OPEN-DOMAIN QA

Problem Setting

context

question → Large doc. collection → answer

Extractive phrase
Sentence
Entire Passage

Datasets Commonly Used

- TriviaQA [Joshi et al., 2017] — Trivia questions — Web pages from BING search
- SearchQA [Dunn et al., 2017] — Jeopardy — Google search snippets
- Quasar-T [Dhingra et al., 2017] — Reddit — ClueWeb09
- Natural Questions [Kwiatkowski et al., 2019] — Google queries — Wikipedia pages in results

Dataset	Train	Val	Test
NQ	79,168	8,757	3,610
WebQ	3,417	361	2,032
TREC	1,353	133	694
TriviaQA	78,785	8,837	11,313
SQuAD	78,713	8,886	10,570

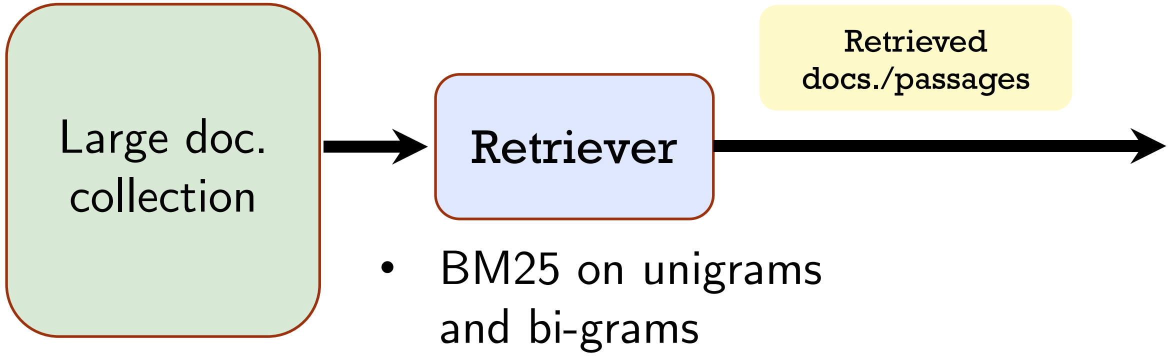
Repurposed for ODQA

- SQuAD [Rajpurkar et al., 2016]
- CuratedTREC [Baudis & Sedivy, 2015]
- WebQuestions [Berant et al., 2013]
- WikiMovies [Miller et al., 2016]

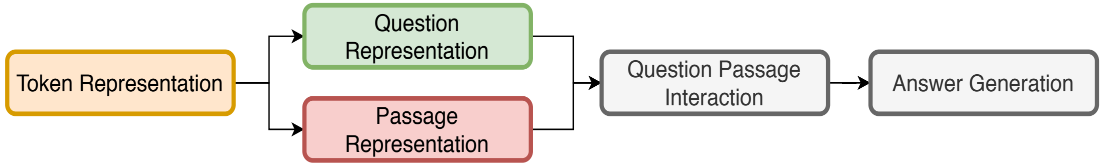
Metric Used

- Exact Match: measures whether the two strings, after preprocessing, are equal or not.
- F1 Measure: measures the overlap between the two bags of tokens in answers, after preprocessing
- Entity Match

Retrieve and Read



- BM25 on unigrams and bi-grams
- Vector Index



Use your favorite MRC model
- Attentive reader (DrQA)
- BiDaf (Seo 17)
- BERT (Bertserini)

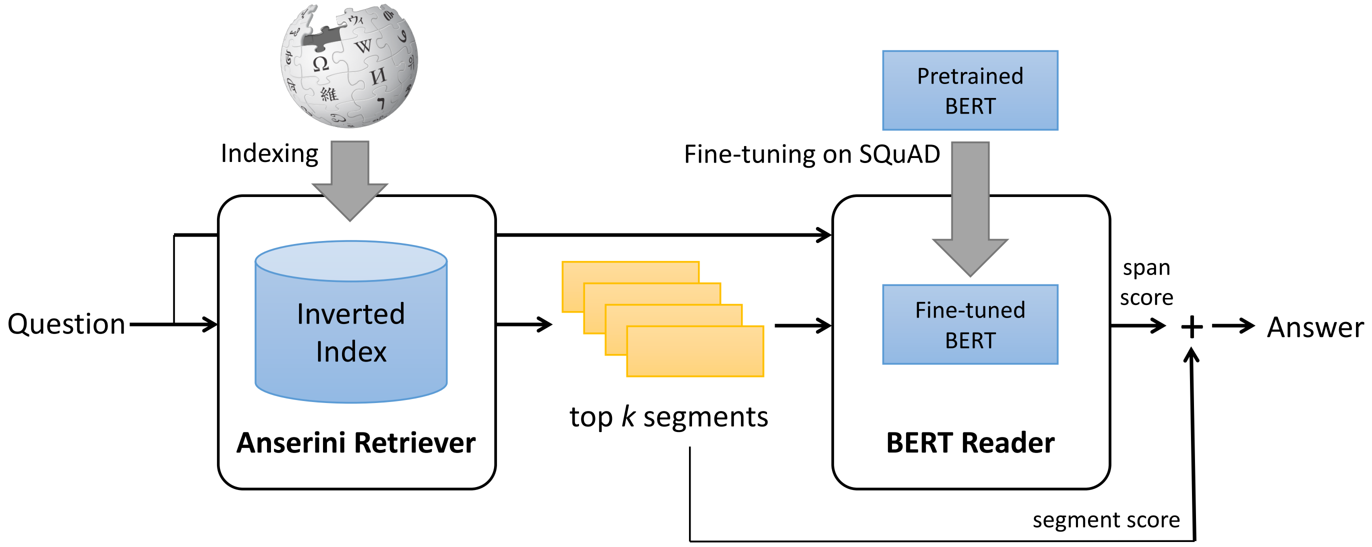
How is the reader model trained ?

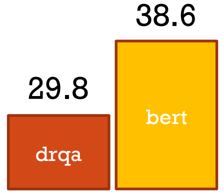
Using an existing QA dataset (e.g. SQUAD)

How does it answer questions ?

Independently find answers for tok-k passage and return the most "probable" span

BERTserini





Squad (EM)

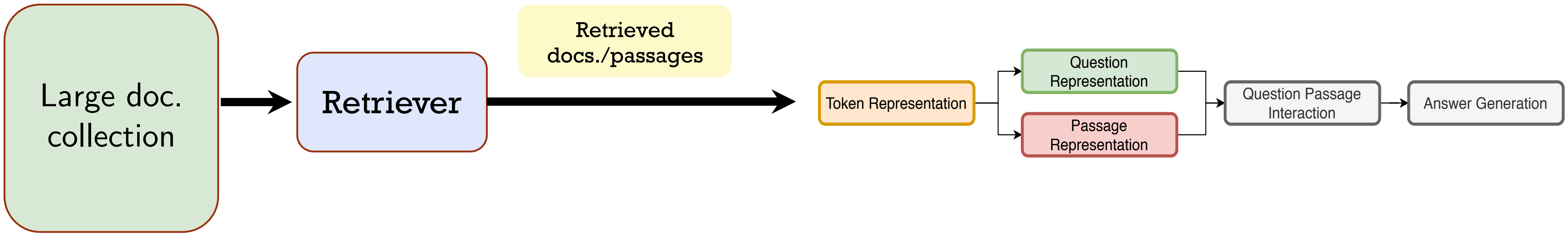
Retriever

- Using Anserini (based on Lucene)
- Segments = sentence/passage are indexed
- Retrieved sentences are scored using BM25

Reader

- Fine-tuned BERT on SQUAD
- Final score is interpolation of
  - Span score
  - BM25(segment)

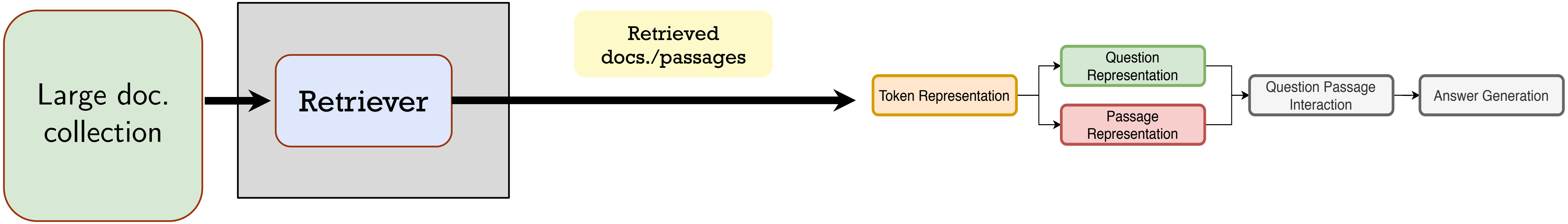
Design Questions



How do we aggregate evidence in retrieved passages ?

How do exploit the collection for a better reader model ?

How do we exploit reader state to re-retrieve more relevant passages ?



How Do We Aggregate Evidence In Retrieved Passages ?

Support

Question1: What is the more popular name for the londonderry air?

A1: tune from county

P1: the best known title for this melody is londonderry air - lrb- sometimes also called the **tune from county** derry -rrb- .

A2: danny boy

P1: londonderry air words : this melody is more commonly known with the words `` **danny boy** ''

P2: londonderry air **danny boy** music ftse london i love you .

P3: **danny boy** limavady is most famous for the tune londonderry air collected by jane ross in the mid-19th century from a local fiddle player .

P4: it was here that jane ross noted down the famous londonderry air -lrb- ` **danny boy** ' -rrb- from a passing fiddler .

Coverage

Question2: Which physicist, mathematician and astronomer discovered the first 4 moons of Jupiter

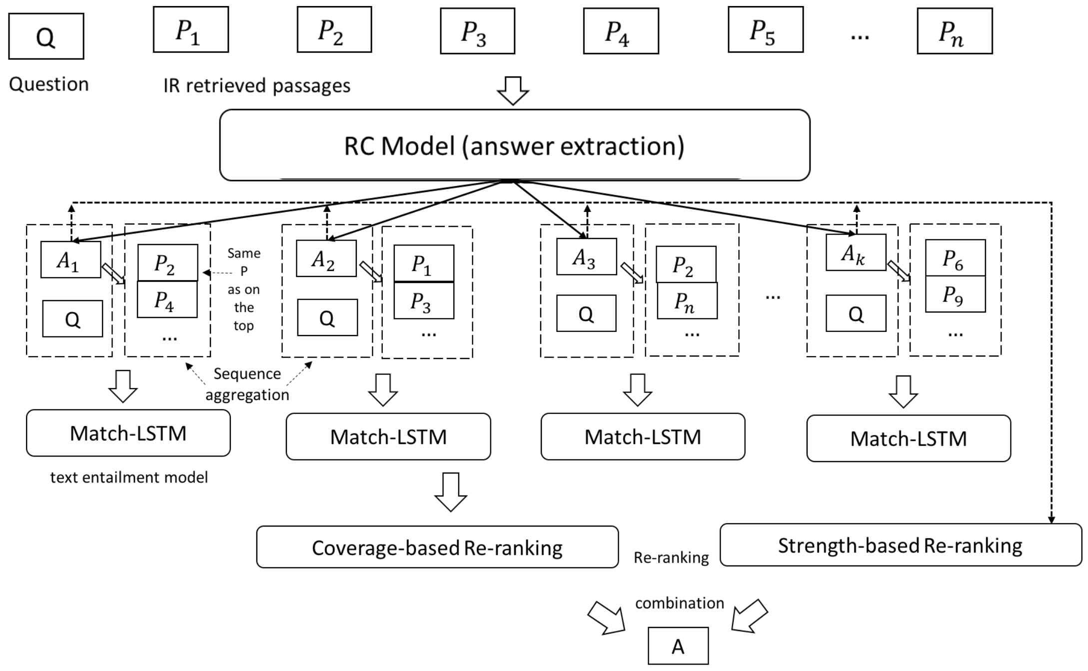
A1: Isaac Newton
P1: **Sir Isaac Newton** was an English physicist , mathematician , astronomer , natural philosopher , alchemist and theologian …
P2: **Sir Isaac Newton** was an English mathematician, astronomer, and physicist who is widely recognized as one of the most influential scientists …

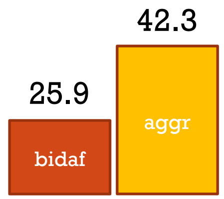
A2: Galileo Galilei
P1: **Galileo Galilei** was an Italian physicist , mathematician , astronomer , and philosopher who played a major role in the Scientific Revolution .
P2: **Galileo Galilei** is credited with discovering the first four moons of Jupiter .

Support And Coverage

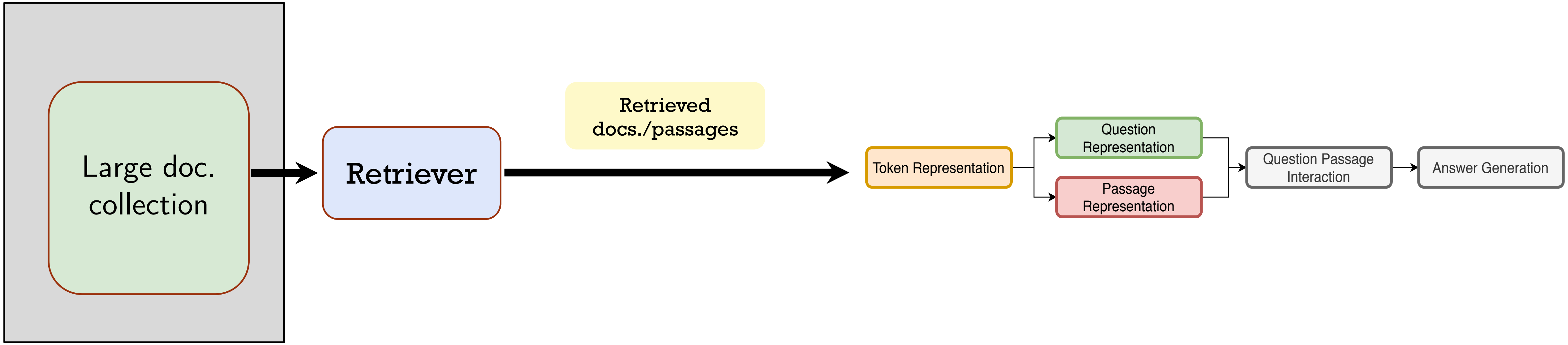
[Wang et al.' 18]

- For each candidate answer, re-rank retrieved passages based on
  - Support – counts
  - Coverage – attention mechanism





Quasar (EM)

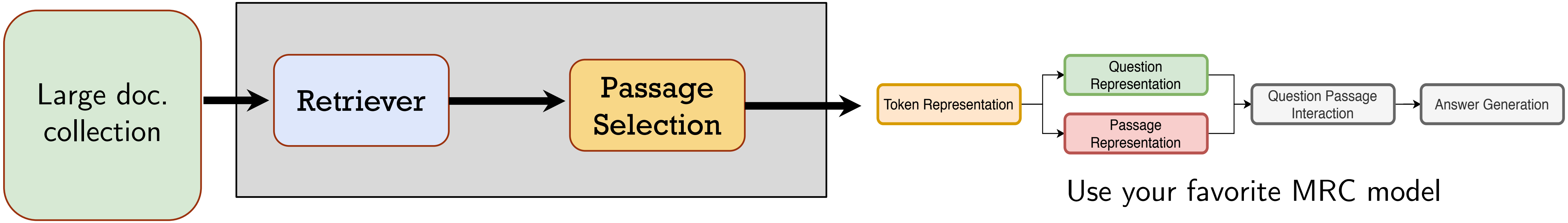


How do we Exploit Evidence from the collection ?

Extract Answers to a given Question In the large-scale un-labeled Corpus.

Distant Supervision

Exploit information about the question that is ignored in retrieved passages



BM25 on unigrams and bi-grams

- In MRC training data – (question, passage, answer)
- Distance Supervision [Chen et al.'17]
  - Create extra (question, passage, answer) triples
  - Simple Idea: Add all retrieved passages that mention the answer

Distant Supervision

- Add all retrieved passages that mention the answer

- Which passages to learn from ?
  - Liberal addition
    - All passages in the corpus containing answer added
    - All retrieved passages
  - Restrictive addition
    - Named entities constraints, passage length limits

- Noise in vanilla DS
  - Noise due to indiscriminate addition DSQA Model [Lin et al, '18]
  - Information loss due to filtered paragraphs DRQA [Chen '17]
  - Noise due to increasing collection sizes and retrieval depth [Kratzwald & Feuerriegel '18]

Distractors

Question: What is the capital of Ireland?

A: Dublin

P1: As the capital of Ireland, Dublin is …
P2: Ireland is an island in the North Atlantic…
P3: Dublin is the capital of Ireland. Besides, Ottawa is one of famous tourist cities in Ireland and …

- Key Idea: Select passages judiciously from the retrieved docs/passages

Selecting Passages

[Wang et al. '18]

Likelihood of the passage containing the answer

$$\Pr(a|q,P) = \sum_{p_i \in P} \boxed{\Pr(a|q,p_i)}\,\boxed{\Pr(p_i|q,P)}$$

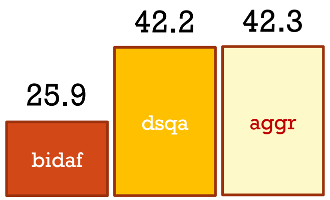
Likelihood of the answer given a cand. passage

Selecting Passages

$$\Pr(a|q,p_i) = P_s(a_s)P_e(a_e)$$

$$\Pr(a|q,P) = \sum_{p_i \in P} \boxed{\Pr(a|q,p_i)}\,\boxed{\Pr(p_i|q,P)}$$

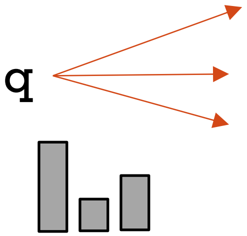
Question: What is the capital of Ireland?



Quasar (EM)

A: Dublin

q



P1: As the capital of Ireland, Dublin is ...

P2: Ireland is an island in the North Atlantic...

P3: Dublin is the capital of Ireland. Besides, Ottawa is one of famous tourist cities in Ireland and ...

Selecting Passages

$$\Pr(a|q,P) = \sum_{p_i \in P} \boxed{\Pr(a|q,p_i)} \boxed{\Pr(p_i|q,P)}$$

Answer Selection

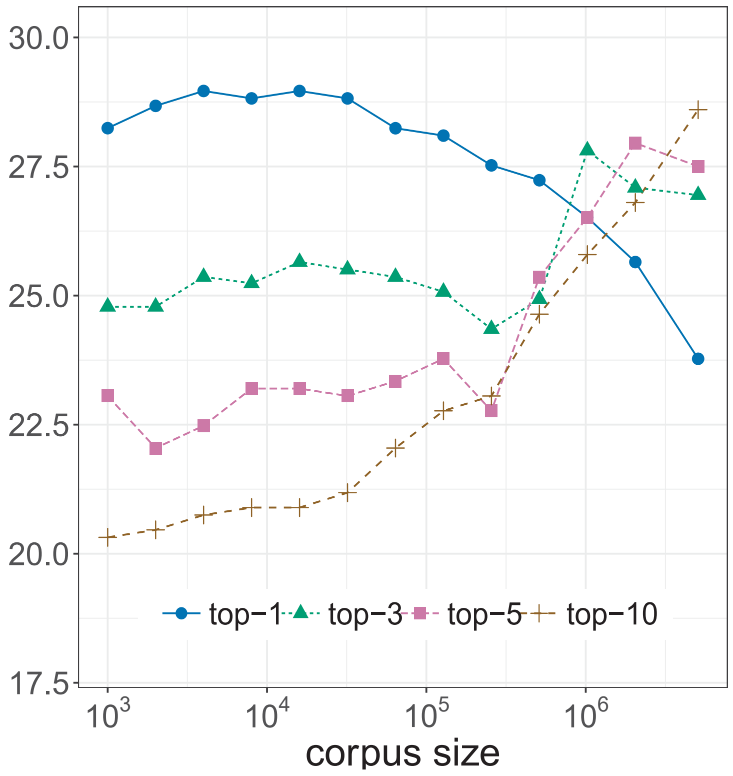
1. Detect spans for each passage
2. Multiple answers possible in a passage
3. Use the same rep. space for passage sel. and answer sel.

Passage Selection

1. Compute representations for query and passage independently
2. Compute relevance of passage to the query
3. Relevance is used as weights later

[Kratzwald & Feuerriegel ' 18]

Retrieval Depth and Collection Size



Large corpus = more noise

Idea: The more confident we are, the less we should retrieve

$$n_i = \max_k \sum_{j=1}^{k} s_i^{(j)} < \theta$$

Retrieved doc/passage score

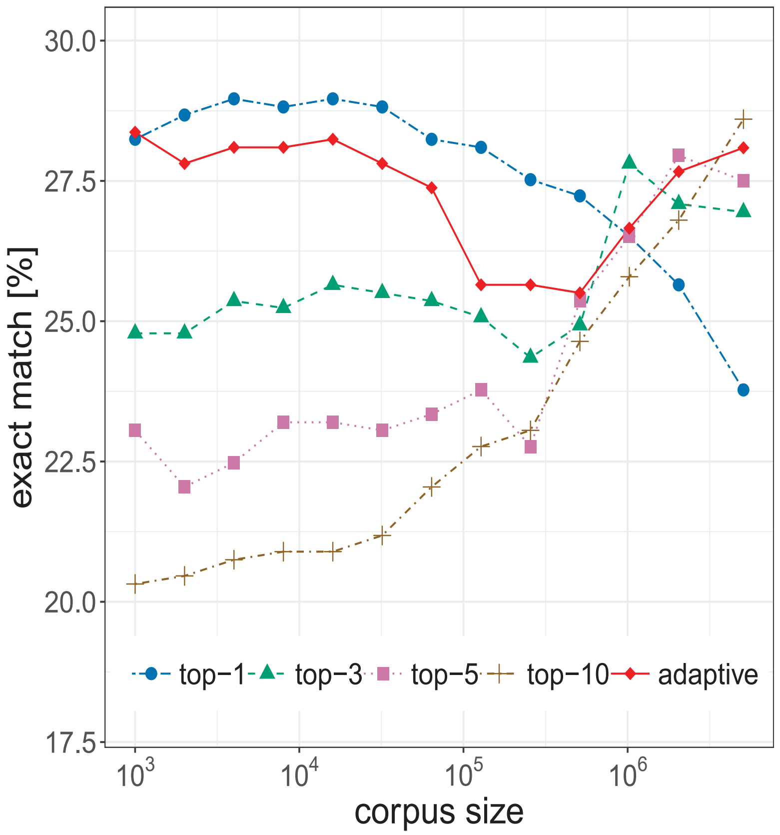
$$s_i = \left[s_i^{(1)}, \ldots, s_i^{(\tau)} \right]^T$$

$$\sum_j s_i^{(j)} = 1$$

Choose passages until surpassing a certain confidence threshold

- if document retrieval is certain → selects fewer docs/passages
- If uncertain → retrieval depth is higher

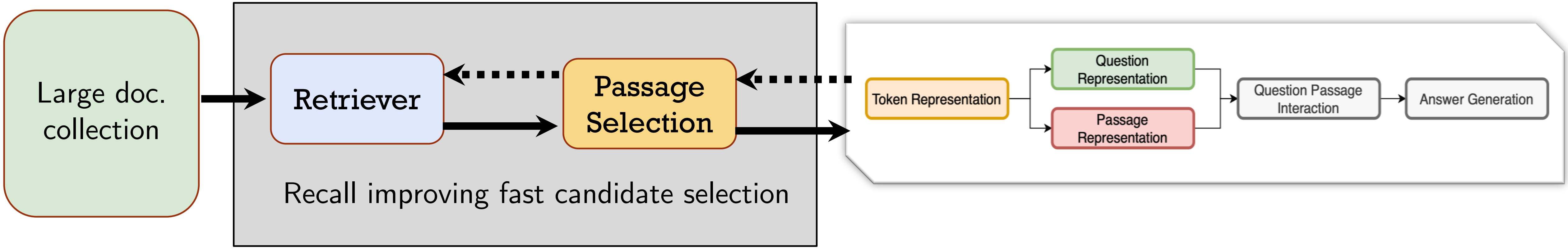
Retrieval Depth and Collection Size



Slightly more involved depth prediction

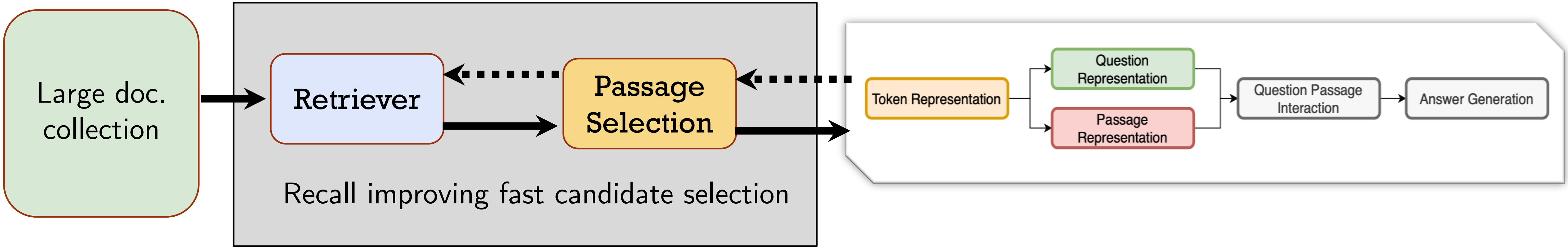
- Predict the rank of the first relevant document
- With a small tolerance

$$\underbrace{n_i}_{\text{Ret. depth}} = \left\lceil s_i^T \underbrace{\beta}_{\text{Learnable param.}} \right\rceil + \underbrace{b}_{\text{tolerance}}$$



How we exploit reader feedback for better retrieval ?

Retriever Reader Interaction

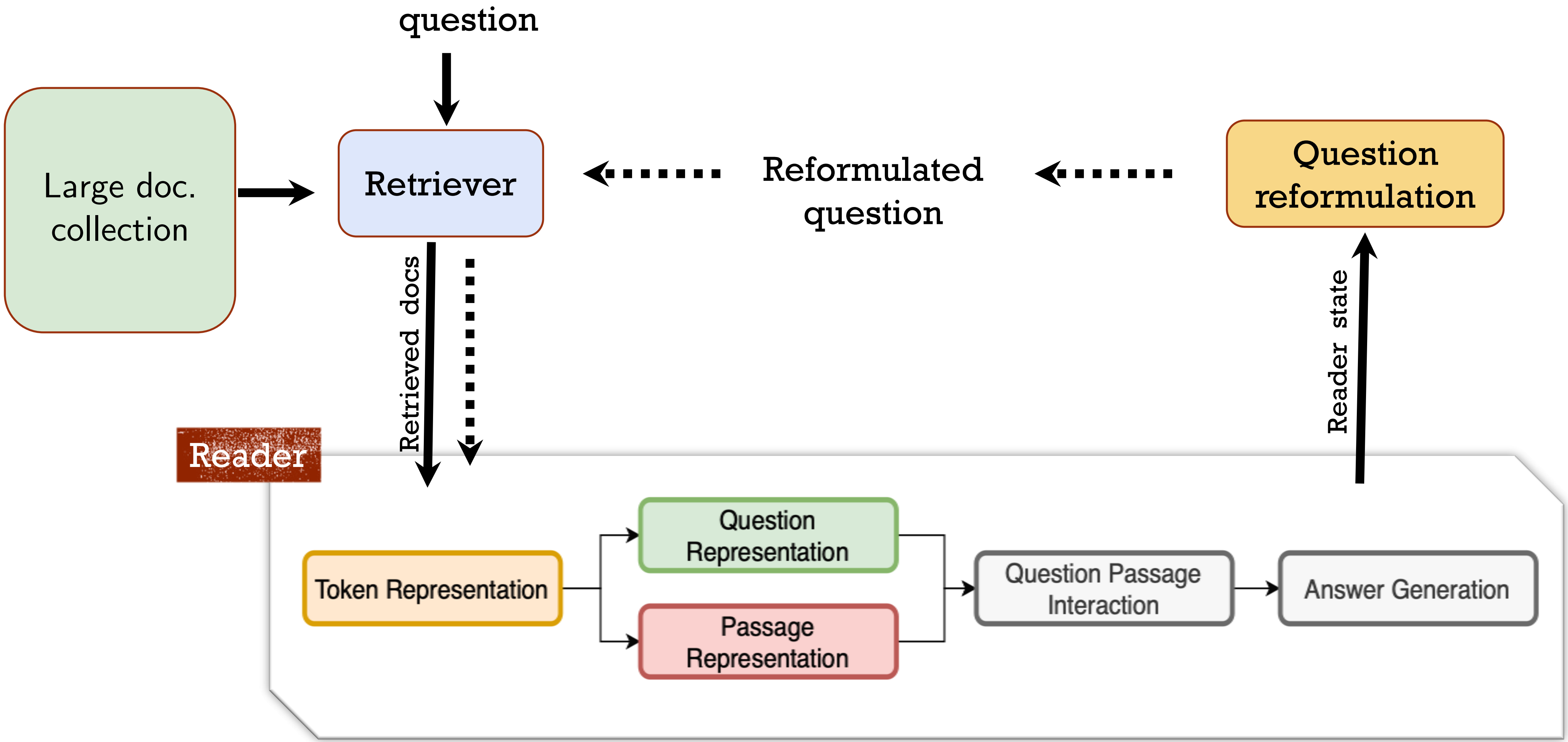


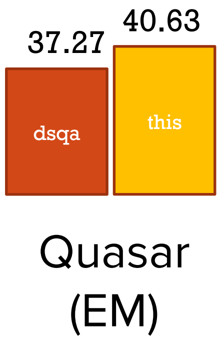
- Single retrieve and read step is limiting – vocabulary gap between question and corpus passages

- How can we enable multi-stage retriever-reader interaction ?
  - Akin to Neural Query Expansion
  - Take care about efficiency concerns

[Das et al. ICLR '19]

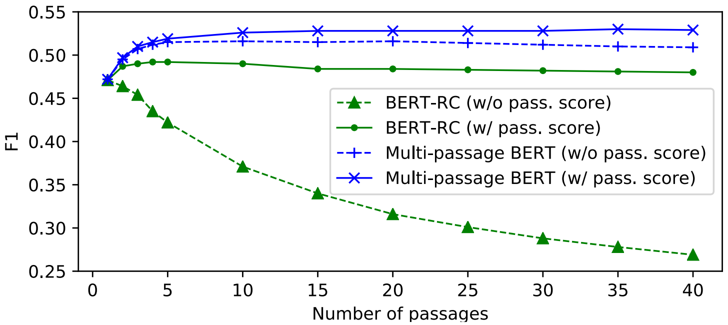
Retriever Reader Interaction





Other Notable Approaches

- Document gated reader [Wang et al. ' 19]
  - Document gating during span prediction
- Tracernet [Dehgani et al '19]
  - Larger contextual models to incorporate reasoning between multiple passages
- R3 [Wang et al '19]
  - Train reader over retrieved docs using the final answer as signal (using REINFORCE)
- Shared Normalization [Clark & Gardner '18, Wang '19]
  - process passages independently, but compute the span probability across spans in all passages in every mini-batch



Other Notable Approaches

Instead of an inverted index, use a vector index

- ORQA [Lee et al '19]
  - Both retriever and reader are learnable (BERT)
- REALM [Wang et al '19]
  - Train reader over retrieved docs using the final answer as signal (using REINFORCE)
- DenSPI [Seo '19]
  - Turns the QA problem into a retrieval problem why sparse encoding of docs and dense indexing of phrases