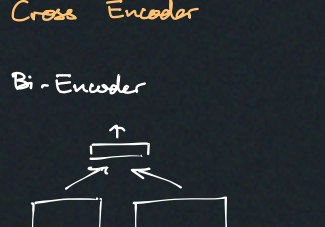
Lec 10 Retrieval & Retrieval-Augmented Generation (RAG)

Problem with LM generation
- Knowledge out of date
- LM not trained on private data
- Learning failure: may be wrong even on training data
- Output not verifiable

RAG



Retrieval

▷ Sparse retrieval

query & docs both as word freq vector (len-norm count) then take inner product

Problem: high freq words like "we", "is" not informative

▷ Term weighting: make low-freq words more important

$$TF(t,d) = \frac{freq(t,d)}{\sum_{t'} freq(t',d)}$$ (term freq: t; term: t; doc: d)

$$IDF(t) = \log\left(\frac{|D|}{\sum_{d \in D} \delta(freq(t,d)>0)}\right)$$ (in-doc freq: t)

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

→ BM25 – similar but with additional smoothing

▷ Inverted index – map token to doc

▷ Dense Retrieval

doc, query into dense emb, find nearest nbor

The embedding:
- Out of the box ... sentence transformer ← emb tends to encode more high-freq
- Specialised embs ← usually better at low-freq upweight

Learning emb: pick pos & neg examples, train with contrastive loss

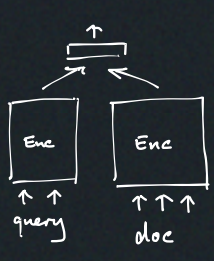
→ DPR: BM25 based, use both hard negs & in-batch negs
  └ seems right but wrong

→ Contriever: contrastive learning on rand. spans as pos

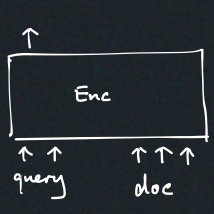
Problem: dot prod btwn query & every doc is costly
Sol: Approximate nearest nbor search
→ Locality-sensitive hashing. Divide cont. space into regions. Use index to find region
→ Graph-based search. Make hubs in graph
Software
→ FAISS, ChromaDB

▷ Cross Encoder

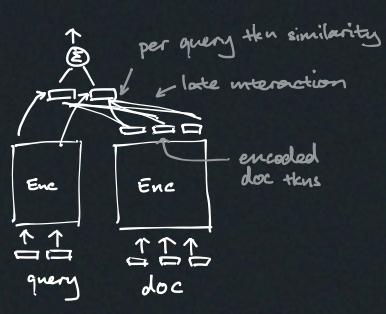
Bi-Encoder



Cross-Encoder ← Good acc



ColBERT



▷ Hypothetical Doc Emb

Idea: easier to match doc against doc

query → (LLM) → hypothetical doc ↔ (retrieve) docs

what a retrieved doc should look like?

Retriever-Reader Models

▷ Chaining ← works pretty well

Just feed retrieval output to GPT

▷ Retriever Generator End-to-end

Reader: maximise likelihood for single retrieved doc
Retriever: maximise overall likelihood over docs
sum probs weighted by doc score
viz. linear interpolation across docs

- No need for labels, end2end
- Search index needs to be updated because of doc emb updates

▷ Multi-round Retrieval

Can retrieve again if
- Model outputs search token
  train: if search tkn led to success, good
- Model is uncertain
  FLARE: when model outputs sentence with low prob, search, regenerate
- Every token
  kNN-LM: retrieve similar examples, use their next tkn (interpolate with model probs)

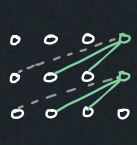
Unlimiformer: sliding window on input, attention to retrieved

Long-context Transformers

Naïve: concat all docs ← quadratic on Transformer

RNN can do infinite context when inference
can do truncated backprop through time (BPTT) to train

→ Truncated BPTT-Transformer



Not prop too far before

→ Sparse Transformer (Child et al. 2019)

→ Compress previous states

→ Low-rank Approx ← attention has big matrix approximate by smaller one

Long-context model benchmarking

→ Long Range Arena
→ SCROLLS

Problem: Context models tend to miss info in middle

Context relevance
→ Filter
→ Remove irrelevant info in doc before feeding into reader