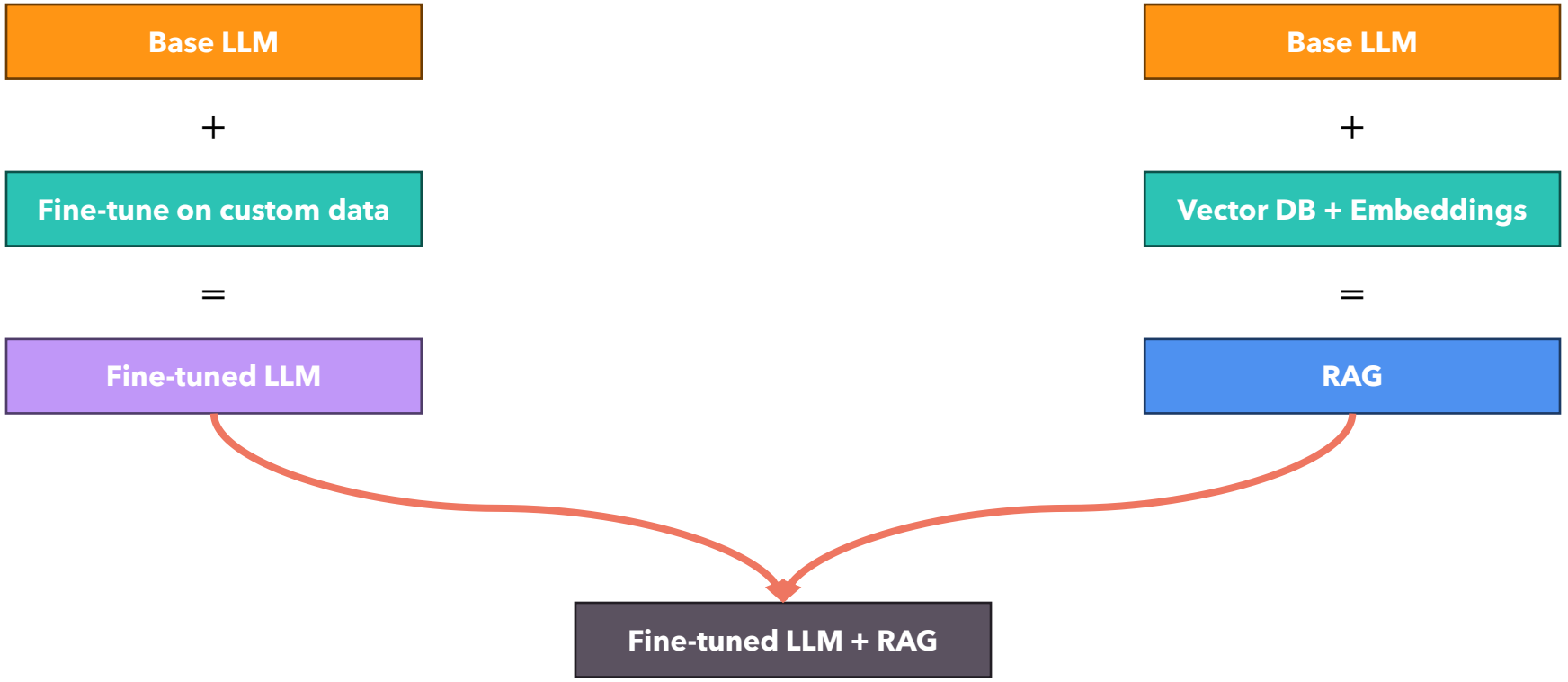
"My father plays with my me at the park"

"I play with my dad at the park"

QA with Retrieval Augmented Generation

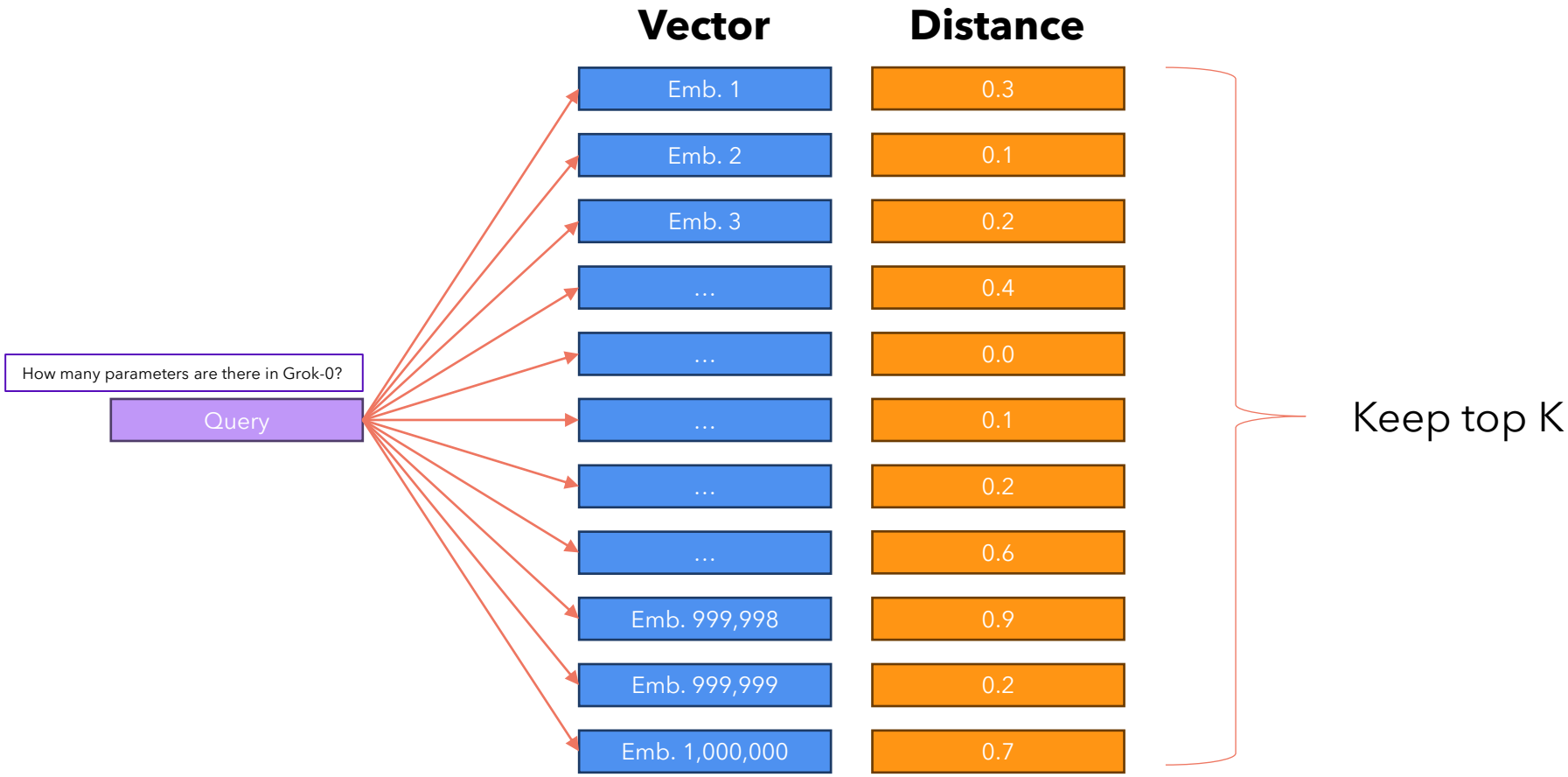


Strategies to teach new concepts to LLM



K-NN: a naïve approach

Imagine we want to search for the query in our database: a simple way would be comparing the query with all the vectors, sorting them by distance, and keeping the top K.

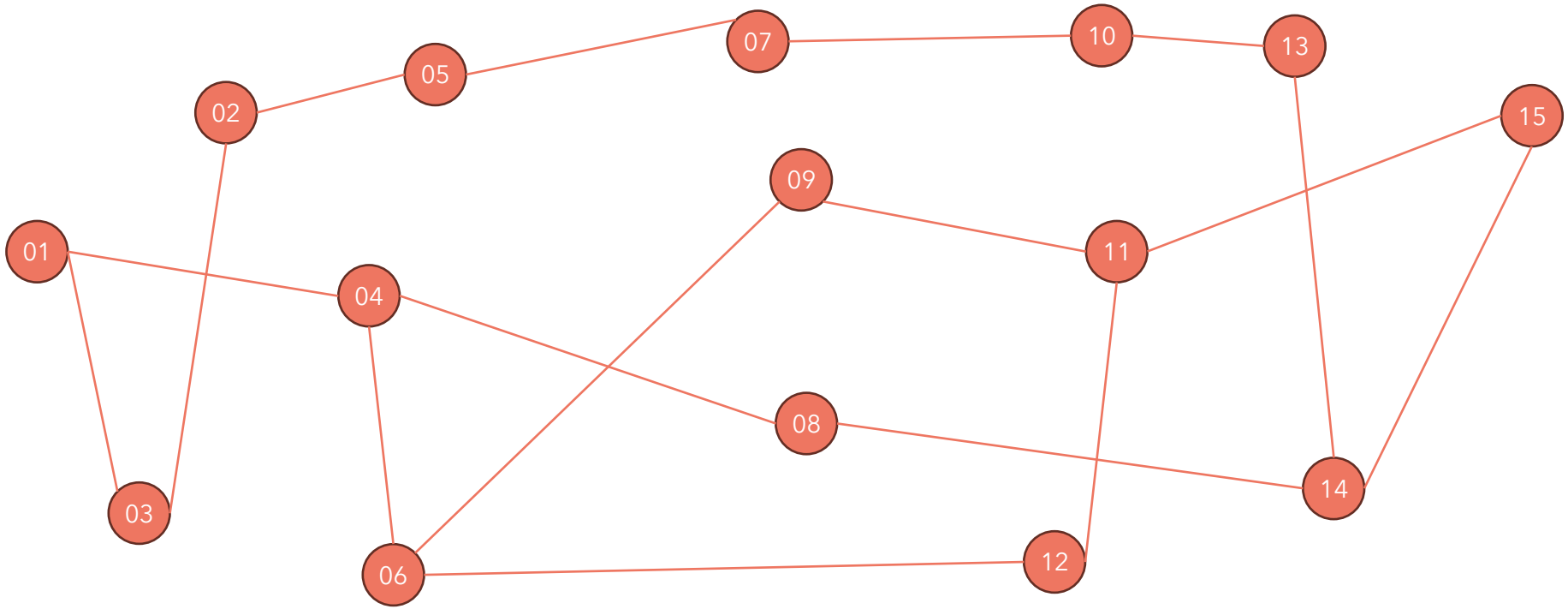


If there are N embedding vectors and each has D dimensions, the computational complexity is in the order of **O(N*D), too slow!**

Navigable Small Worlds

The NSW algorithm builds a graph that – just like Facebook friends – connects close vectors with each other but keeping the total number of connections small. For example, every vector may be connected to up to 6 other vectors (to mimic the Six Degrees of Separation).

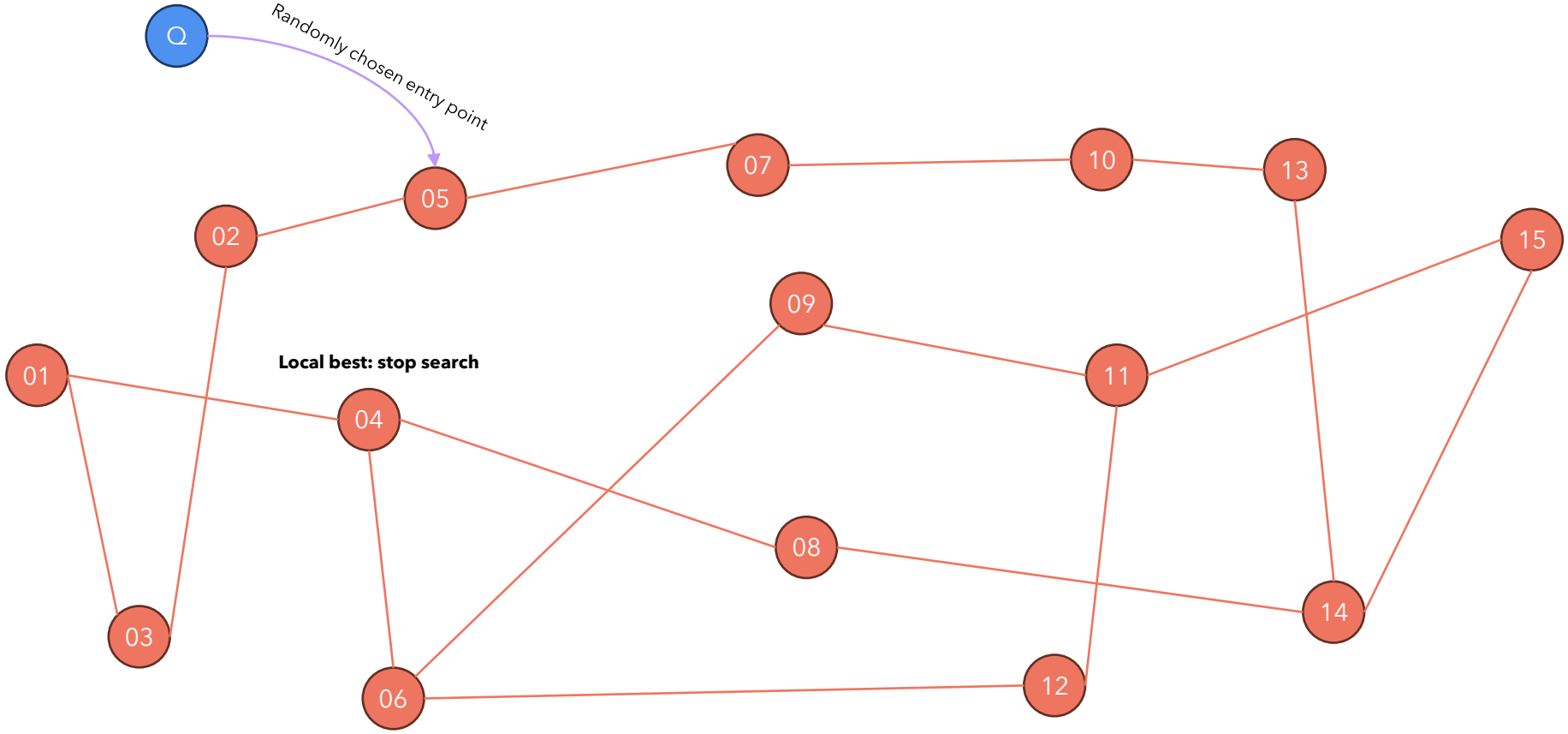
Node	Text
01	[…] The Transformer is a model […]
02	[…] Diagnose cancer with AI […]
03	[…] A transformer-based model […]
04	[…] The Transformer has 6 layers […]
05	[…] An MRI machine that costs 1$ […]
06	[…] The dot-product is a […]
07	[…] Big-Pharma is not so big […]
08	[…] Cross-Attention is a great […]
09	[…] To solve an ODE […]
10	[…] We are aging too fast […]
11	[…] Open-source models like […]
12	[…] MathBERT: a new model […]
13	[…] AI to control aging […]
14	[…] Attention is all you need […]
15	[…] LLaMA 2 has 7B params […]



Navigable Small Worlds: searching for K-NN

Given the following query: "*How many Encoder layers are there in the Transformer model?*"
How does the algorithm find the K Nearest Neighbors?

Node	Text
01	[…] The Transformer is a model […]
02	[…] Diagnose cancer with AI […]
03	[…] A transformer-based model […]
04	[…] The Transformer has 6 layers […]
05	[…] An MRI machine that costs 1$ […]
06	[…] The dot-product is a […]
07	[…] Big-Pharma is not so big […]
08	[…] Cross-Attention is a great […]
09	[…] To solve an ODE […]
10	[…] We are aging too fast […]
11	[…] Open-source models like […]
12	[…] MathBERT: a new model […]
13	[…] AI to control aging […]
14	[…] Attention is all you need […]
15	[…] LLaMA 2 has 7B params […]

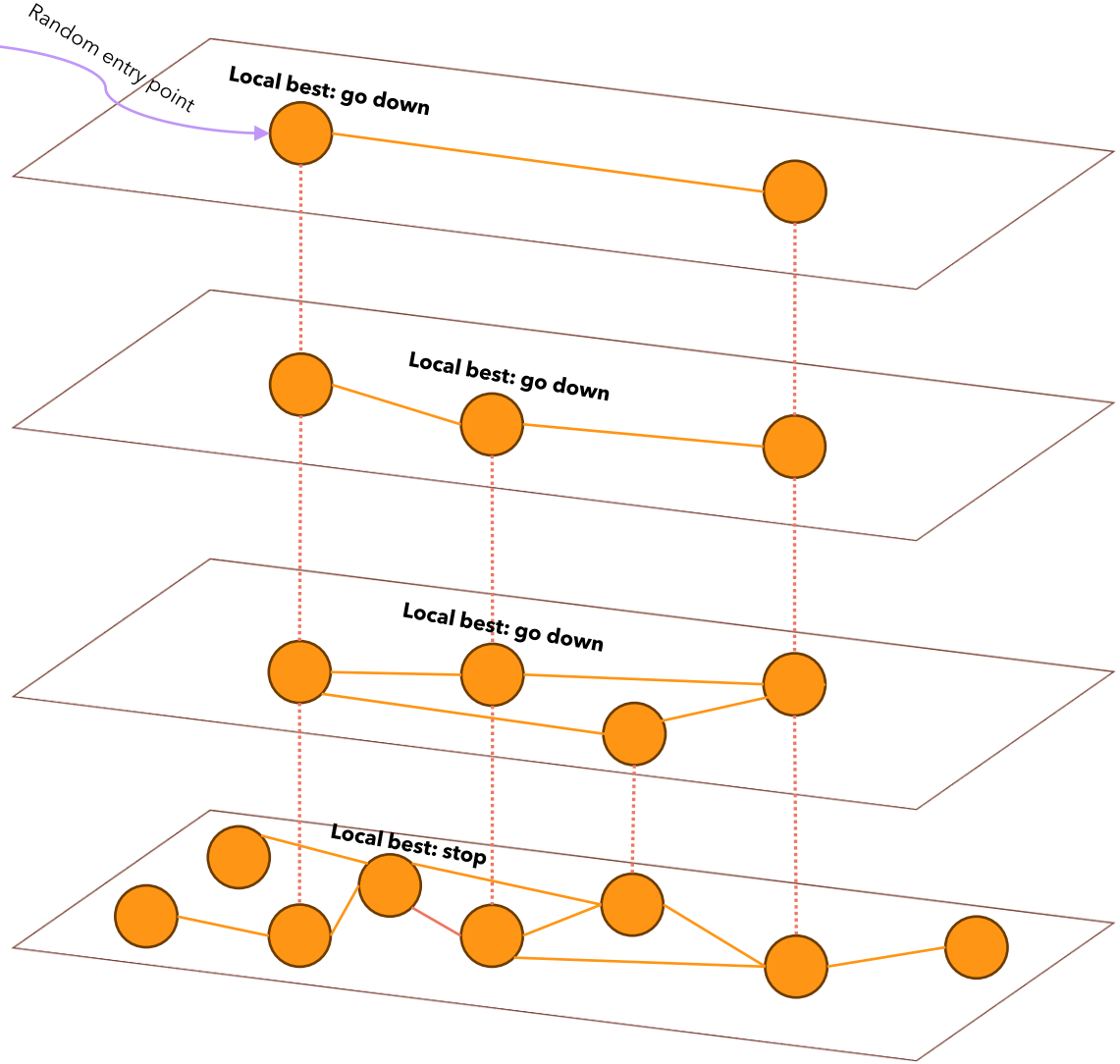


We repeat the search with randomly chosen starting points and then keep the top K among all the visited nodes.

HNSW: Hierarchical Navigable Small Worlds

Let's search!

We repeat the search with randomly chosen starting points (on the top layer) and then keep the top K among all the visited nodes.



Special thanks to Umar Jamil :)

 linkedin.com/in/ujamil

 x.com/hkproj

 umarjamil.org

 discord.gg/JRKsaNbhCg