Summer School Week6

洪晙宸

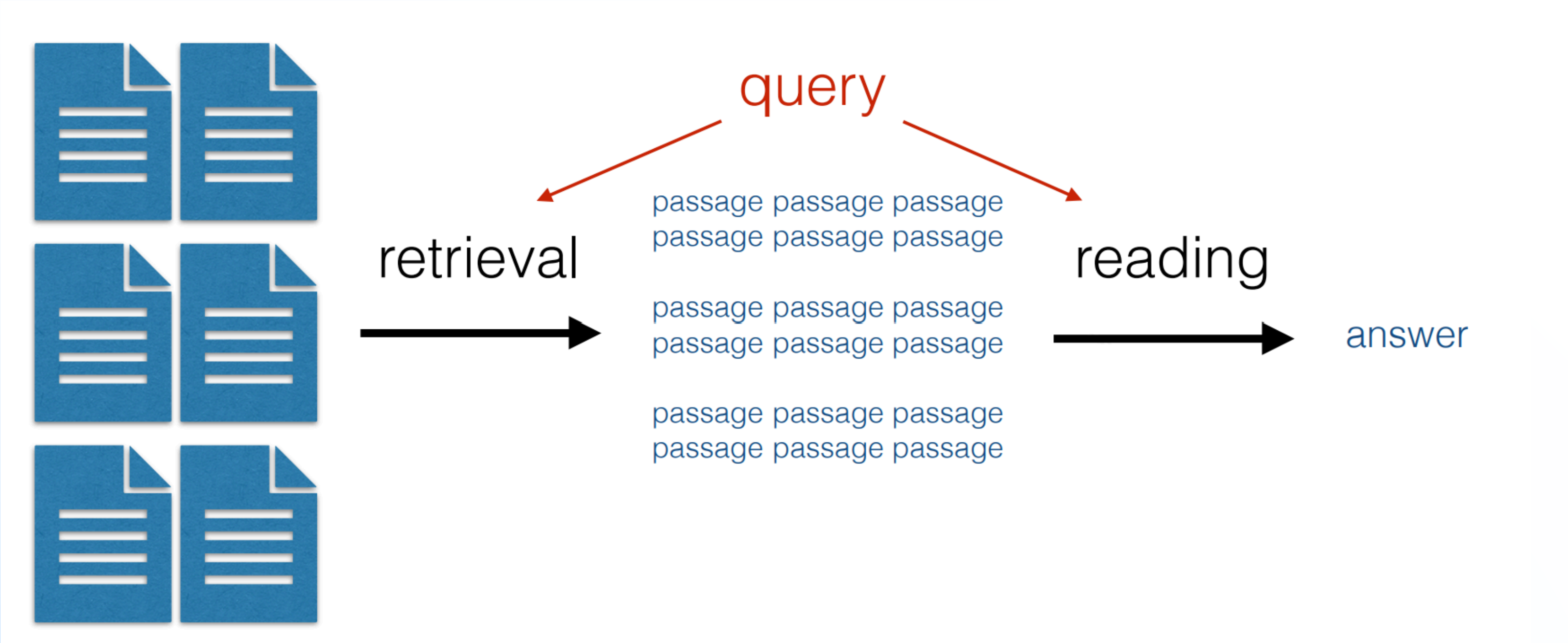
Slide here!

Retrieval and Retrieval Augmented Generation

Problem with standard prompting

- Accuracy issues:
    - **Knowledge cutoffs**: parameters are usually only updated to a particular time
    - **Private data**: data stored in private text or data repositories not suitable for training
    - **Learning failures**: even for data that the model was trained on, it might not be sufficient to get the right answer
- Verifiability issues: It is hard to tell if the answer is correct

RAG



- Retrieve relevant passages efficiently
- Read the passages to answer the query

Retrieval

- Sparse retrieval
- Document-level dense retrieval
- Token-level dense retrieval
- Cross-encoder reranking
- Black-box retrieval(ask Google)

Sparse Retrieval

- Some terms are more important than others => **Low-frequency words** are often more important
- Term Frequency - In-Document Frequency (TFIDF)

$$\mathrm{TF}(t,d) = \frac{\mathrm{freq}(t,d)}{\sum_{t'} \mathrm{freq}(t',d)} \qquad \mathrm{IDF}(t) = \log\left(\frac{|D|}{\sum_{d' \in D} \delta(\mathrm{freq}(t,d') > 0)}\right)$$

$$\text{TF-IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t)$$

- BM25: TF term similar to smoothed count-based LMS

$$\text{BM-25}(t,d) = \mathrm{IDF}(t) \cdot \frac{\mathrm{freq}(t,d) \cdot (k_1 + 1)}{\mathrm{freq}(t,d) + k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)}$$

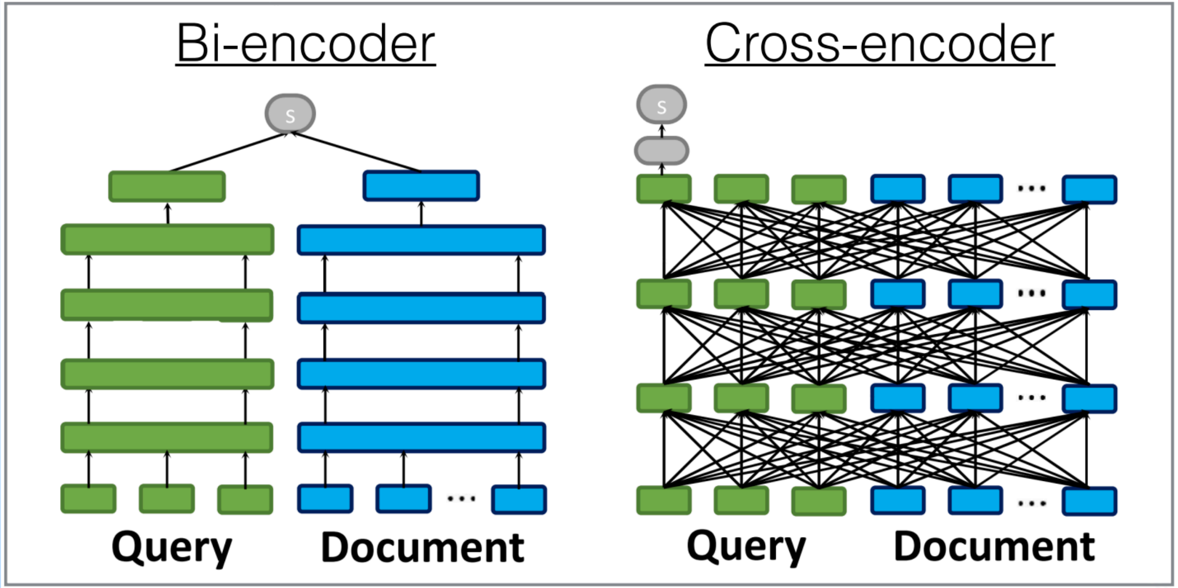
Dense Retrieval

- Encode document/query and find nearest neighbor
- Using
  - Out-of-box embeddings
  - Learned embeddings
    - Select positive and negative documents, train using a contrastive loss (e.g. hinge loss). ex: DPR, Contriever

$$\mathcal{L}(\theta, q) = \sum_{d_{\text{pos}} \in D_{\text{pos}}} \sum_{d_{\text{neg}} \in D_{\text{neg}}} \max(0, s(q, d_{\text{neg}}; \theta) - s(q, d_{\text{pos}}; \theta))$$

Cross-encoder Ranking

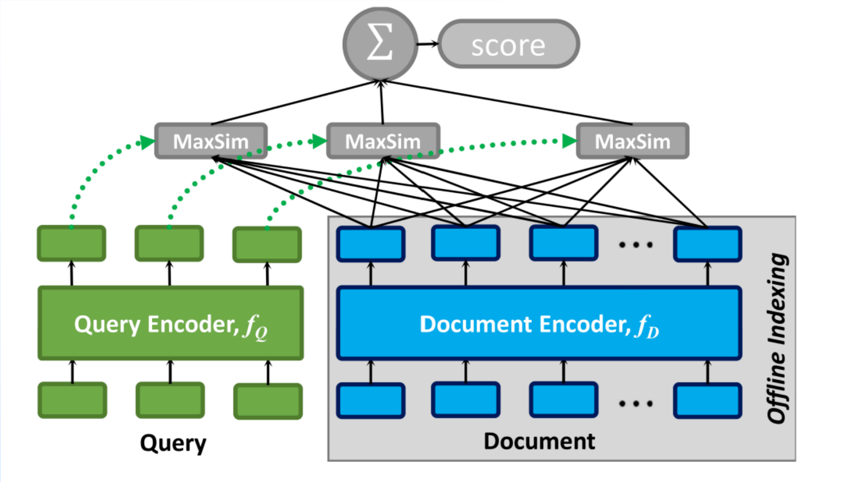
- Jointly encode both queries and documents using neural model



- Precludes ANN lookup, so can only be used on small number of candidates

Token-level Dense Retrieval

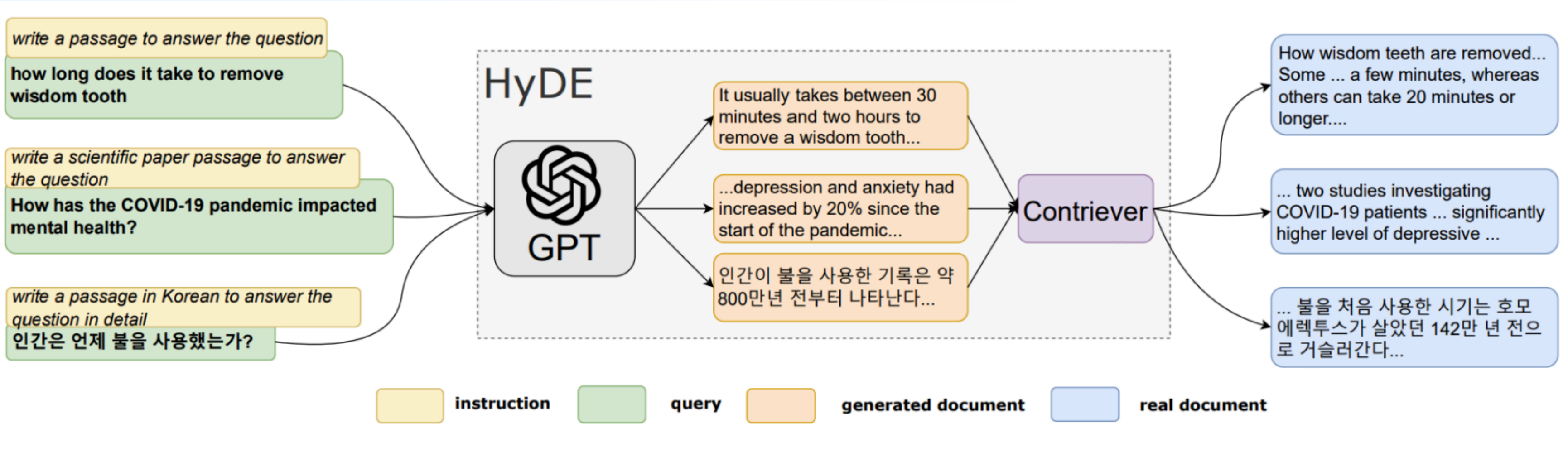
- **ColBERT** use contextual representations of all query and document tokens to compute retrieval score.



- Significantly more effective (but more costly) than single-vector retrieval

Hypothetical Document Embeddings

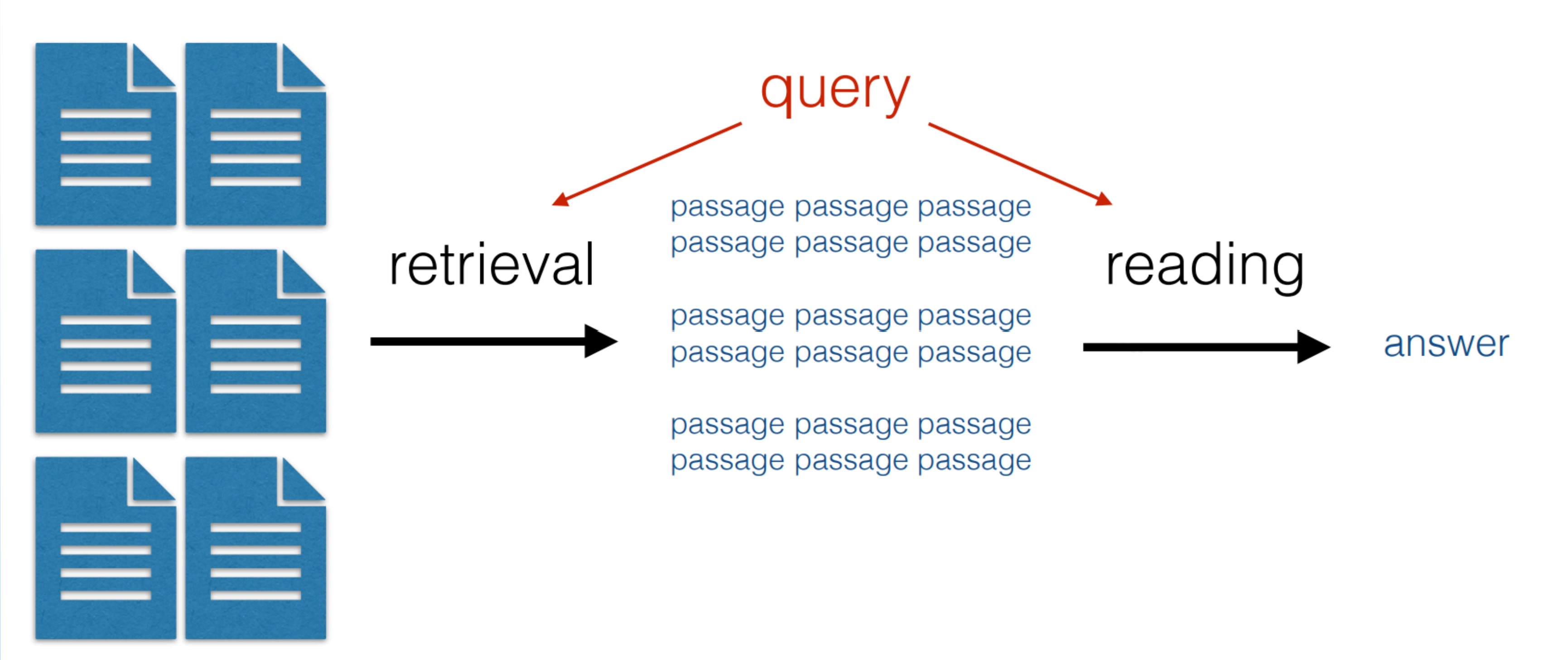
- Generate a **hypothetical document** for the query using an LLM, and try to look it up
- Can be easier than trying to match under-specified query
- But it needs more genertion time



Retriever-Reader Model

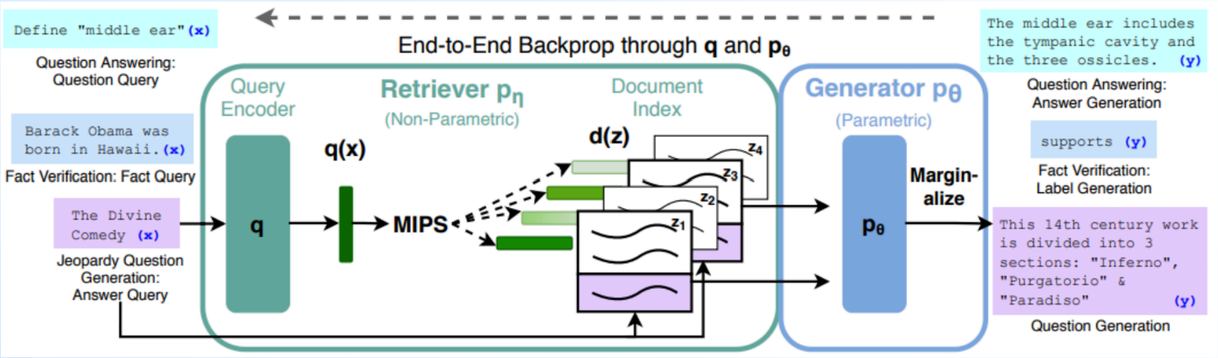
Simple Implement

- Simply add passage in prompt



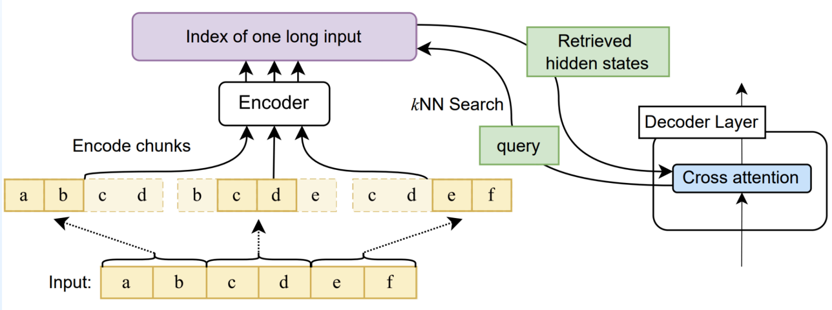
Retriever + Generator End to End Training

- Train the retriever and reader to improve accuracy
  - **Reader**: Maximize generation likelihood given single retrieved document
  - **Retriever**: Maximize overall likelihood by optimizing mixture weights over documents
- Problem: search index becomes stale



When retrieve

- Once, before generate: used by most
- Several times during generation, as necessary
  - Trigger retrieval with specific token, ex: `WikiSearch(...)`
  - Trigger retrieval when uncertainly: `FLARE` trie to generate, and retrieve when LM certainly is low
- Every token
  - Token-level softmax modification: `KNN-LM` retrieves similar examples, and uses the following token from them
  - Token-level Approximate Attention



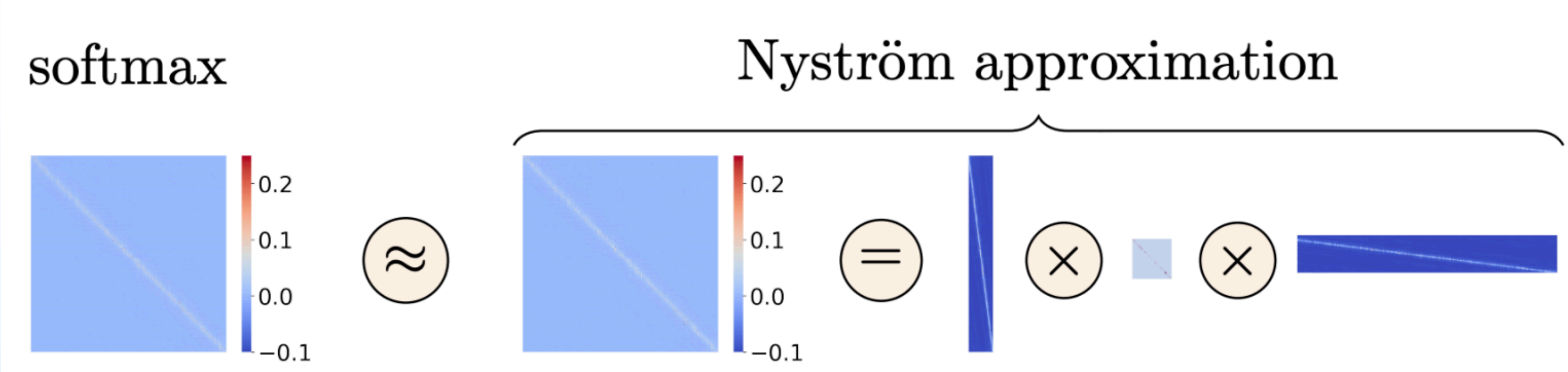
Long Context Transformer

Training Transformer over longer sequences

- Simply use all previous wrds in document
  - But the computation is quadratic in sequence length
- In RNN, we can truncate backdrop
- In Transformer-XL, we can use the previous state but not backprop them. Or `sliding window attention` in Mistral
- In sparse transformer, Added `stride`, only attending to every n previous states

Low-rank Approximation

- Project matrix to lower dimension to avoid heavy computation



Effectively Using Long Contexts

- As context increase, the model pay less attention to things in the middle of the context
- Ensuring Use of Relevant Context: distill content

RAG demo

- This is my own project, which use RAG to generate a class schedule.