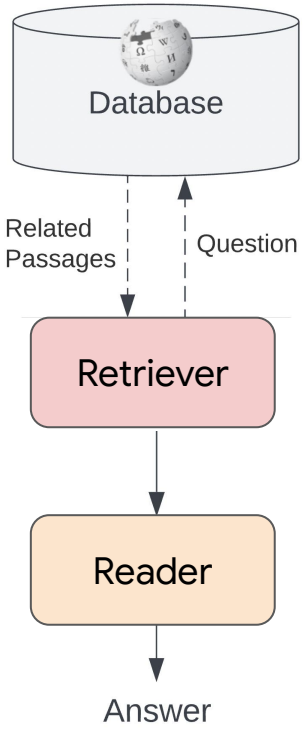
DevRev's Expert Answers in a Flash: Improving Domain-Specific QA

Primary ID: 13
Secondary ID: 26

Domain Specific Question Answering



- Information Retrieval: Identify potential candidate contexts from a large knowledge base.
- Extractive Question Answering: Extract (one) specific answer span to given query question.

Existing Work

Text Retrieval

- Dense, Sparse or Hybrid representations.
- Recent methods employ,
  - complex transformer models
  - expensive fine-tuning techniques
  - instruction-tuned LLM prompting

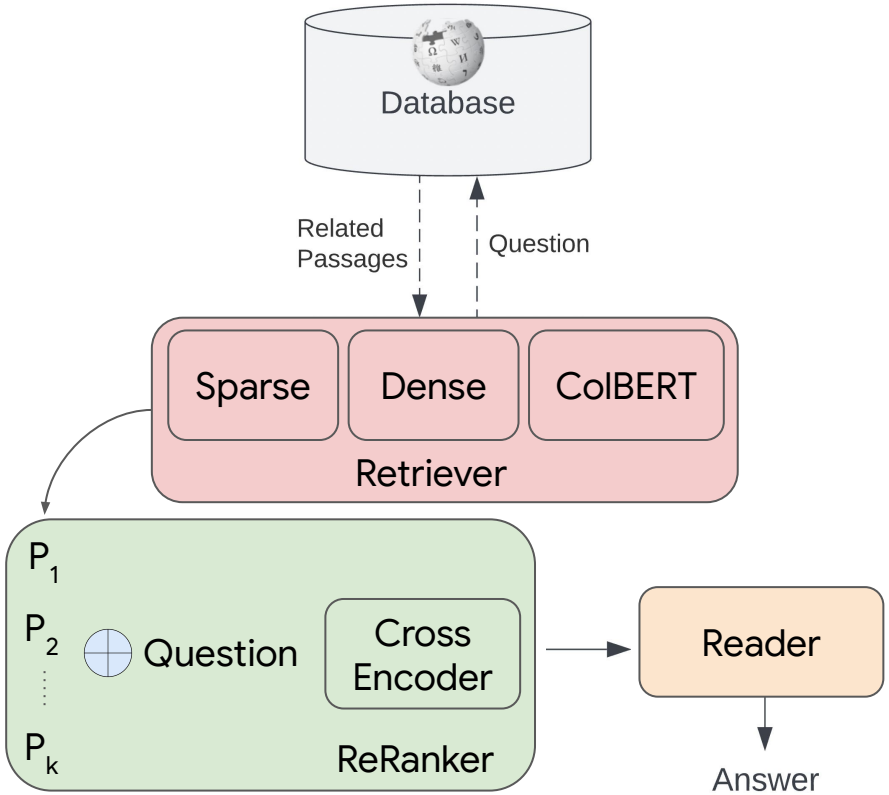
Extracting Question Answering

- Ensembles like IENet and Retro-Reader
- Recent methods employ,
  - ensembling high performing diverse LLMs
  - multi-stage reading to predict candidates
  - span-based entity recognition

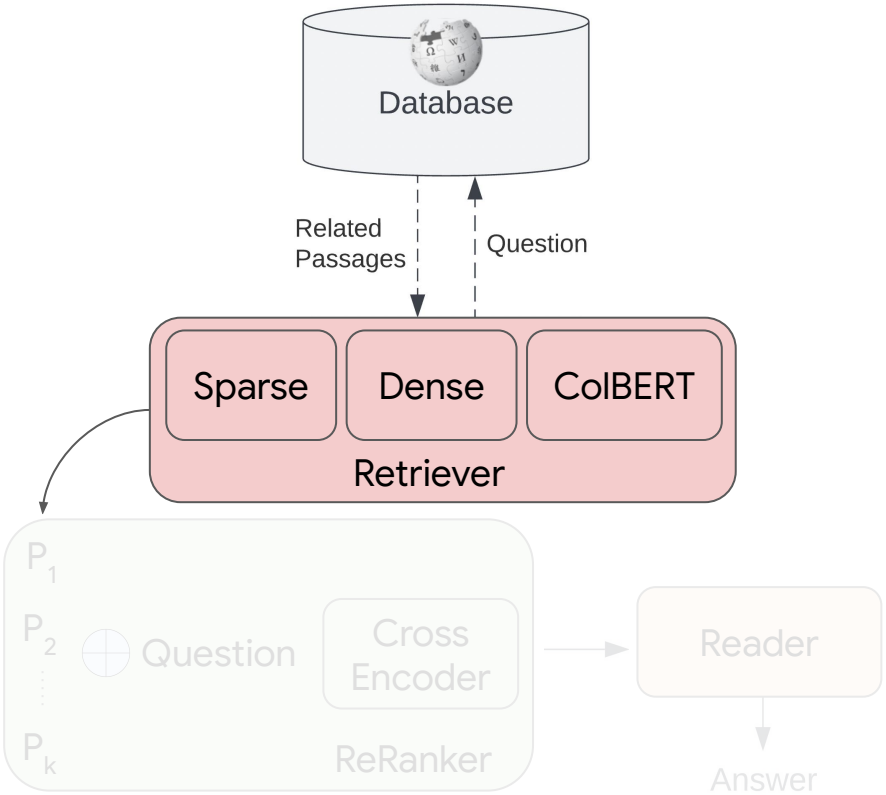
Existing Work

- Although these methods push SOTA, they still leverage computationally expensive strategies at several stages, unsuitable for real-time deployment.
- We argue that more simpler strategies (e.g BM25, DPR, etc) guarantees efficiency and when used appropriately can showcase competitive performance.

Retrieve Thrice Rank and Answer (R3RA)



Retrieve Thrice



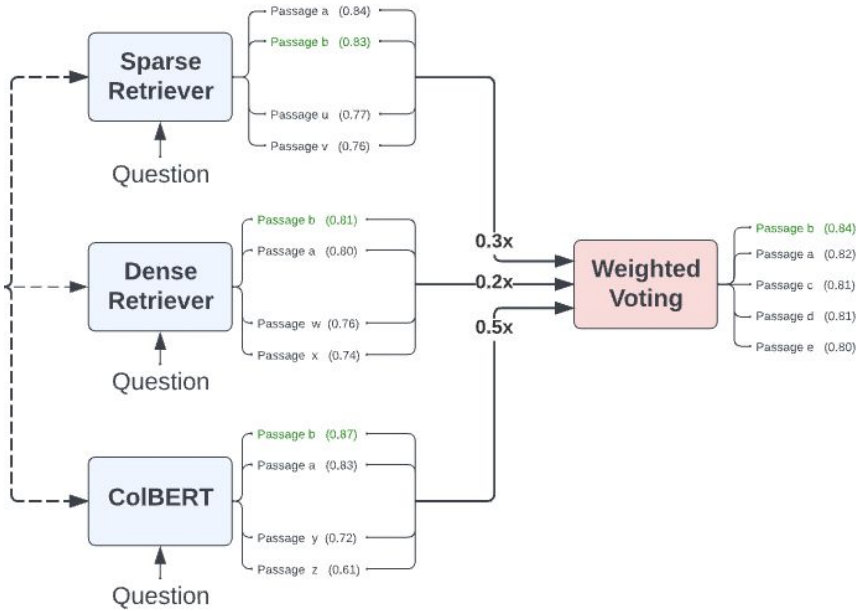
We make use of two popular retriever embedding strategies:

- BM25: focuses on lexical similarity.
- Dense Retriever: focuses on complex semantic relations.

An ideal retriever must use both.

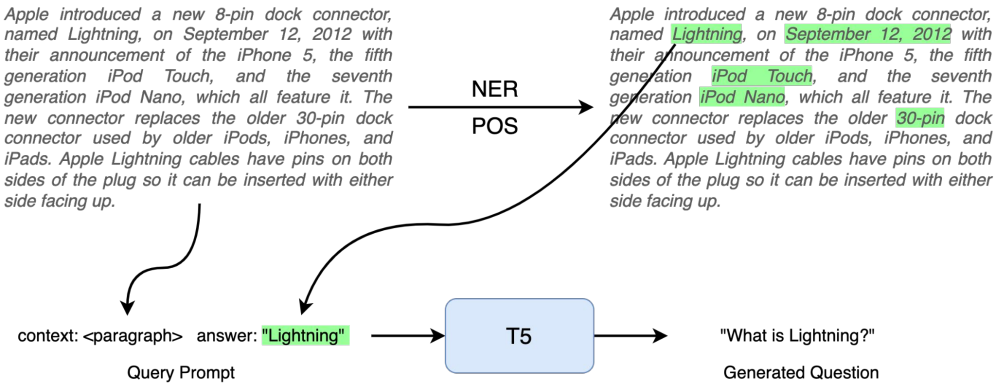
- smaller, simpler passages can be retrieved by exact match.
- overall semantic meaning by dense embeddings.

Ensembled Voting



To make best use of all worlds, we employ a carefully designed voting mechanism by retrieving closest matching paragraphs from each module, and their corresponding confidence scores to estimate a voted aggregation of all.

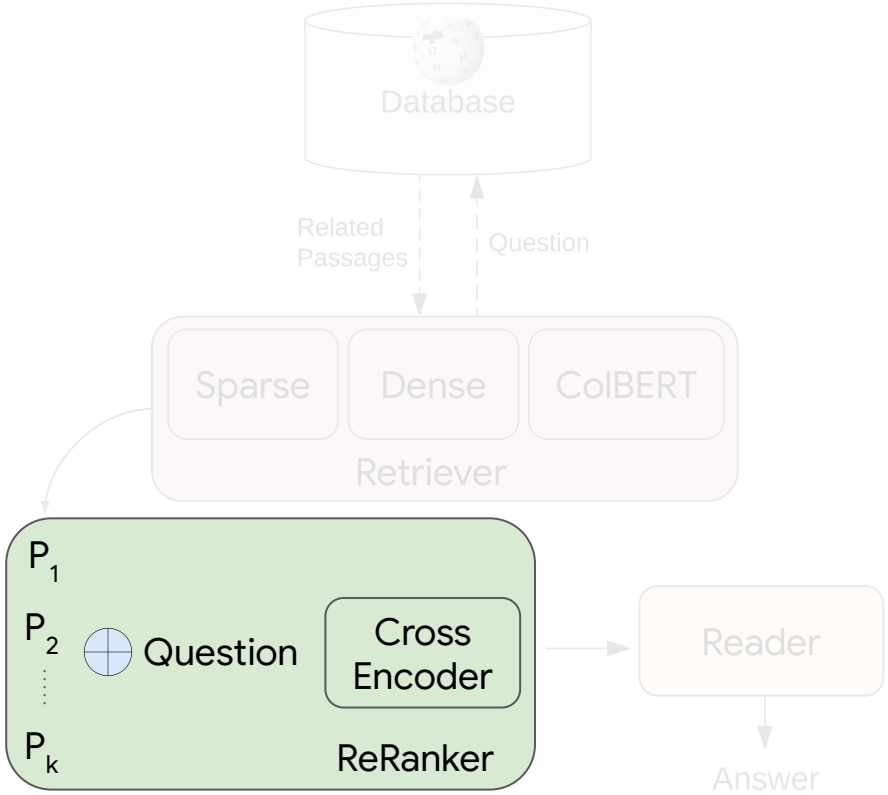
Automatic Question Generation



High capacity feature encoders might still fail to correlate abstract questions with their related paragraphs, either due to (i) inability to hallucinate answerable questions, or (ii) size of paragraph tokens.

Hypothesize that by identifying potentially answerable questions for each paragraph, one can reuse them for effective retrieval.

Rank (R)



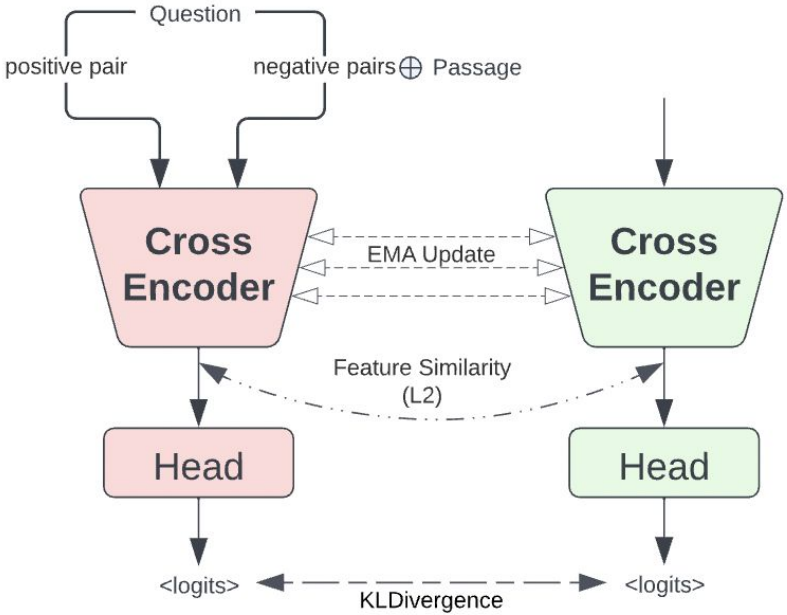
- (Optionally) re-rank the retrieved paragraphs using a more complex encoder that captures fine-grained information.
- A one-stage retriever (using just a ranker), would significantly increase computational overhead.

I: Fast and Effective Retriever
to identify potential candidates

↓

II: Fine Grained Retriever
to find closest match

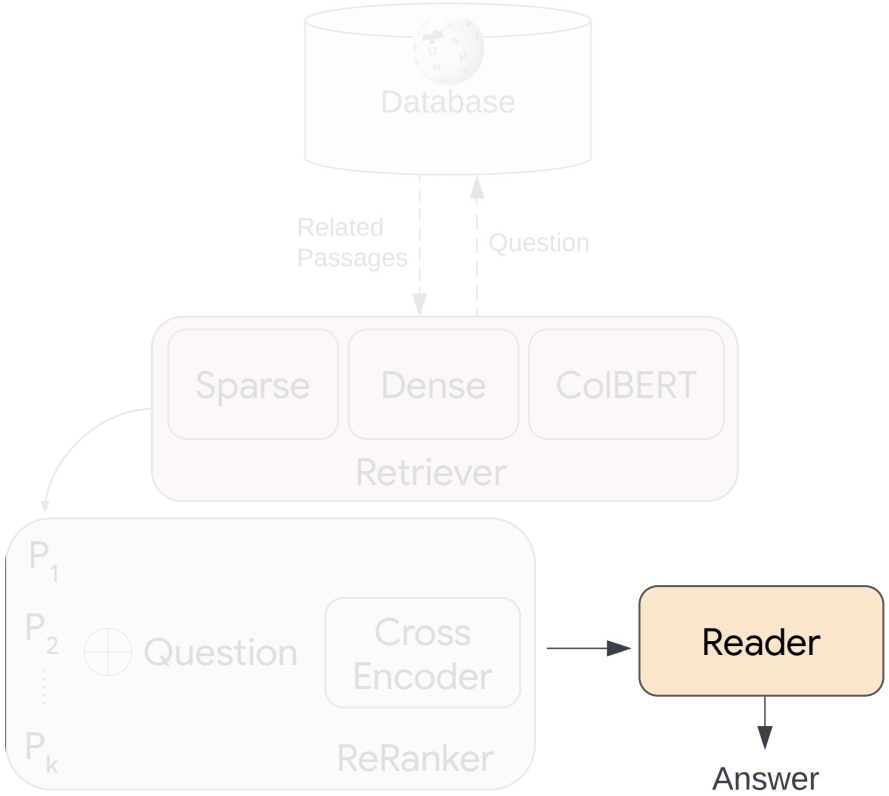
Self Distill And FineTune (SDAFT)



- Given a knowledge base categorized into a fixed set of themes, train theme-specific cross encoders.
- Fine Tuning Strategies:
  - All layers: non-trivial optimum in the case of domain drifted data.
  - Linear classifier: not the best solution.

- SDAFT, combines self-distillation and fine-tuning to preserve knowledge from the pretrained model during transfer learning on the downstream task.

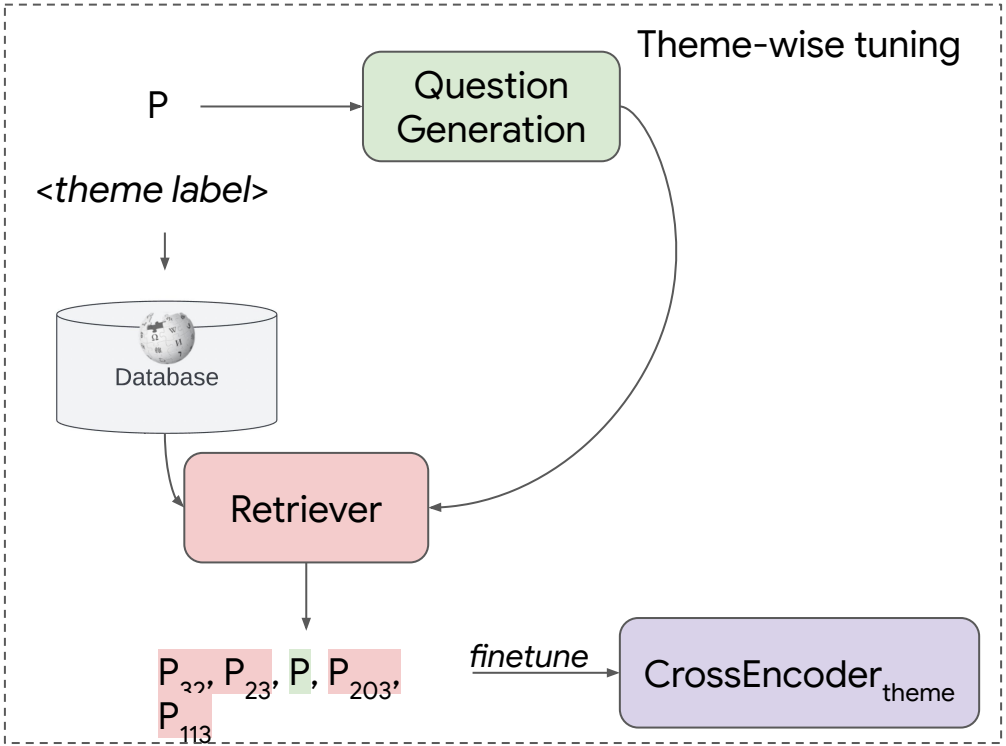
Answer (A)

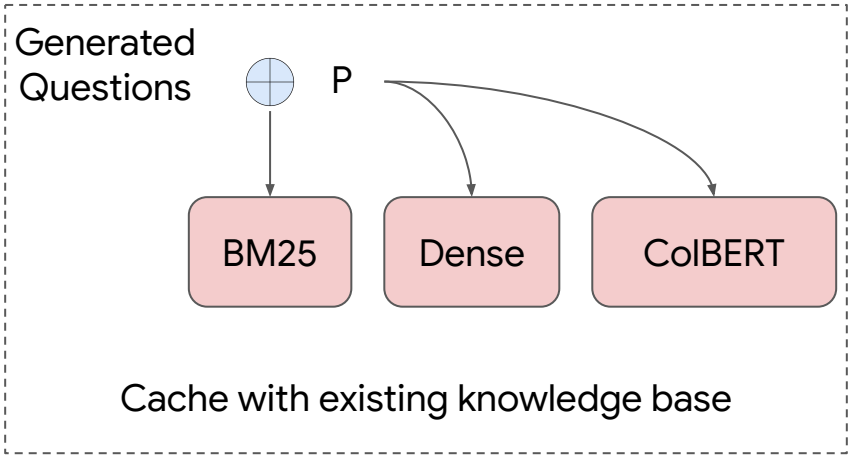


- Using off-the-shelf pretrained QA models, we obtain the start and end indices of the required answer span.

R3RA Pipeline Preparation

Given an incoming paragraph,





Results

Method	Top-1	Top-5	F1
Sparse BM25	0.701	0.99	-
Dense MPNet	0.730	1.00	-
ColBERT	0.826	1.00	-
w/ Voting Ensemble	0.834	1.00	-

Our voting strategy outperforms individual retrievers, achieving **perfect top-5 score** for reranking.

Results

Method	Top-1	Top-5	F1
Sparse BM25	0.701	0.99	-
Dense MPNet	0.730	1.00	-
ColBERT	0.826	1.00	-
w/ Voting Ensemble	0.834	1.00	-
w/ Voting Ensemble (questions appended)	0.854	1.00	-

Augmenting the retrieval stage with generated questions further improves performance by **2%**

Results

Method	Top-1	Top-5	F1
Sparse BM25	0.701	0.99	-
Dense MPNet	0.730	1.00	-
ColBERT	0.826	1.00	-
w/ Voting Ensemble	0.834	1.00	-
w/ Voting Ensemble (questions appended)	0.854	1.00	-
w/ QA Reader			
deepset/tinyroberta-squad2	-	-	0.794

Without reranking, we obtain **79 %** F1 score using off-the-shelf pretrained QA models.

Results

Method	Top-1	Top-5	F1
Sparse BM25	0.701	0.99	-
Dense MPNet	0.730	1.00	-
ColBERT	0.826	1.00	-
w/ Voting Ensemble	0.834	1.00	-
w/ Voting Ensemble (questions appended)	0.854	1.00	-
w/ QA Reader			
deepset/tinyroberta-squad2	-	-	0.794
w/ Re Ranker			
cross-encoder/ms-marco-MiniLM-L2	0.849	-	-
cross-encoder/ms-marco-MiniLM-L2 (fine-tuned)	0.871	-	-
w/ QA Reader			
deepset/tinyroberta-squad2	-	-	0.817

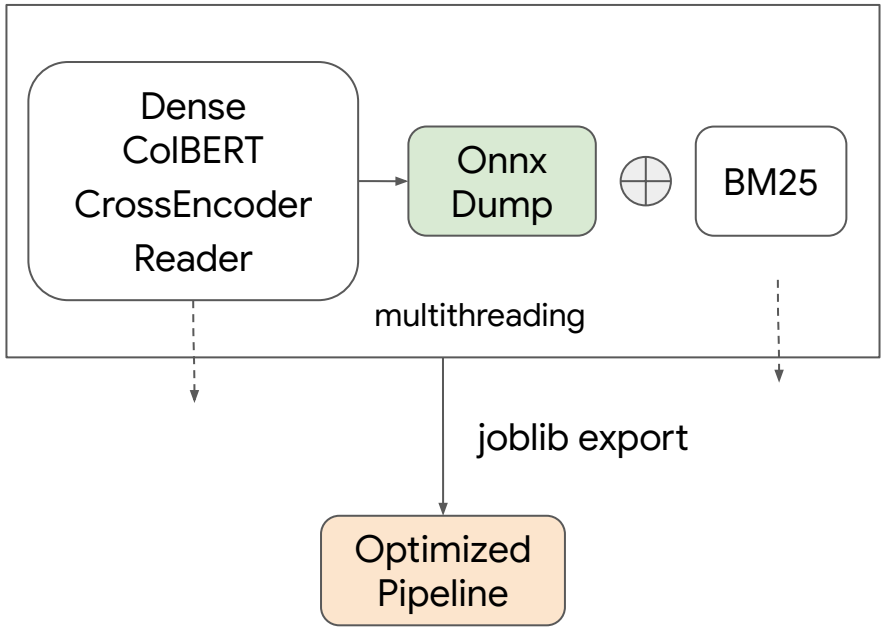
We observe that naively plugging in pretrained cross-encoders **hurts** top-1 retrieval

Results

Method	Top-1	Top-5	F1
Sparse BM25	0.701	0.99	-
Dense MPNet	0.730	1.00	-
ColBERT	0.826	1.00	-
w/ Voting Ensemble	0.834	1.00	-
w/ Voting Ensemble (questions appended)	0.854	1.00	-
w/ QA Reader			
deepset/tinyroberta-squad2	-	-	0.794
w/ Re Ranker			
cross-encoder/ms-marco-MiniLM-L2	0.849	-	-
cross-encoder/ms-marco-MiniLM-L2 (fine-tuned)	0.871	-	-
w/ QA Reader			
deepset/tinyroberta-squad2	-	-	0.817

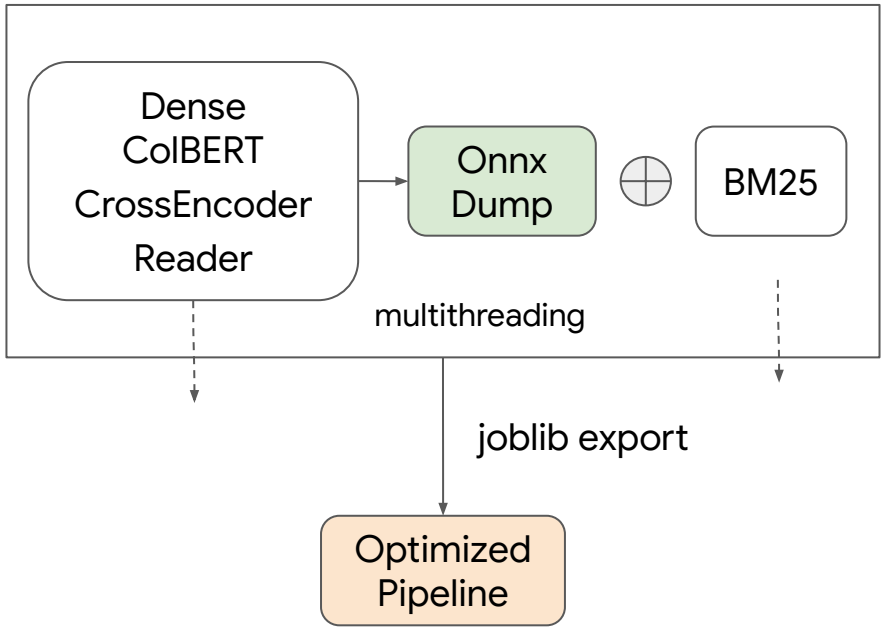
Our theme-wise fine tuned cross encoders obtain best retrieval performance (**2 % ↑**), hence improving F1 scores by **2 %**

Latency Improvements

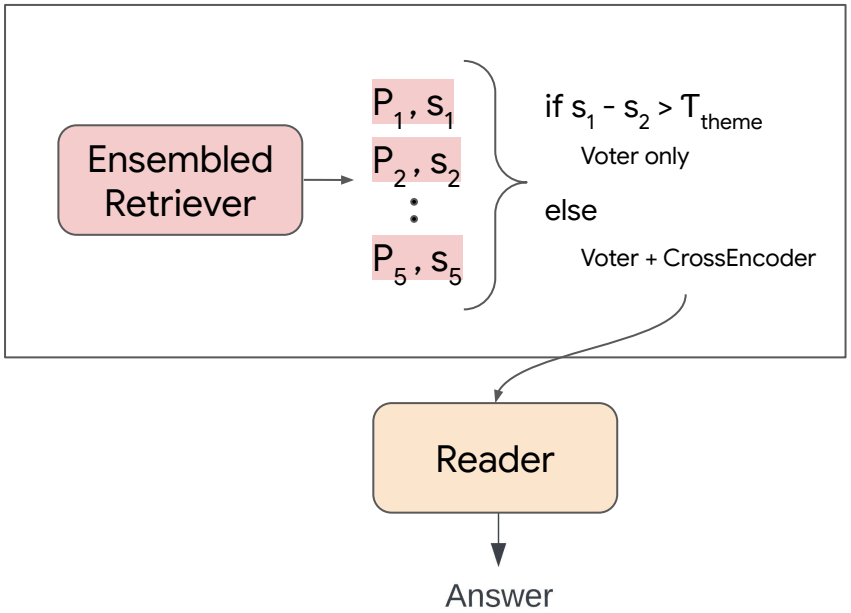


Heavily abstracted implementation, with several tricks to improve efficiency during inference.

Latency Improvements



Heavily abstracted implementation, with several tricks to improve efficiency during inference.



Our adaptive reranking strategy enables us to make best use of efficient stage I retrieval, and fine-grained stage-II reranker

Results: Latency

Method	Latency (ms) w/o Threading Retriever	QA Reader	Overall	w/ Threading Overall
R2RA (w/o Ranker)	167.014	512.401	679.415	455.316
R2RA (w/ Ranker)	215.461	512.401	727.86	487.588

Our overall choice of latency tricks, enables our entire pipeline (with or without reranker) to efficiently answer a given query in **under 50 %** of the given time.

Conclusion

- We propose R3RA, a fast and effective three stage approach for domain-specific question answering.
- Our ensemble voting strategy enables us to use simpler sparse, dense encoding strategies while maintaining competitive performance.
- Further, we leverage a fully-automatic question generation pipeline to identify potentially answerable questions to improve retrieval performance.
- We propose SDAFT, a novel fine tuning mechanism that preserves knowledge from the pretrained initialization during transfer learning on the downstream task.
- Our overall pipeline achieves impressive retrieval and F1 scores while ensuring computational efficiency.