الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

الذكاء الصنعي العملي | كلية الهندسة | المحاضرة 5

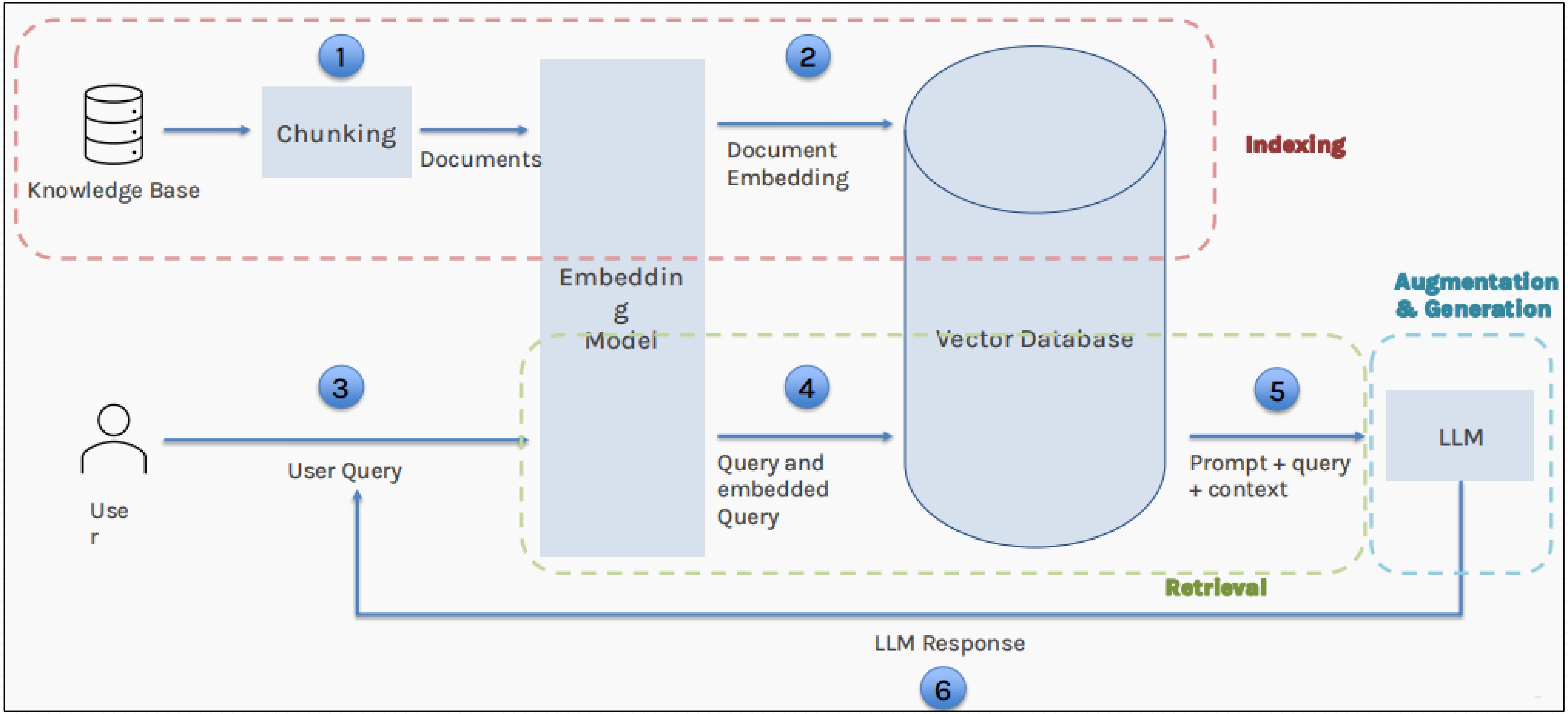
LLM in practice
Retrieval-Augmented-Generation (RAG)

د. رياض سنبل

Retrieval-Augmented-Generation (RAG)

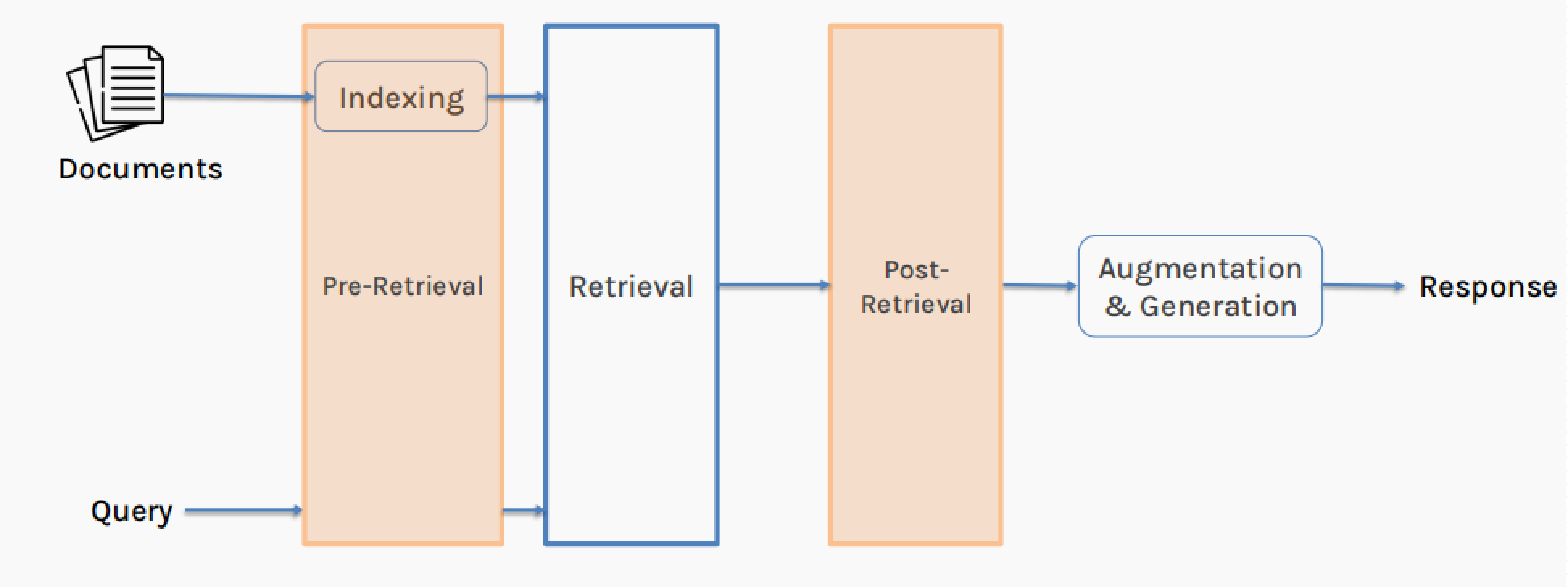
- RAG stands for Retrieval-Augmented-Generation.
- It is technique that improves the performance of a LLM, especially for tasks that require accurate and detailed information.
- **Benefits of RAG:**
  - Increased Accuracy: By incorporating external knowledge.
  - Contextual Relevance: RAG allows LLMs to tailor responses to specific contexts and user needs.
  - Up-to-Date Information: RAG can access and utilize the latest information from external sources.
  - Customization: RAG enables organizations to integrate their specific knowledge bases and data into LLM applications.
  - Improved Reliability: Users can verify the sources of information used by RAG models, enhancing trust in their responses.

RAG



1
2
Chunking
Documents
Document
Embedding
Indexing
Knowledge Base
Embeddin
g
Model
Vector Database
Augmentation
& Generation
3
4
5
User Query
Query and
embedded
Query
Prompt + query
+ context
LLM
Use
r
Retrieval
LLM Response
6

RAG



The Pre-Retrieval Phase deals with:
- Chunking the data
- Converting the chunks into embeddings
- Handling the embeddings

The Post-Retrieval phase deals with polishing what was obtained from the retriever.

Pre-Retrieval Optimization (Improve the chunking process)

Let's take another example

Advancements in Transfer Learning for NLP

Abstract:
"Transfer learning has become a crucial technique in NLP. This paper explores recent advancements, including fine-tuning pre-trained models like BERT and GPT-3, and domain adaptation methods. Our experiments demonstrate significant improvements in performance across various NLP tasks."

Methodology:
"We fine-tuned BERT and GPT-3 models on specific NLP tasks, adapting them to different domains. Domain adaptation involved additional pre-training on domain-specific data. Our approach leverages the pre-trained knowledge and adapts it to new tasks, achieving higher accuracy and efficiency."

Results:
"The results indicate a 20% increase in accuracy for domain-specific tasks using our fine-tuning and domain adaptation techniques. We observed substantial performance gains compared to baseline models."

Now, if we do character splitting for chunks (chunk size=200), we get:

Chunk 1:
Recent techniques in transfer learning for NLP Abstract: Transfer learning has become a crucial technique in NLP. This paper explores recent advancements, including fine-tuning pre-trained models like BERT and GPT-3, and **dom**

Chunk 2:
ain adaptation methods. Our experiments demonstrate significant improvements in performance across various **NLP tasks. Methodology:** We fine-tuned BERT and GPT-3 models on specific NLP tasks, adapting them to different do

....

Pre-Retrieval Optimization (Improve the chunking process)

Let’s take another example

Advancements in Transfer Learning for NLP

Abstract:
"Transfer learning has become a crucial technique in NLP. This paper explores recent advancements, including fine-tuning pre-trained models like BERT and GPT-3, and domain adaptation methods. Our experiments demonstrate significant improvements in performance across various NLP tasks."

Methodology:
"We fine-tuned BERT and GPT-3 models on specific NLP tasks, adapting them to different domains. Domain adaptation involved additional pre-training on domain-specific data. Our approach leverages the pre-trained knowledge and adapts it to new tasks, achieving higher accuracy and efficiency."

Results:
"The results indicate a 20% increase in accuracy for domain-specific tasks using our fine-tuning and domain adaptation techniques. We observed substantial performance gains compared to baseline models."

Now, if we do character splitting for chunks (chunk size=200), we get:

Chunk 1:
Recent techniques in transfer learning for NLP Abstract: Transfer learning has become a crucial technique in NLP. This paper explores recent advancements, including fine-tuning pre-trained models like BERT and GPT-3, and **dom**

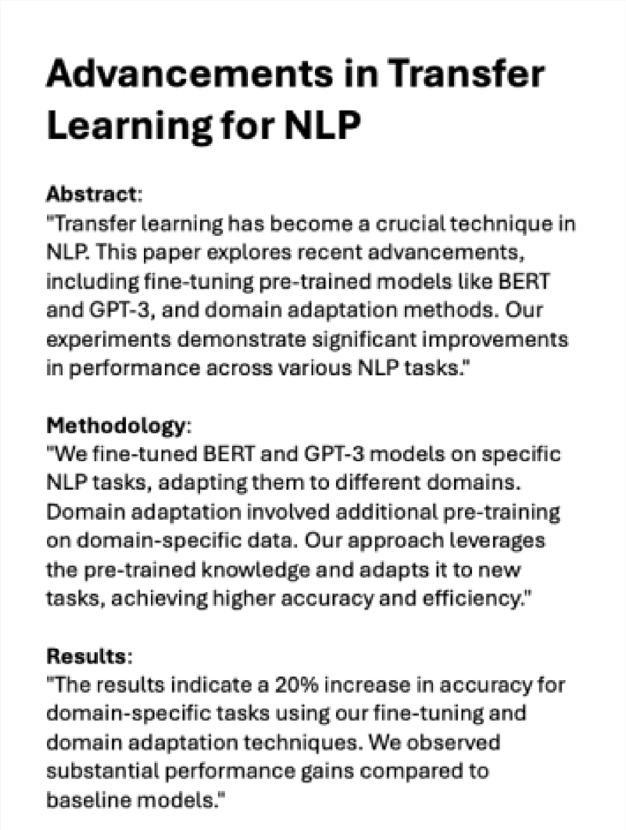
Chunk 2:
ain adaptation methods. Our experiments demonstrate significant improvements in performance across various **NLP tasks. Methodology:** We fine-tuned BERT and GPT-3 models on specific NLP tasks, adapting them to different do

....

Pre-Retrieval Optimization (Improve the chunking process)

Better Approach!
Semantic chunking

Let's take another example


Advancements in Transfer Learning for NLP

Abstract:
"Transfer learning has become a crucial technique in NLP. This paper explores recent advancements, including fine-tuning pre-trained models like BERT and GPT-3, and domain adaptation methods. Our experiments demonstrate significant improvements in performance across various NLP tasks."

Methodology:
"We fine-tuned BERT and GPT-3 models on specific NLP tasks, adapting them to different domains. Domain adaptation involved additional pre-training on domain-specific data. Our approach leverages the pre-trained knowledge and adapts it to new tasks, achieving higher accuracy and efficiency."

Results:
"The results indicate a 20% increase in accuracy for domain-specific tasks using our fine-tuning and domain adaptation techniques. We observed substantial performance gains compared to baseline models."

The optimal way to do it would be:

Chunk 1:
Advancements in Transfer Learning for NLP

Chunk 2:
Abstract:
Transfer learning has become a crucial technique in NLP. This paper explores recent advancements, including fine-tuning pre-trained models like BERT and GPT-3, and domain adaptation methods. Our experiments demonstrate significant improvements in performance across various NLP tasks.

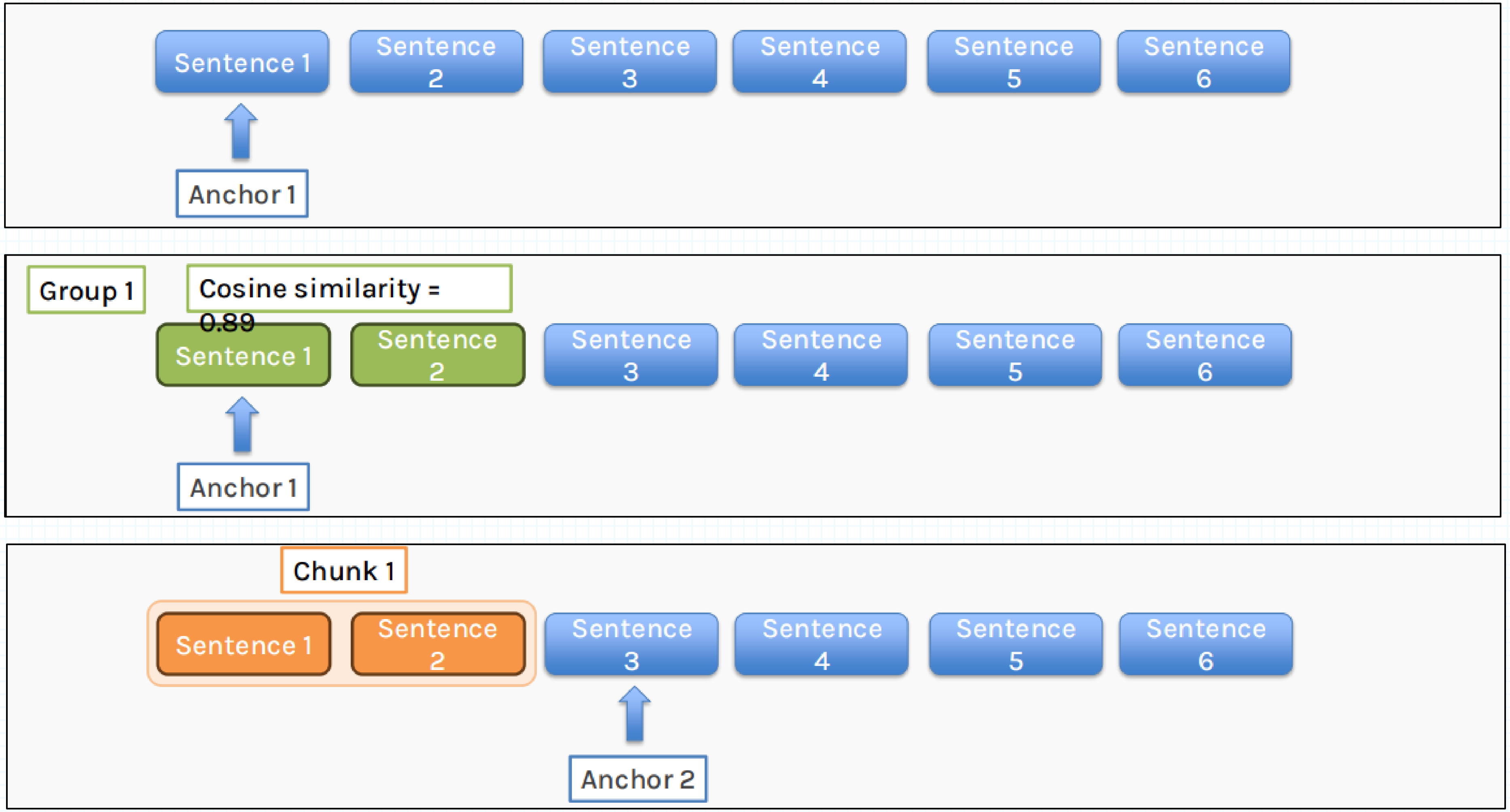
Chunk 3:
Methodology:
We fine-tuned BERT and GPT-3 models on specific NLP tasks, adapting them to different domains. Domain adaptation involved additional pre-training on domain-specific data. Our approach leverages the pre-trained knowledge and adapts it to new tasks, achieving higher

Pre-Retrieval Optimization (Improve the chunking process)

- **Semantic Chunking – Steps**
  - Splitting: We split the document to sentences using separators(.,?,!).
  - Grouping: Select anchor sentences and choose how many sentences to consider at either side of the anchor (window size).
  - Similarity Check: Calculate the distance between the group of sentences (e.g.: cosine similarity).
  - Chunking: Chunk together the similar sentences.

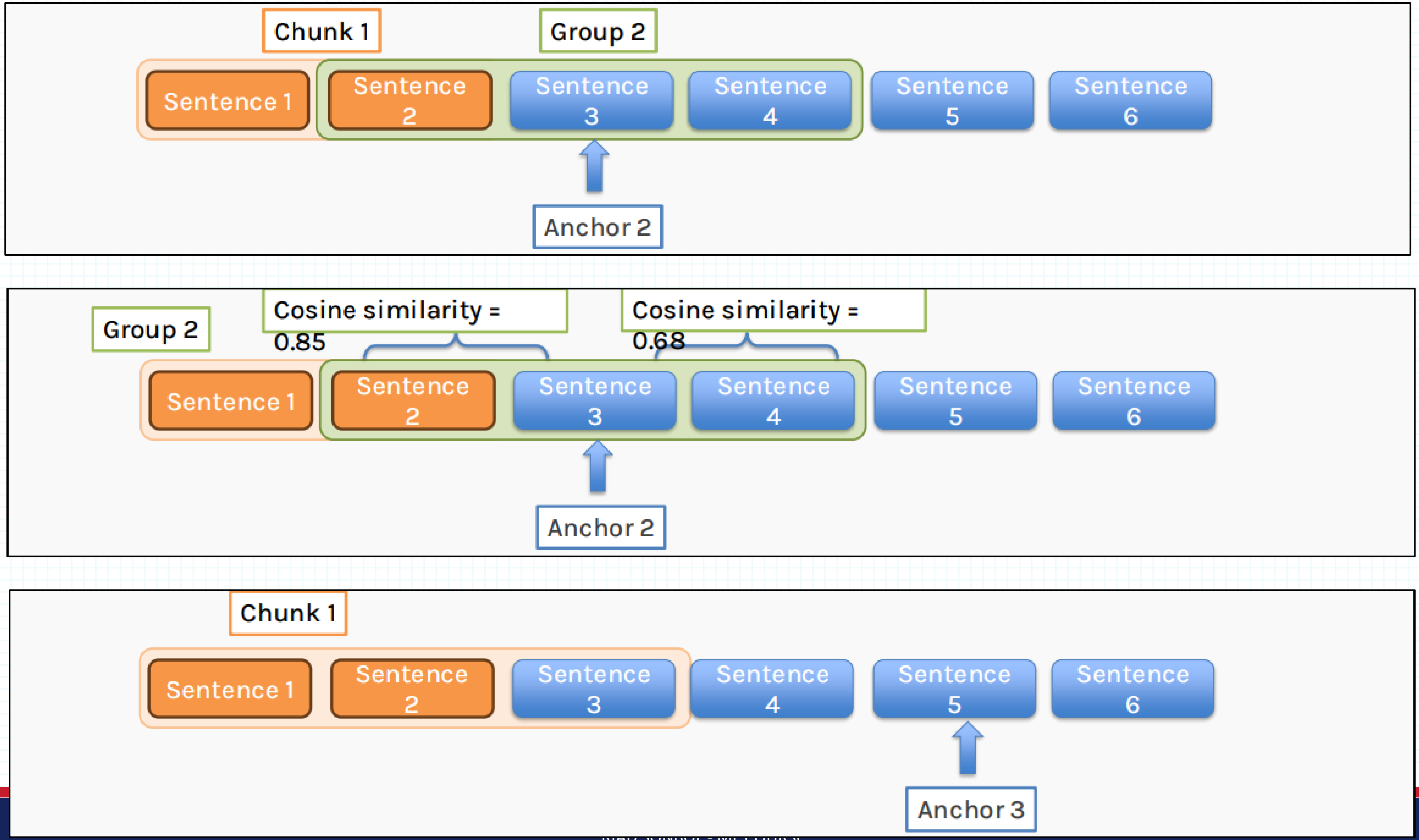
Pre-Retrieval Optimization (Improve the chunking process)

Semantic Chunking



Pre-Retrieval Optimization (Improve the chunking process)

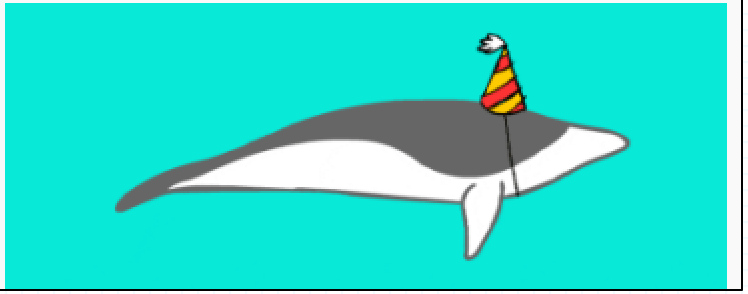
Semantic Chunking



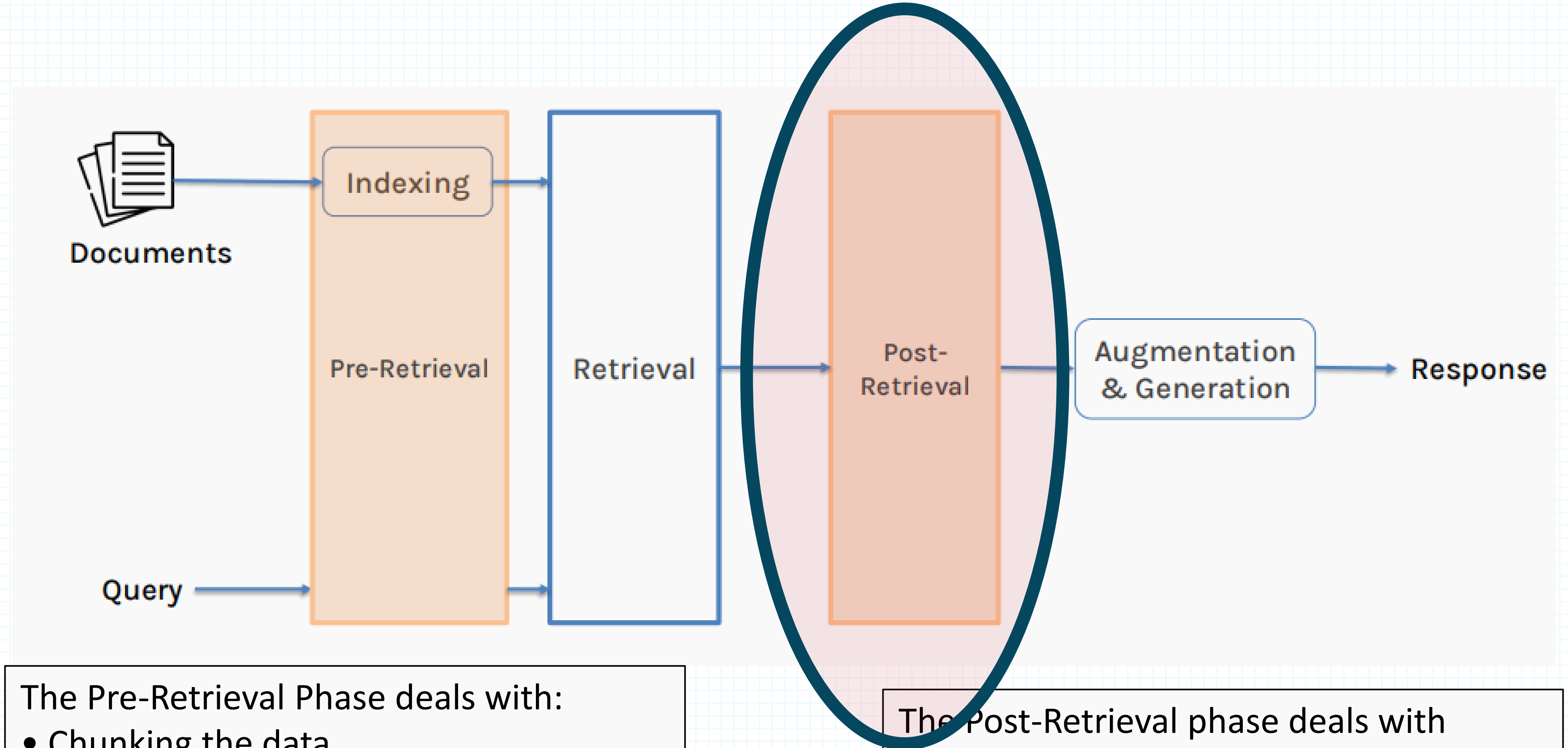
Pre-Retrieval Optimization (Query Manipulation)

- 2 problems can come up when it comes to queries provided by a user:
  - The query is 'cluttered': This can be due to it being sprinkled with a lot of irrelevant information.
  - The query is ambiguous: The query doesn't have sufficient information
- Example:

❓Original Query	We have an essay due tomorrow. We have to write about some animal. I love penguins. I could write bout them. But I could also write about dolphins. Are they animals? Maybe. Let's do dolphins. Where do they live, for example?
🧠Rewritten query	Where do dolphins live?



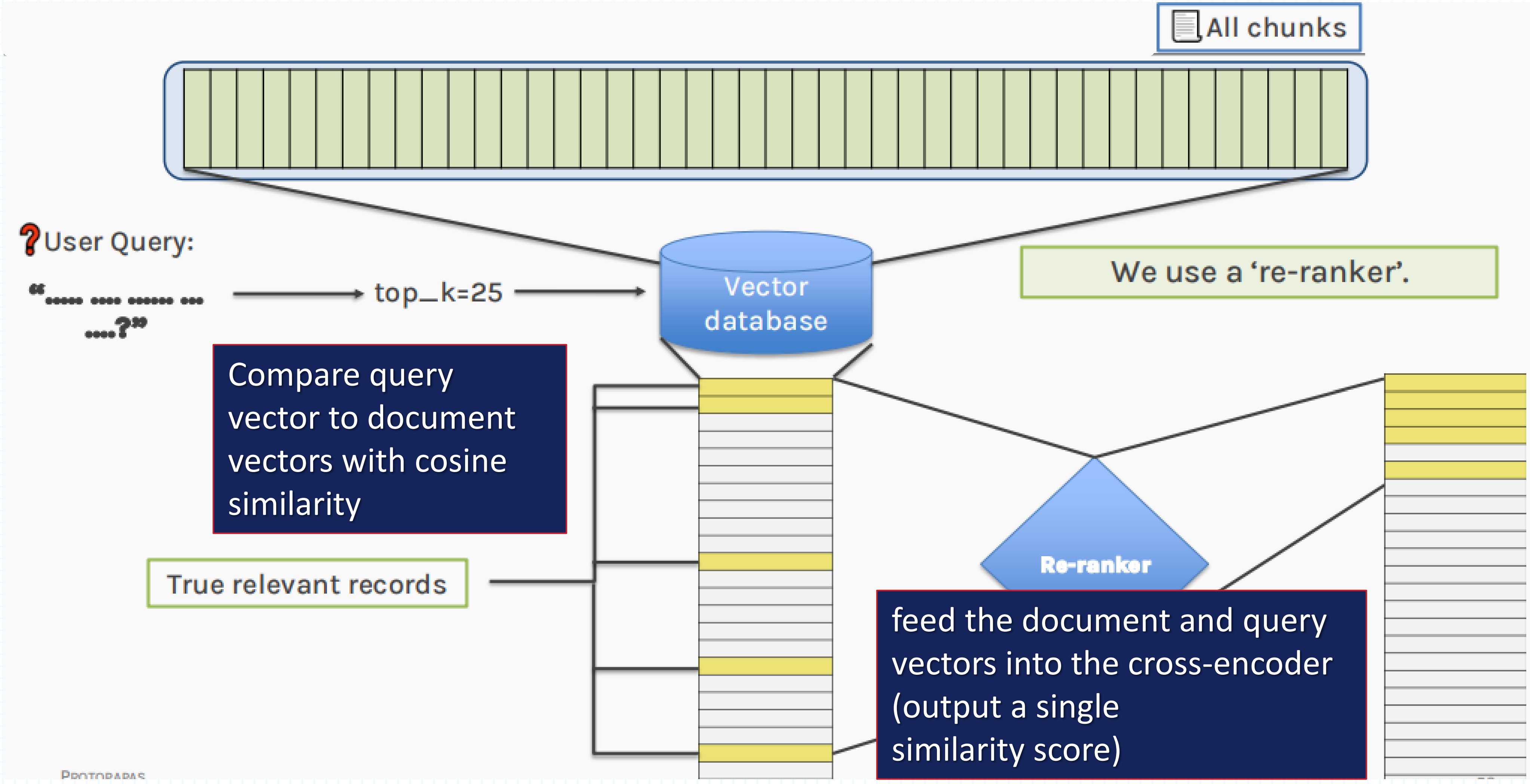
Naïve RAG



The Pre-Retrieval Phase deals with:
- Chunking the data
- Converting the chunks into embeddings
- Handling the embeddings

The Post-Retrieval phase deals with polishing what was obtained from the retriever.

Post-Retrieval – Re-ranking



Post-Retrieval – Re-ranking

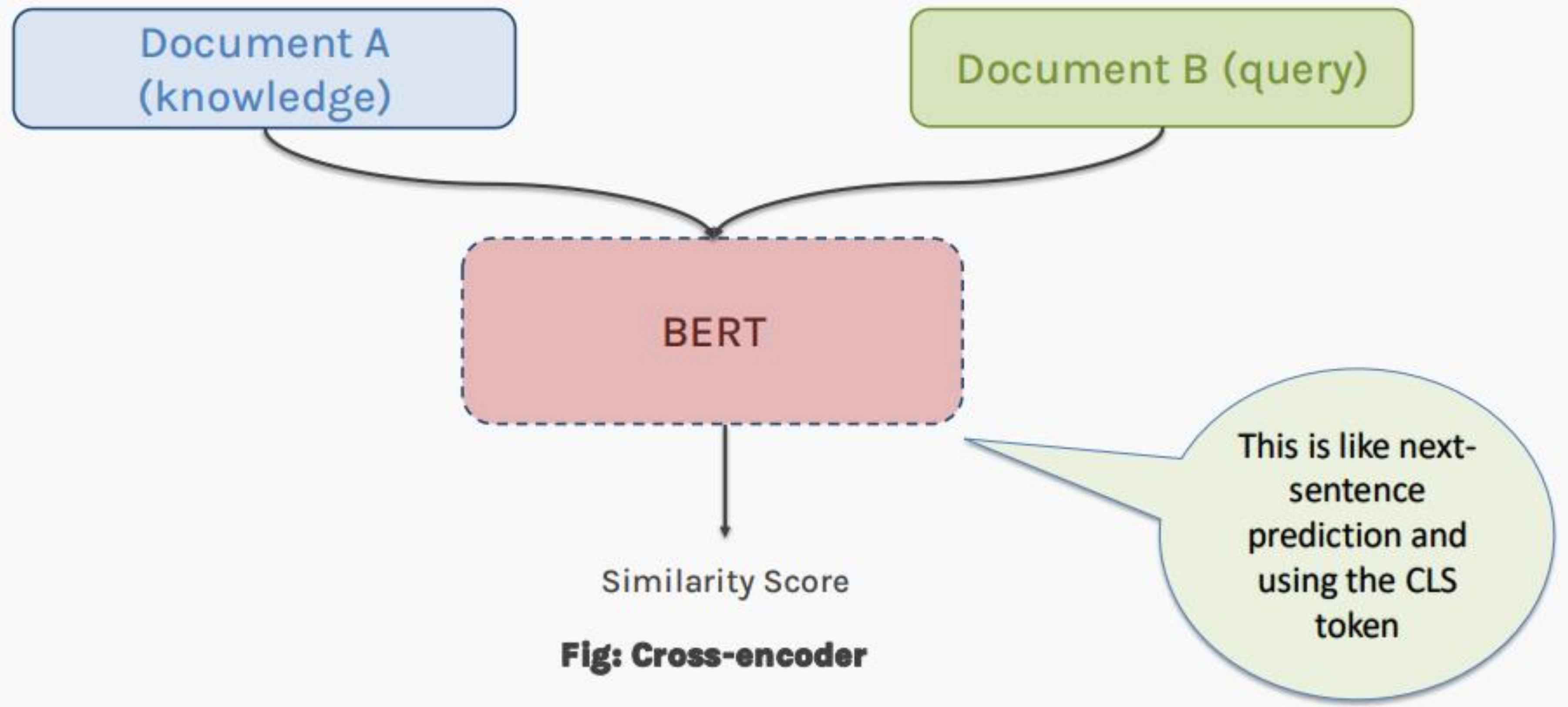


Fig: Cross-encoder