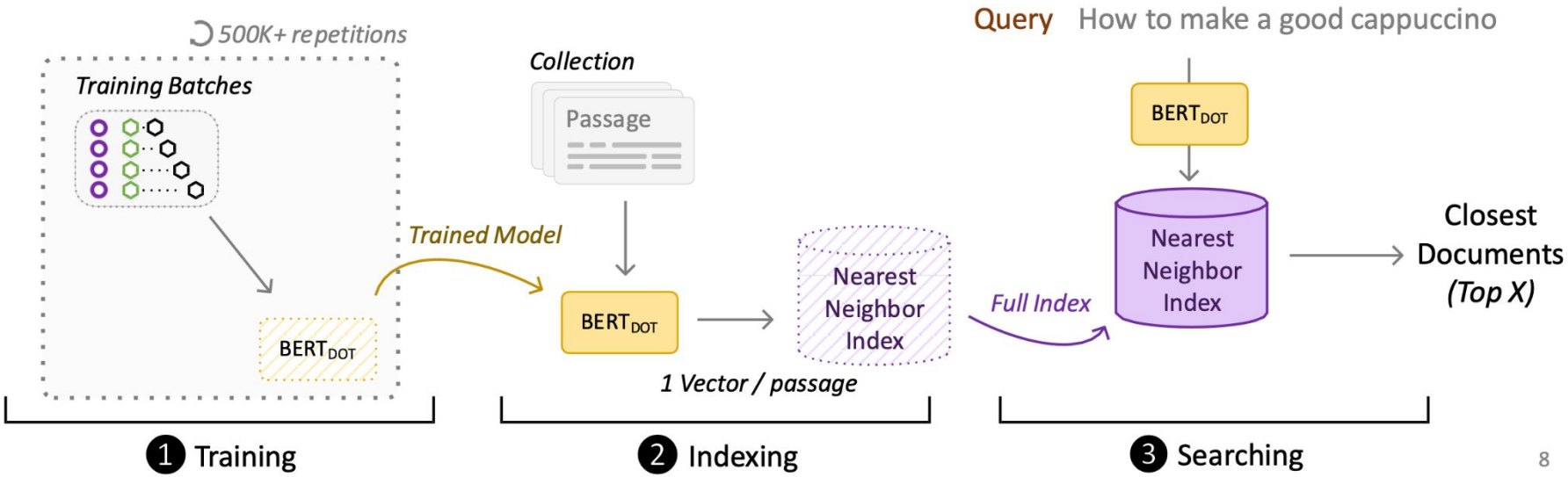
Introduction to LLM

Practice Session 12

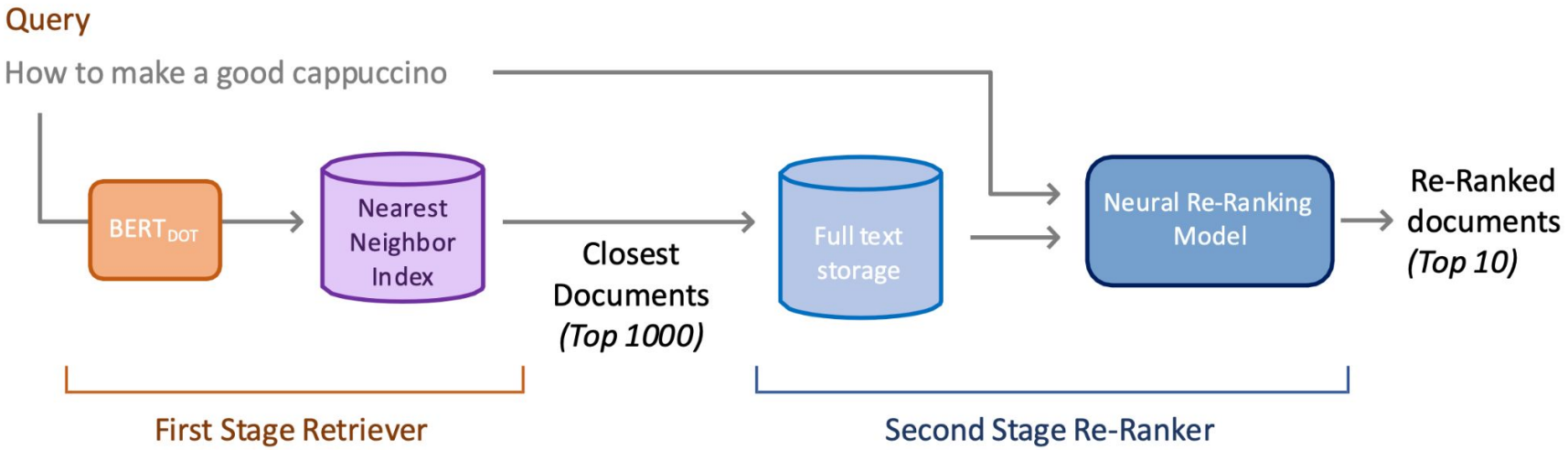
Dense Retrieval, QA, and ANN

High-level Dense Retriever Pipeline



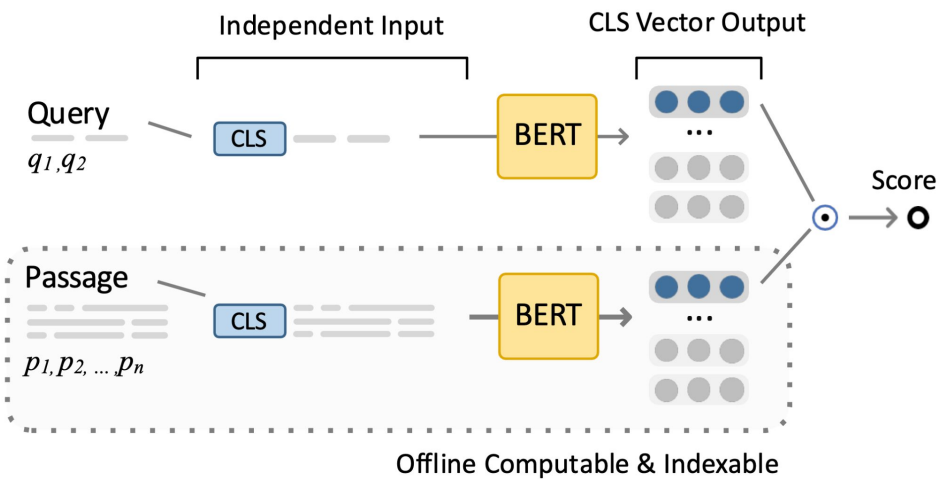
- Training data is prepared as a triplet (q, D+, D-)
  - The goal is to maximize the similarity score between (q, D+) and minimize that between (q, D-)
  - Hard training means that D- shouldn't be random, they must be confusing (similar but not relevant)

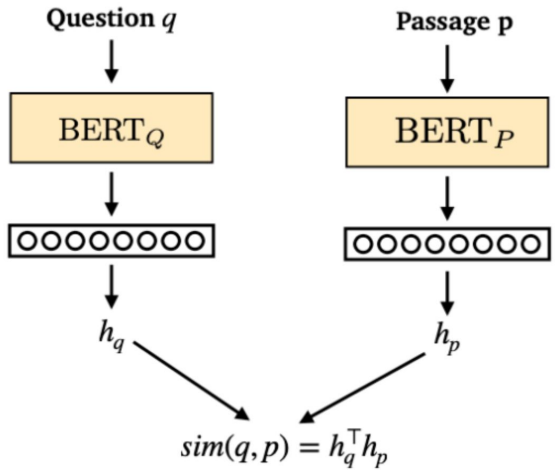
First Stage Retrievers



- First stage retrievers can be:-
  - Dense retrievers, e.g., BERT-DOT
  - Sparse retrievers, e.g., BM25
  - Hybrid retrievers (combination of both)

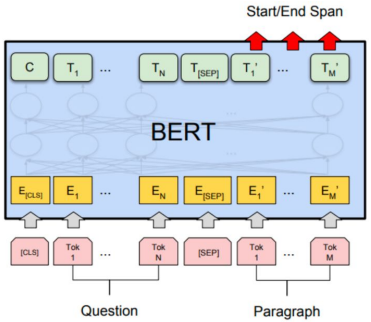
Dense Retriever Architecture (Bi-encoder)





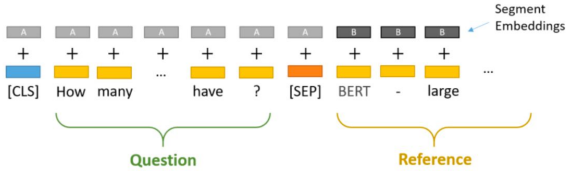
- Bi-encoders is an architecture paradigm in which there two networks:-
  - One network to embed the query
  - Another network to embed the documents
- BERT-DOT model is a type of Bi-encoder whose networks are based on BERT

Span-QA (Reading Comprehension)



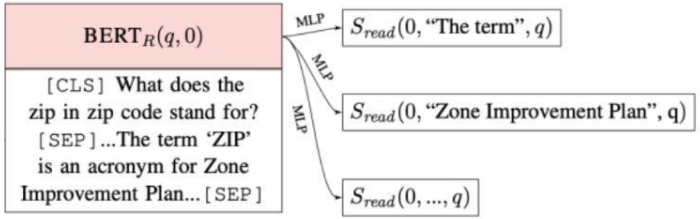
$$p_{\text{start}}(i) = \text{softmax}(\mathbf{W}_s \mathbf{h}_i)$$
$$p_{\text{end}}(i) = \text{softmax}(\mathbf{W}_e \mathbf{h}_i)$$

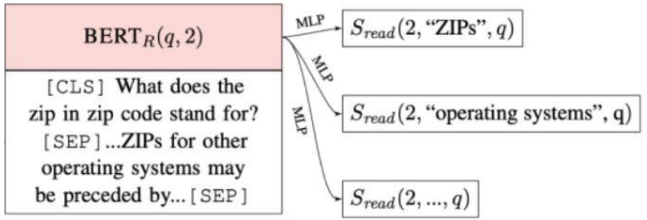
- This simplified version of QA aka **Reading Comprehension**.
  - (Passage, Question) ⇒ Answer



Question: How many parameters does BERT-large have?

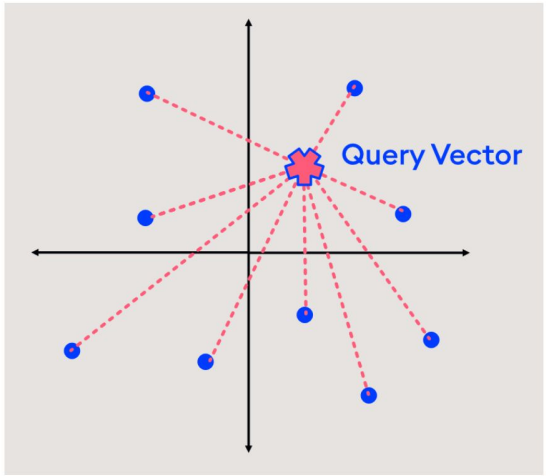
Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

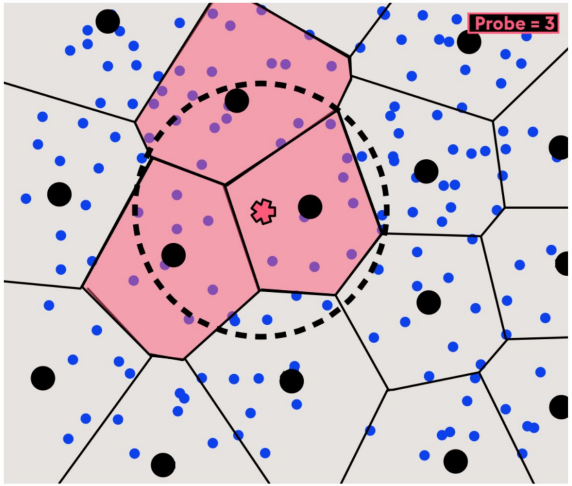




- Two heads (simple linear projection layers) are trained on top of BERT contexualized token embeddings
  - One head is trained to maximize probability for the token that should be the start position of the answer span
  - Another head is trained to maximize probability for the token that should be the end position of the answer span
- You enforce constraints, e.g., end position >= start position, or start & end position must be in paragraph segment

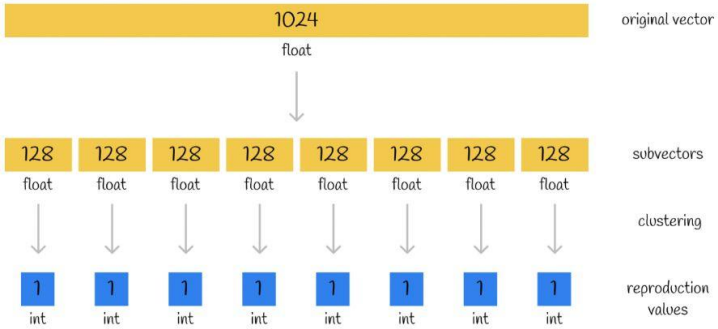
Indexing: Approximate Nearest Neighbor (ANN) Search

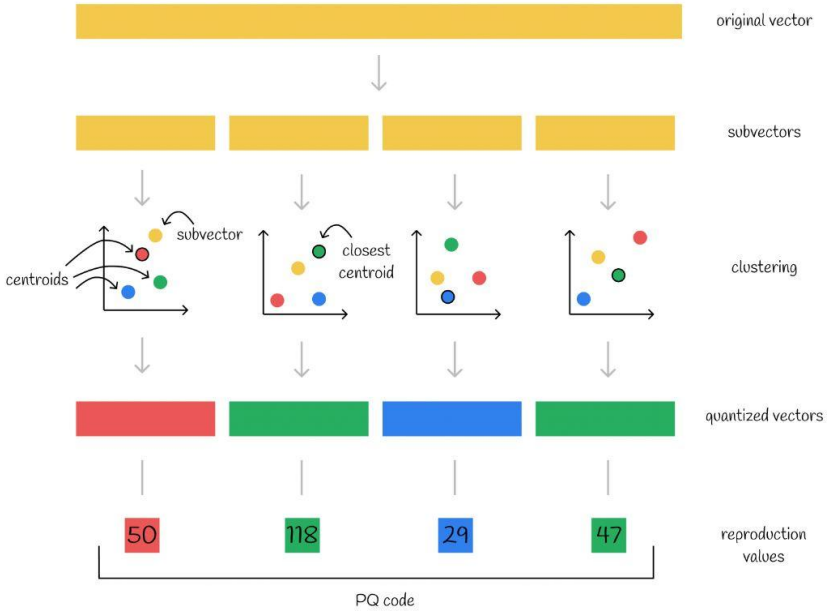




- Flat (exact) indexing is comparing query vector with all documents vectors
- IVF (inverted file) indexing is comparing query vector with cluster nodes
  - Then compare with documents vectors existing only on top-k clusters

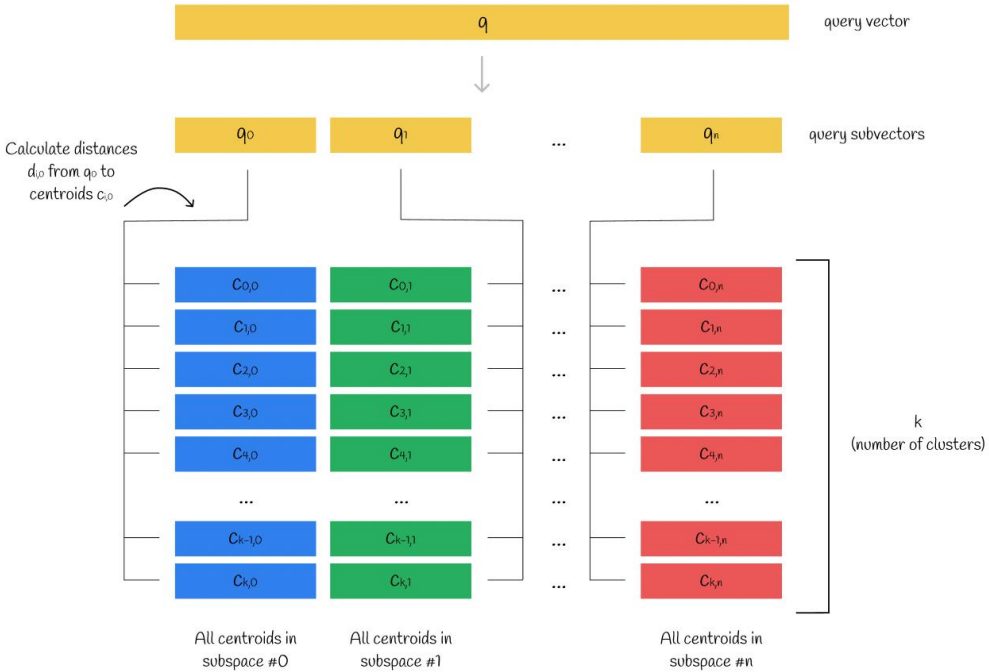
ANN: Product Quantization (PQ)





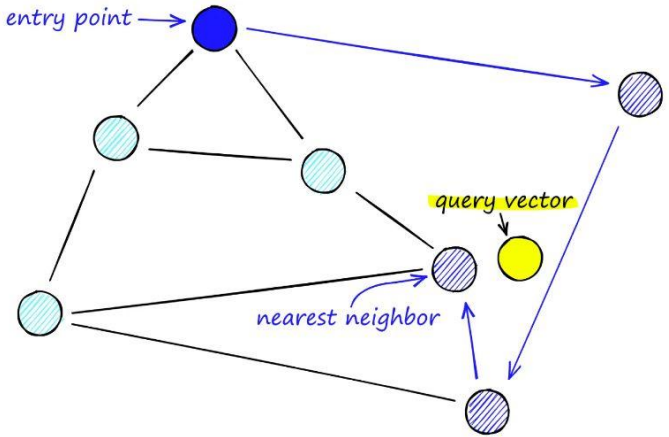
- Source: [Article Link]
- Two main hyperparameters:-
  - m (number of subvectors) and the original vector dimension should be divisible by it
  - nbits (2^{nbits} is total number of centroids) which is the size of the PQ code (representation of each subvector)

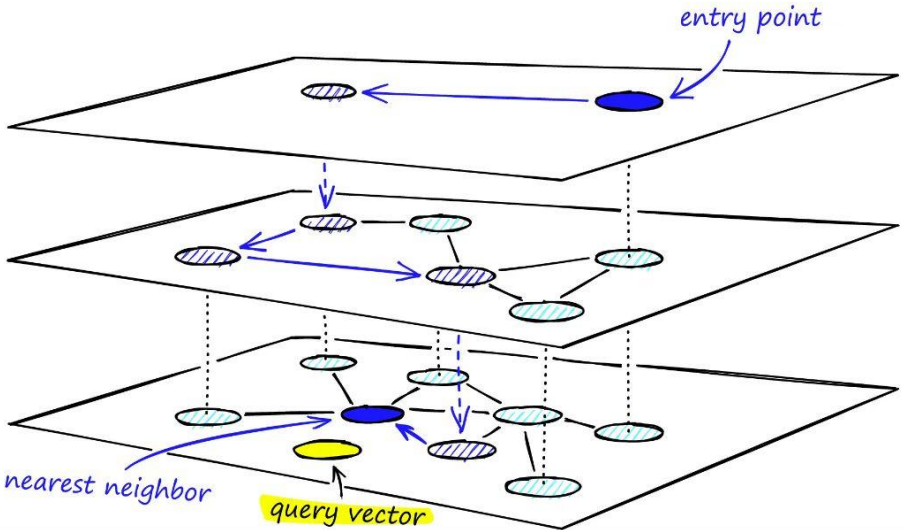
ANN: Product Quantization (PQ)



- Source: [Article Link]
- The distance between a query and a document is the sum of distances between their subvectors

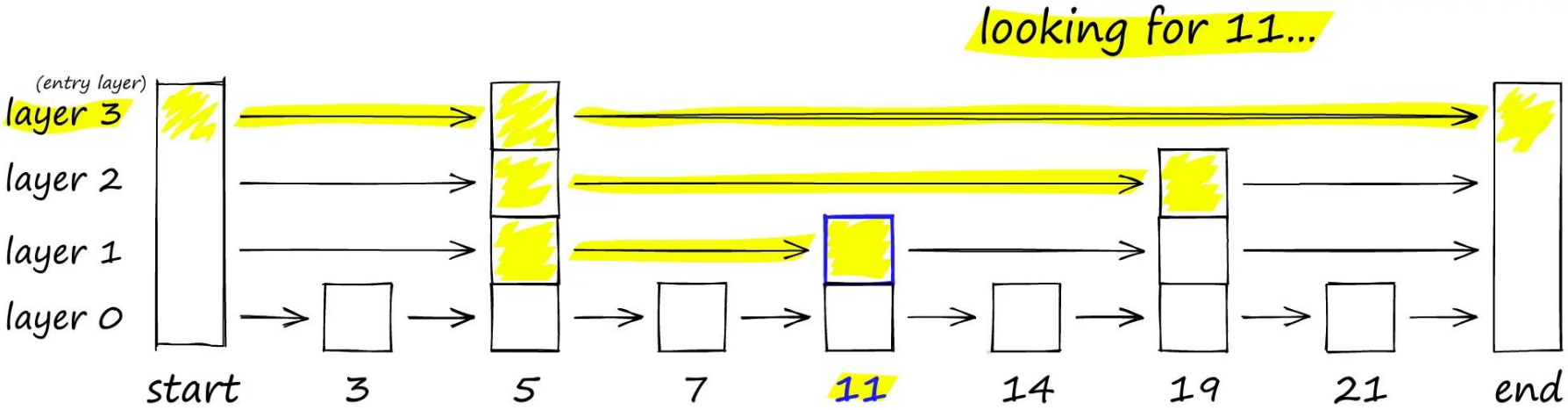
ANN: Hierarchical Navigable Small Worlds (HNSW)





- Source: [Article Link]
- Two main hyperparameters:-
  - m (number of neighbors for each vector) which is done offline during indexing
  - n (number of neighbors to visit) which is done online

ANN: Hierarchical Navigable Small Worlds (HNSW)



- Source: [Article Link]
- Typically implemented by a skip linked list data structure:-
  - You go to the lower layer when you reach the nearest node to the query in the current layer
  - The nodes you visit along the way in all layers will be the top-k nodes to the query

Timeline

Python & ML Foundations

- Train/Dev/Test
- Hyperparameters
- Loss Fn.

CL/NLP Basics

- Morphology
- Syntax
- PoS
- Parsing
- Semantics
- Lemma
- Ambiguity

Word Representation I:

- Basics
- Evaluation
- Inverted Index
- TF-IDF
- BM25
- LSA
- nDCG

Word Representation II:

- Statistical Methods
- Cosine Similarity
- Word2Vec
- CBOW & Skip-gram
- GloVe

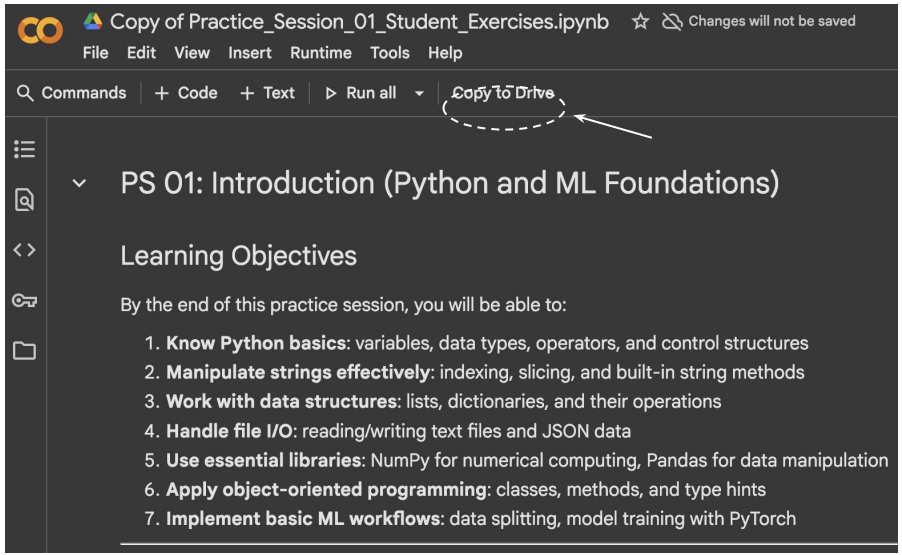
Dense Retrieval QA, and ANN

- Bi-encoders
- BERT-DOT model
- Hard Negatives
- Margin Loss Training
- Span QA
- Indexing Comparison

Rerankers, RAG, and Agents

- BERT-CAT
- Tool Use
- ReAct

PS12: Colab Notebook (Available on Moodle)



- https://colab.research.google.com/drive/1CQHvt18Y_tujmPTNGeLxLrgzmAxijyvM
- Before running any cell, please choose T4 GPU by clicking on "Runtime" in the main menu then on "Change runtime type".