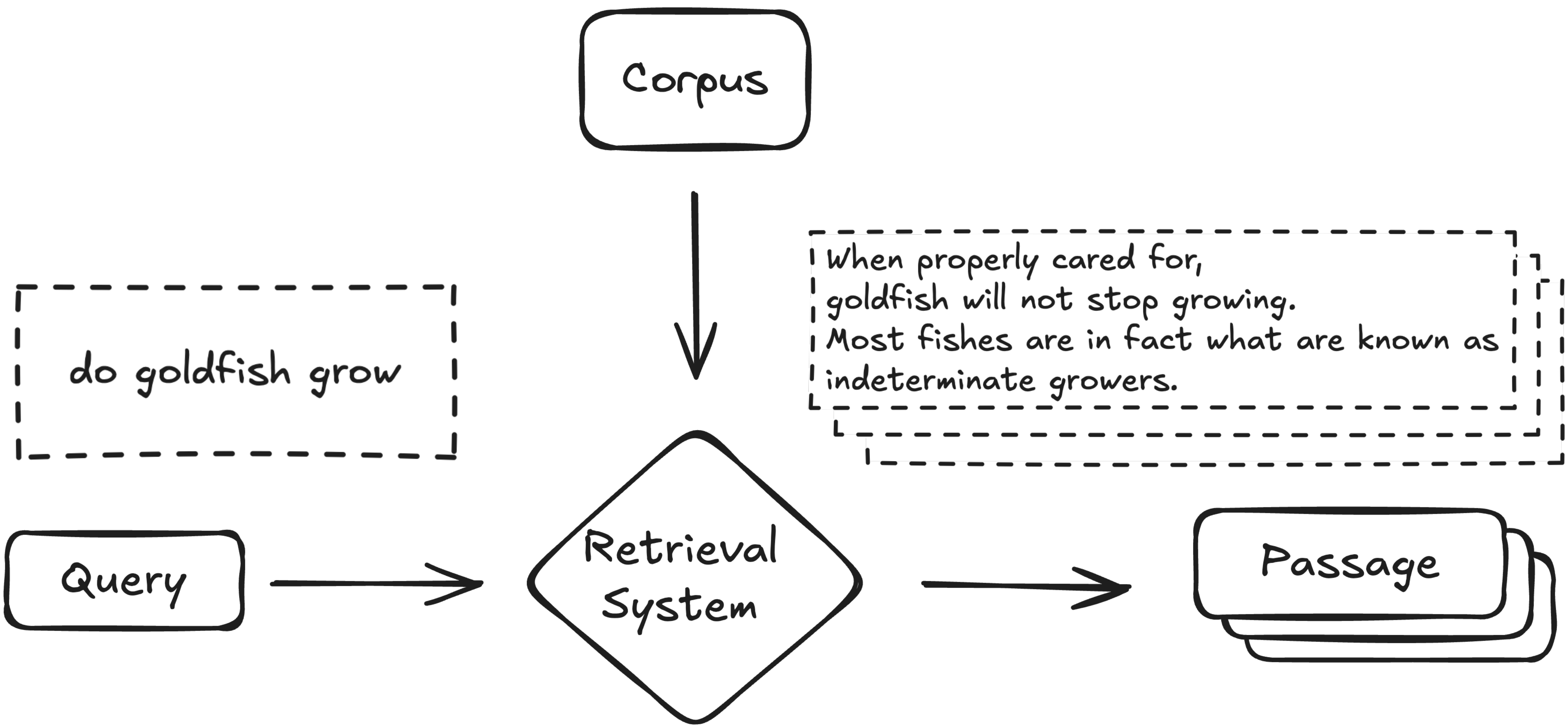
Lightweight Passage Re-ranking

Using Embeddings from Pre-trained Language Models

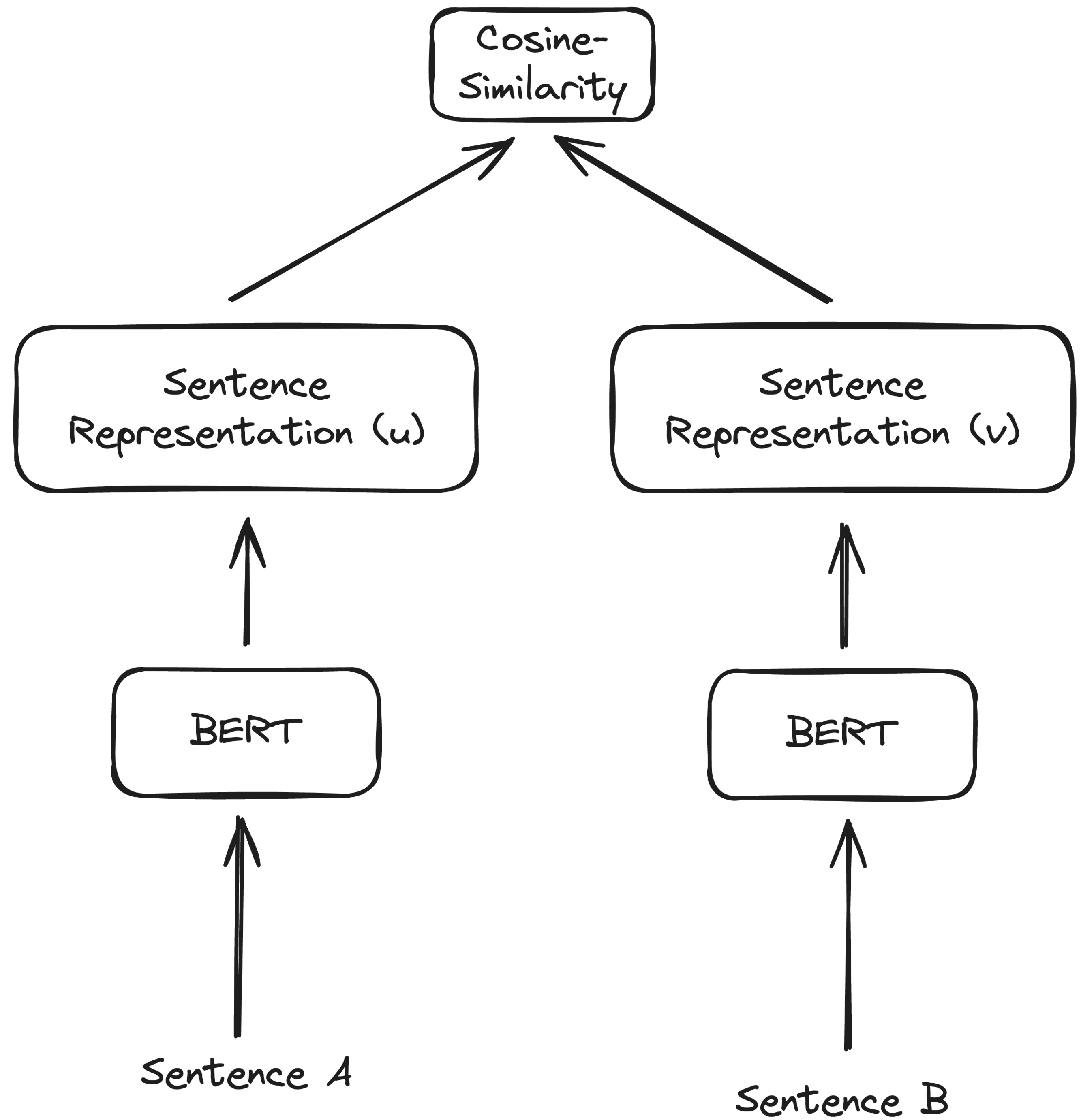
Tobias Jennerjahn - 17.09.2024

Passage Ranking



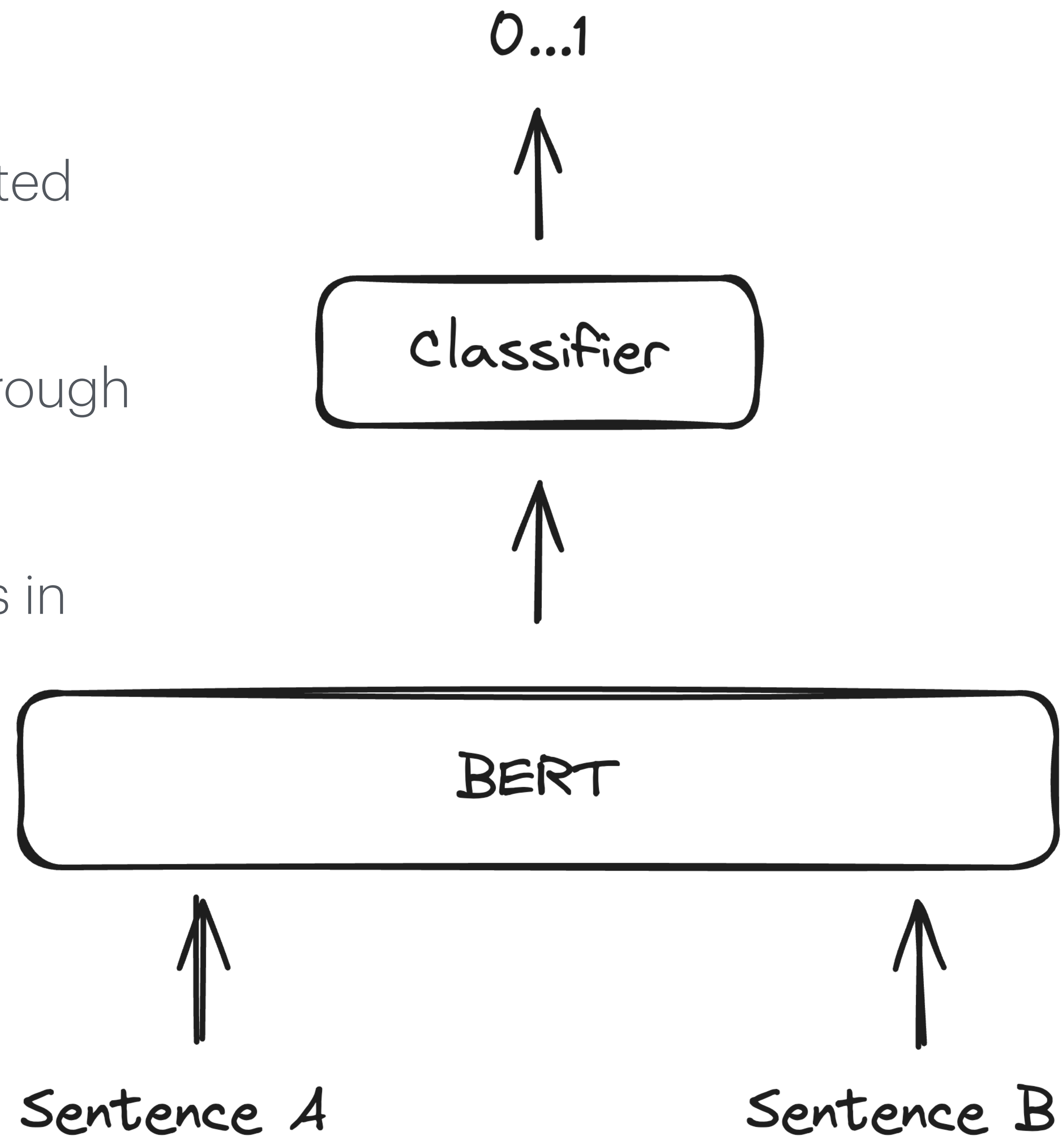
Bi-Encoder

- Separate encoding of queries and documents
- Enables offline indexing of document embeddings
- Efficient retrieval via similarity measures
- Limitation: No direct query-document interaction during encoding.

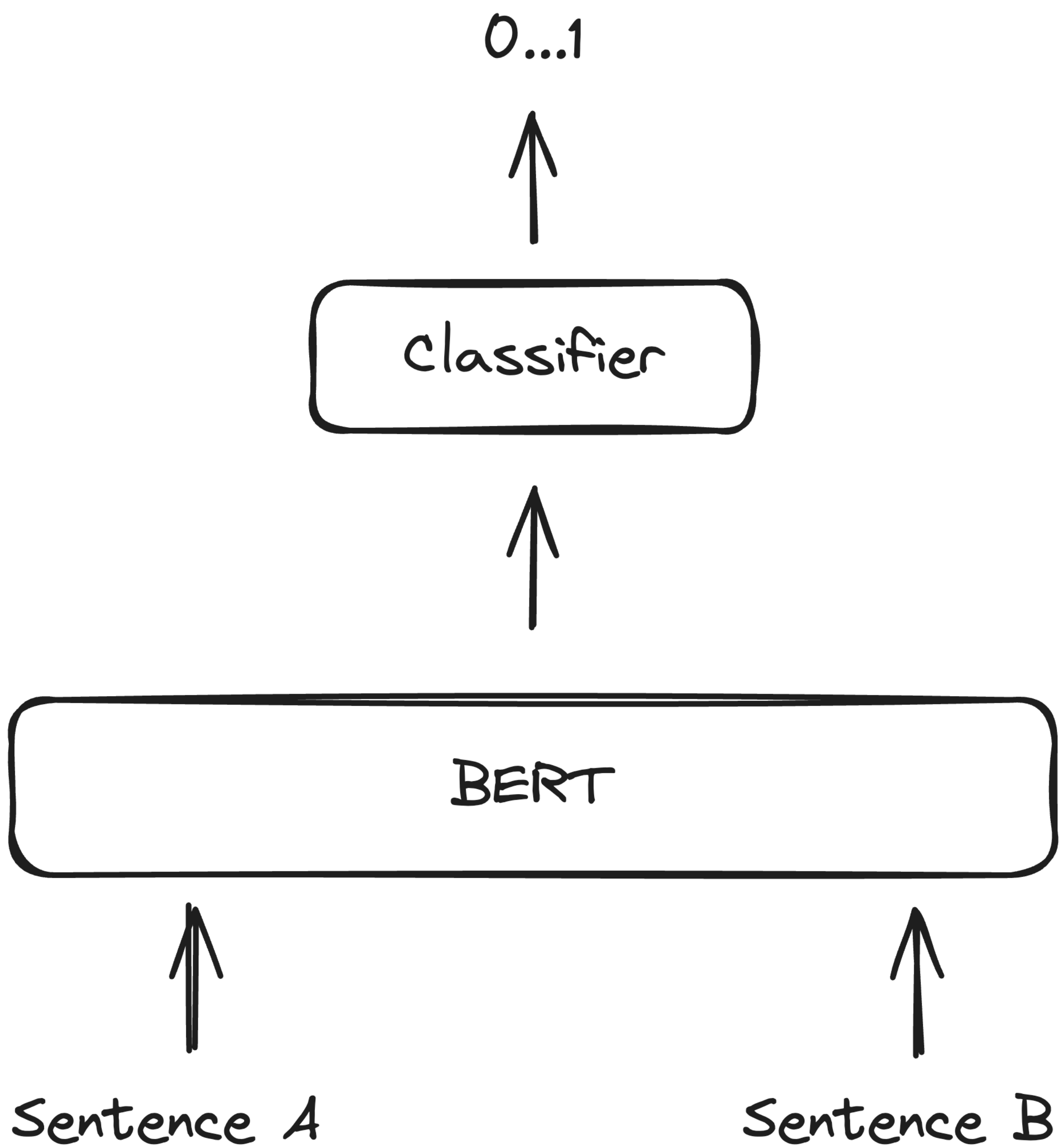


Cross-Encoder

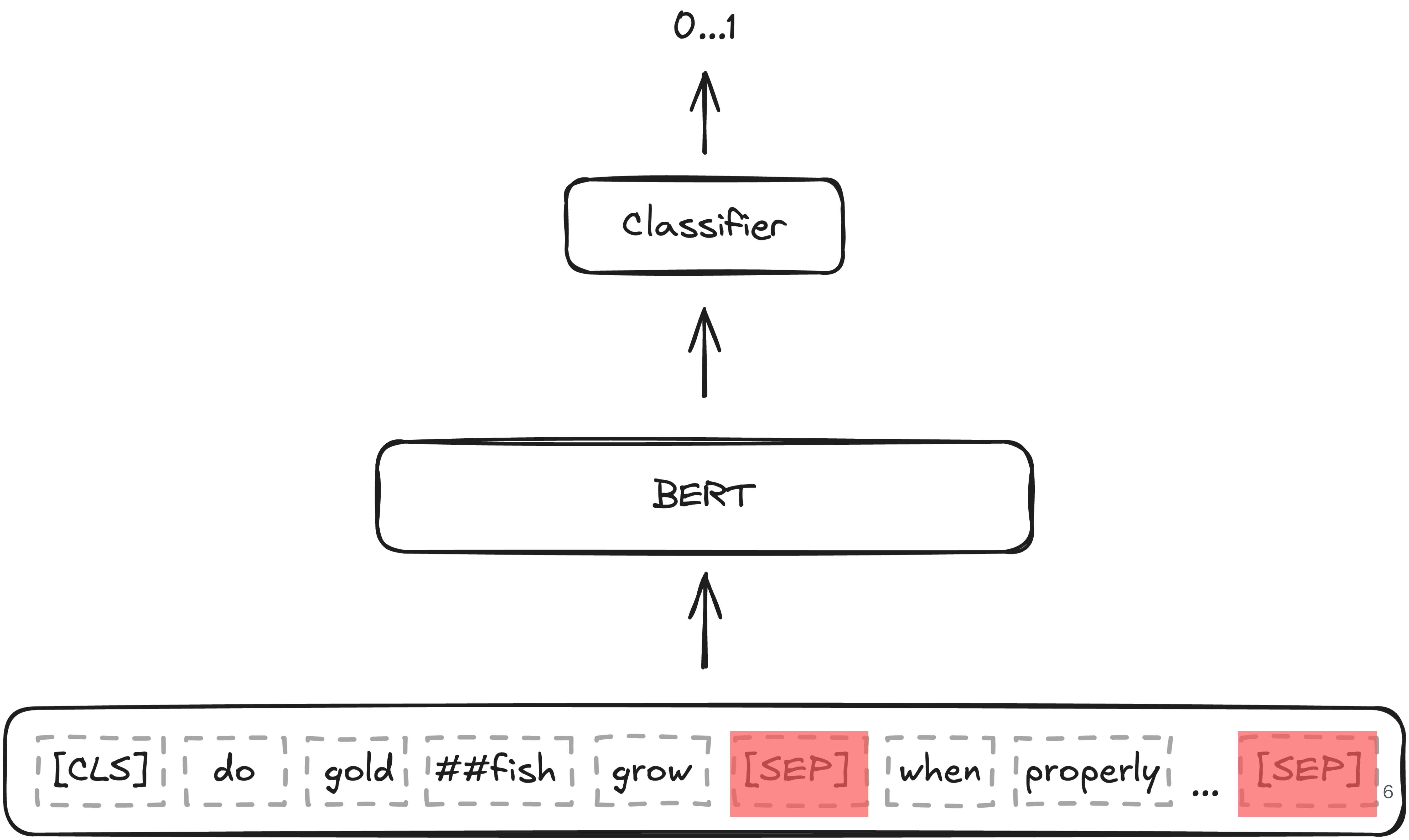
- Joint encoding of concatenated query-document pairs
- Captures rich interactions through attention mechanisms
- Achieves higher effectiveness in ranking tasks
- Computationally intensive



Cross-Encoder



Cross-Encoder



Problem Statement

- **Balancing Efficiency and Effectiveness:**
  - Bi-encoders offer efficient retrieval but lack interaction modeling
  - Cross-encoders capture rich interaction but are computationally intensive
- **Research Challenge:**
  - Can we enhance bi-encoder effectiveness using lightweight models
  - Is it possible to bridge the gap to cross-encoders without processing raw text jointly?

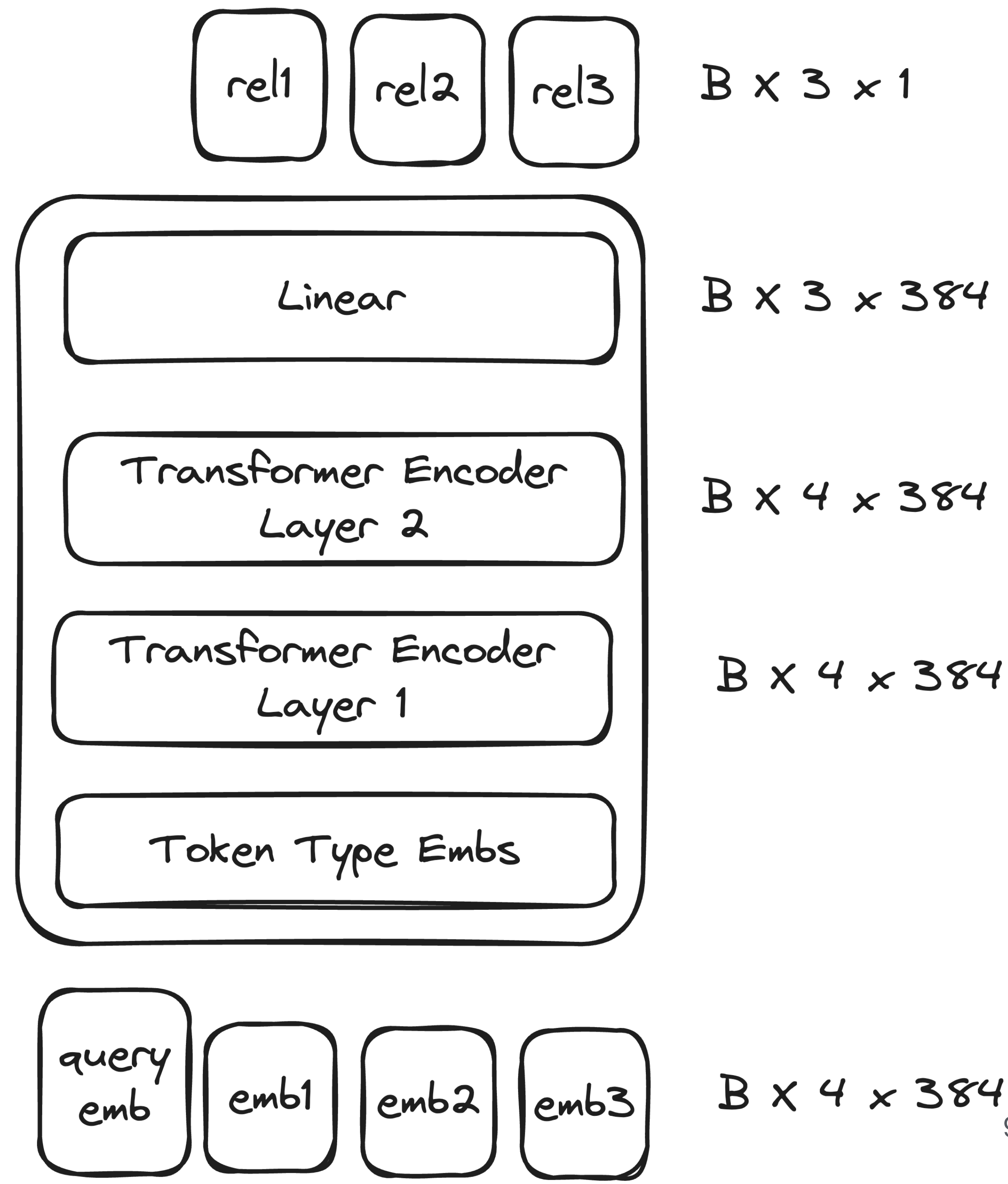
Proposed Approach

- Utilise pre-computed embeddings from a bi-encoder
- Introduce a lightweight transformer to model interactions
- Process query and passage embeddings together
- Maintain efficiency by avoiding full text processing

Methodology

Model Architecture

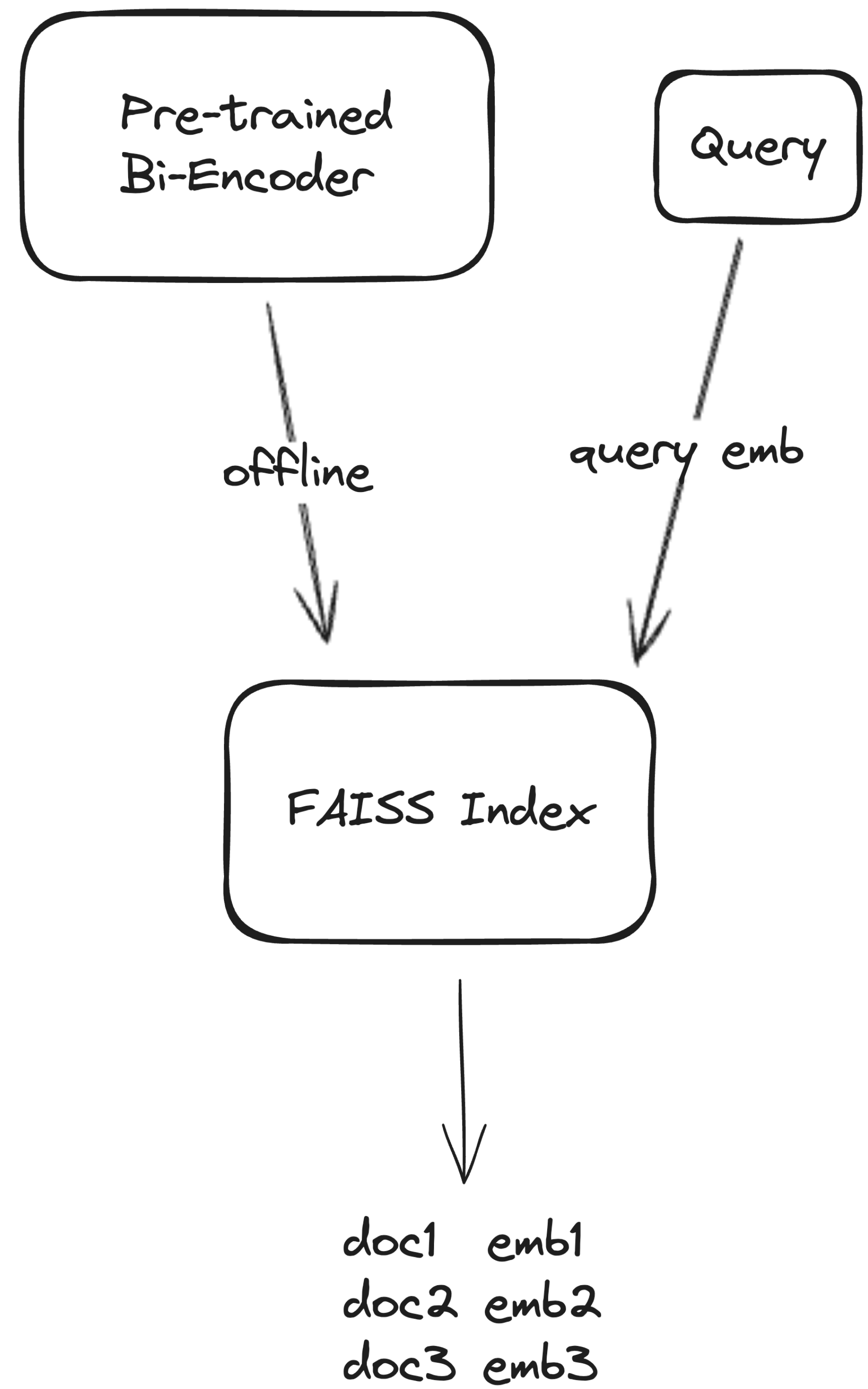
- Embeddings: Pre-computed using a bi-encoder model
- Concatenation: Query and document embeddings combined
- Token Type Embeddings: Added to distinguish query from passages
- Output: Relevance score for re-ranking



Methodology

Data Preparation

- Training Data based on standard IR Dataset (MS MARCO)
- Samples consist of 64-way tuples (1 query, 1 highly-ranked passage, 63 lower-ranked passages)
- Pre-compute embeddings for all documents using a bi-encoder



Preliminary Results

TREC-DL 2019 judged

run_name	nDCG@10	nDCG@64	nDCG@100	RR@10	RR@64	RR@100
RandomRun	0,056	0,097	0,120	0,156	0,177	0,177
pyserini-BM25	0,512	0,499	0,507	0,835	0,836	0,836
all-MiniLM-L6-v2	0,636	0,573	0,574	0,936	0,937	0,938
Our Model	0,641	0,575	0,574	0,936	0,937	0,937
ColBERT	0,732	0,691	0,685	0,954	0,954	0,954

Preliminary Results

TREC-DL 2019 judged

run_name	nDCG@10	nDCG@64	nDCG@100	RR@10	RR@64	RR@100
RandomRun	0,056	0,097	0,120	0,156	0,177	0,177
pyserini-BM25	0,512	0,499	0,507	0,835	0,836	0,836
all-MiniLM-L6-v2	0,636	0,573	0,574	0,936	0,937	0,938
Our Model	0,641	0,575	0,574	0,936	0,937	0,937
ColBERT	0,732	0,691	0,685	0,954	0,954	0,954

Analysis of Results

- **Observation:** The lightweight model didn't improve over the bi-encoder
- **Possible Reasons:**
  - Bi-encoder embeddings may lack rich interaction information
  - The lightweight transformer might be insufficient to model complex interactions
  - Operating on fixed embeddings may inherently limit potential gains

Analysis of Results

- **Observation:** The lightweight model didn't improve over the bi-encoder
- **Implications:**
  - The embeddings may not capture relationships between queries and passages
  - Need to consider architecture change, or end-to-end training

Future Work

- Check if the model actually just learned cosine-similarity
- Increase model size (Embedding Model *and* Reranking Model)
- Try End-to-End Training
- Try different Datasets

Methodology

Training Process

- Trained for a single epoch to prevent overfitting
- Loss function: Margin-MSE
- Batch Size: 32 @ 600k Steps
- Applied dropout with a rate of 0.1 in transformer layers
- Used layer normalization to improve convergence