Open Domain Question Answering (Part 2)

(Papers published in 2021)

Department of Computer Science, Yonsei University

Seungone Kim

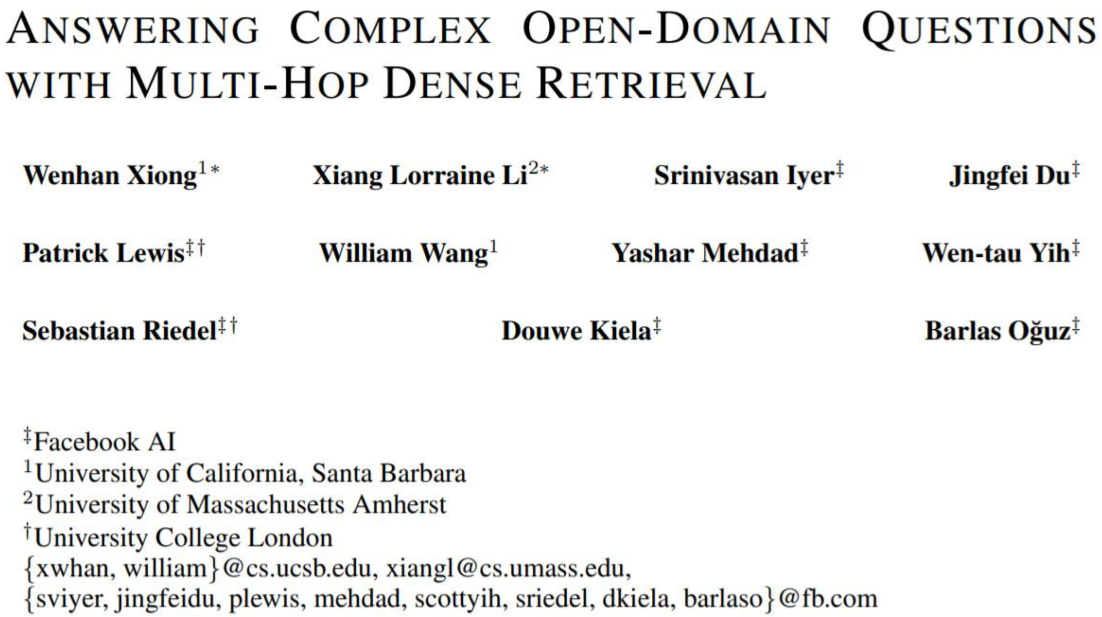
louisdebroglie@yonsei.ac.kr

Referenced Papers

- Prerequisites / Additional Papers
    - Latent retrieval for weakly supervised open domain question answering [ACL 2019]
    - Realm: Retrieval-augmented language model pre-training [ICLR 2020]
    - Dense passage retrieval for open-domain question answering [EMNLP 2020]
    - Revealing the importance of semantic retrieval for machine reading at scale [EMNLP-IJCNLP 2019]
    - Differentiable reasoning over a virtual knowledge base [ICLR 2020]
    - Learning to retrieve reasoning paths over wikipedia graph for question answering [ICLR 2020]
    - Multi-hop reading comprehension through question decomposition and rescoring [ACL 2019]
    - Unsupervised question decomposition for question answering [EMNLP 2020]

    - Answering complex open-domain questions through iterative query generation [EMNLP-IJCNLP 2019]
    - Multi-step entity-centric information retrieval for multi-hop question answering [ACL 2019 Workshop]
    - Transformer-xh : Multi-evidence reasoning with extra hop attention [ICLR 2020]
    - Adaptive information gathering via imitation learning [Robotics: Science and Systems 2017]

- Key Papers
    - Answering Complex Open-Domain Questions with Multi-hop Dense Retrieval [ICLR 2021]
    - Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]
    - Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]



ANSWERING COMPLEX OPEN-DOMAIN QUESTIONS WITH MULTI-HOP DENSE RETRIEVAL

Wenhan Xiong[1*] **Xiang Lorraine Li**[2*] **Srinivasan Iyer**[‡] **Jingfei Du**[‡]

Patrick Lewis[‡†] **William Wang**[1] **Yashar Mehdad**[‡] **Wen-tau Yih**[‡]

Sebastian Riedel[‡†] **Douwe Kiela**[‡] **Barlas Oğuz**[‡]

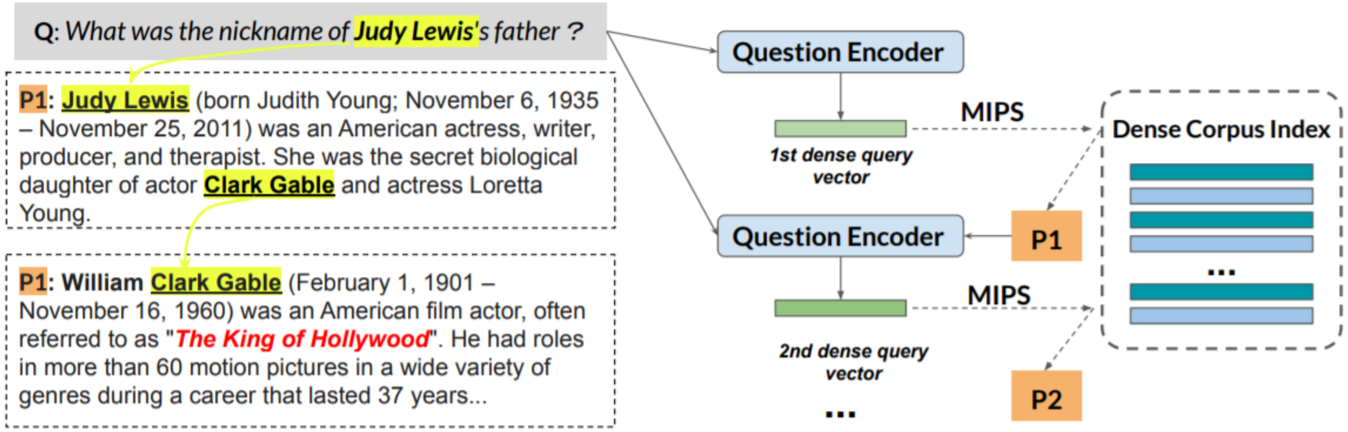
[‡]Facebook AI
[1]University of California, Santa Barbara
[2]University of Massachusetts Amherst
[†]University College London
{xwhan, william}@cs.ucsb.edu, xiangl@cs.umass.edu,
{sviyer, jingfeidu, plewis, mehdad, scottyih, sriedel, dkiela, barlaso}@fb.com

ICLR 2021

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Recent methods remain limited to simple questions, where the answer to the question is explicit in a single piece of text.
  - In contrast, complex questions typically involve **aggregating information from multiple documents**, requiring **logical reasoning** or **sequential processing**.
  - Therefore, single-shot approaches such as *ORQA, DPR, REALM* are insufficient and **iterative methods** are needed to recursively retrieve new information at each step.

  - The problem in answering **Multi-hop Open-Domain Questions** is that the **search space grows exponentially** with each retrieval hop.
  - Although methods constructing a document graph with entity linking or existing hyperlink structure has gained good performance, these methods may not **generalize to new domains**, where entity linking might perform poorly, or where **hyperlinks** might not be as abundant as Wikipedia.
  - In contrast, the authors propose to employ **dense retrieval to the multi-hop setting** with a simple recursive framework, without the help of hyperlinks.



Q: *What was the nickname of **Judy Lewis**'s father ?*

P1: **Judy Lewis** (born Judith Young; November 6, 1935 – November 25, 2011) was an American actress, writer, producer, and therapist. She was the secret biological daughter of actor **Clark Gable** and actress Loretta Young.

P1: **William Clark Gable** (February 1, 1901 – November 16, 1960) was an American film actor, often referred to as "***The King of Hollywood***". He had roles in more than 60 motion pictures in a wide variety of genres during a career that lasted 37 years...

Figure 1: An overview of the multi-hop dense retrieval approach.

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Methodology
  - The authors model the probability of selecting a certain passage sequence as follows.

$$P(\mathcal{P}_{seq}|q) = \prod_{t=1}^{n} P(p_t|q, p_1, ..., p_{t-1}),$$

  - At each retrieval step, they construct a **new query representation** based on previous results and the retrieval is implemented as MIPS over dense representations.

$$P(p_t|q, p_1, ..., p_{t-1}) = \frac{\exp\left(\langle \boldsymbol{p}_t, \boldsymbol{q}_t \rangle\right)}{\sum_{p \in \mathcal{C}} \exp\left(\langle \boldsymbol{p}, \boldsymbol{q}_t \rangle\right)}, \text{ where } \boldsymbol{q}_t = g(q, p_1, ..., p_{t-1}) \text{ and } \boldsymbol{p}_t = h(p_t).$$

  - The formulation is similar to previous methods except that the authors add the **query reformulation process** conditioned on previous retrieval results.
  - Also, instead of using a **bi-encoder architecture** with separately parameterized encoders for queries and passages, the authors use a shared RoBERTa-base.

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Methodology
    - The training procedure is similar to DPR, where the authors obtain **negative passages** in a combination of passages in the current batch and **hard negatives** using TF-IDF retrieved passages and their linked pages in Wikipedia.

    - In addition to in-batch negatives, the authors use a **memory bank mechanism** to further increase the number of negative examples for each question.
    - The memory bank stores a large number of dense passage vectors,
    - After training the shared encoder, it is freezed as a passage encoder and collect a bank of passage representations across multiple batches to serve as a set of negative passages.

    - For inference, the authors **first encode the whole corpus** into an index of passage vectors.
    - Given a question, the authors use beam search to obtain top-k passage sequence candidates, where the candidates to **beam search** at each step are generated by MIPS using the query encoder at step t.
    - The top-k sequences will then be fed into task-specific downstream modules to produce the desired outputs.

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Experiments
  - The authors focus on two datasets : **HotpotQA** and **Multi-evidence FEVER**.
  - Retrieval results are predicted using **exact inner product search index (IndexFlatIP)** in **FAISS** (Johnson et al., 2017).

  - **TF-IDF + Linked** : Also extracts the hyperlinked passages from TF-IDF passages, and then reranks both both TF-IDF and hyperlinked passages with BM25 scores.
  - **DrKIT** : A dense retrieval approach, which builds a entity-level dense index for retrieval. (Restricts next hop to using hyperlinked entities)
  - **Entity Linking** : Used in fact verification; Uses a constituency parser to extract potential entity mentions in fact claim and use MediaWiki API to search documents.

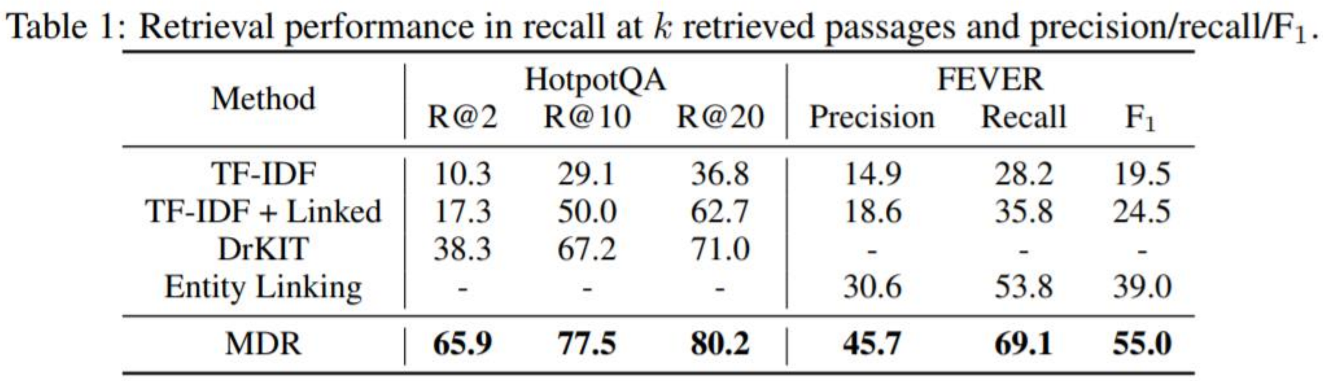


Table 1: Retrieval performance in recall at k retrieved passages and precision/recall/F_1.

Method	HotpotQA			FEVER		
	R@2	R@10	R@20	Precision	Recall	F_1
TF-IDF	10.3	29.1	36.8	14.9	28.2	19.5
TF-IDF + Linked	17.3	50.0	62.7	18.6	35.8	24.5
DrKIT	38.3	67.2	71.0	-	-	-
Entity Linking	-	-	-	30.6	53.8	39.0
MDR	**65.9**	**77.5**	**80.2**	**45.7**	**69.1**	**55.0**

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Experiments
  - Reranking documents returned by retrieval methods with a more sophisticated model is a common strategy for improving retrieval quality.
  - On HotpotQA, the authors test the effectiveness of MDR after a simple cross-attention reranking.
  - Each of the top-k passage sequences from MDR is first prepended with the original question and then fed into a PTLM that predicts relevant scores.

  - **Semantic Retrieval** : Uses BERT at both passage-level and sentence-level to select context from the initial TF-IDF and hyperlinked passages.
  - **Graph Recurrent Retriever** : Learns to recursively select the best passage sequence on top of a hyperlinked passage graph, where passage is encoded with BERT.

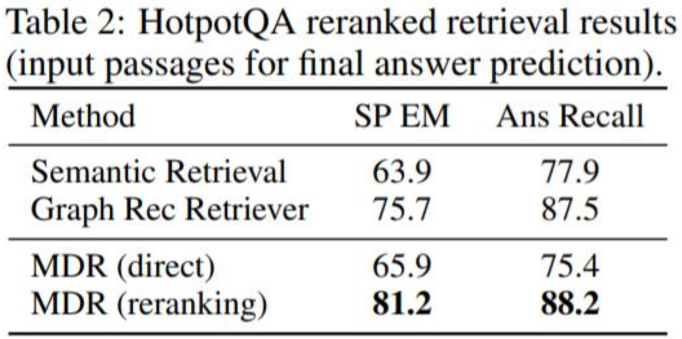


Table 2: HotpotQA reranked retrieval results (input passages for final answer prediction).

Method	SP EM	Ans Recall
Semantic Retrieval	63.9	77.9
Graph Rec Retriever	75.7	87.5
MDR (direct)	65.9	75.4
MDR (reranking)	**81.2**	**88.2**

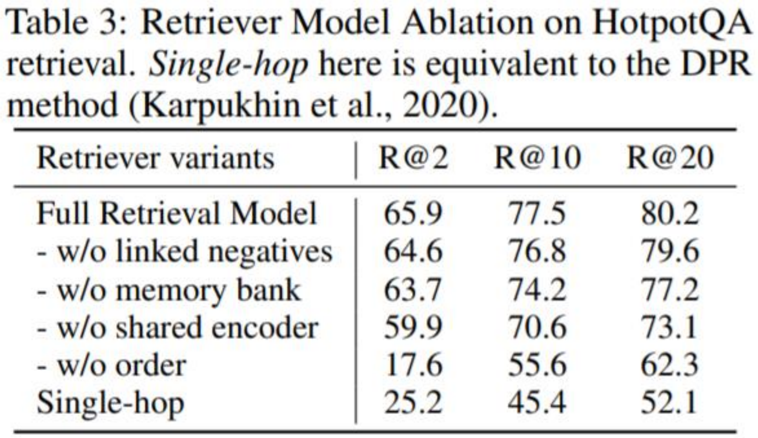


Table 3: Retriever Model Ablation on HotpotQA retrieval. *Single-hop* here is equivalent to the DPR method (Karpukhin et al., 2020).

Retriever variants	R@2	R@10	R@20
Full Retrieval Model	65.9	77.5	80.2
- w/o linked negatives	64.6	76.8	79.6
- w/o memory bank	63.7	74.2	77.2
- w/o shared encoder	59.9	70.6	73.1
- w/o order	17.6	55.6	62.3
Single-hop	25.2	45.4	52.1

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Experiments
  - As multi-hop questions have more complex structures than simple questions,
    recent studies (Min et al., 2019; Perez et al., 2020) propose to use **explicit question decomposition** to simplify the problem.
  - While previous works have shown that with TF-IDF, using decomposed questions improves the retrieval results.
  - The authors investigate if this conclusion holds true with **stronger dense retrieval methods**.
  - With human-annotated question decomposition from QDMR dataset,
  - Q : Mick Carter is the landlord of a public house located at what address?
  - SubQ1 : What is the public house that Mack Carter is landlord of?
  - SubQ@ : What is the address that #1 is located at?
  - The authors do not observe any strong improvements from explicit question decompositions.
  - This suggests that strong pretrained encoders can effectively learn to select necessary information from the multi-hop question at each retrieval step.

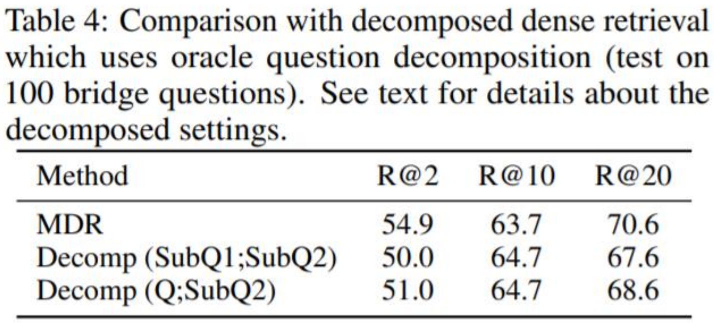


Table 4: Comparison with decomposed dense retrieval which uses oracle question decomposition (test on 100 bridge questions). See text for details about the decomposed settings.

Method	R@2	R@10	R@20
MDR	54.9	63.7	70.6
Decomp (SubQ1;SubQ2)	50.0	64.7	67.6
Decomp (Q;SubQ2)	51.0	64.7	68.6

Answering Complex Open-Domain Questions with Multi-Hop Dense Retrieval [ICLR 2021]

- Experiments
  - The authors evaluate how the better retrieval results of MDR improve multi-hop question answering.
  - The best extractive reader model is based on ELECTRA, BERT-large.
  - The best generative reader model is based on RAG, FiD.

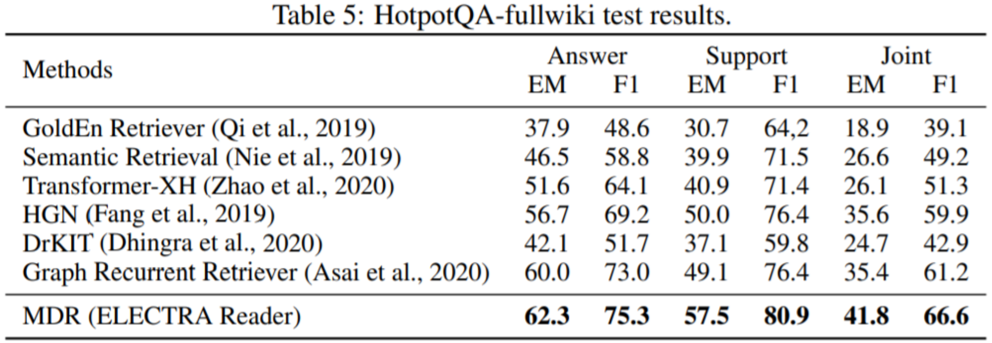


Table 5: HotpotQA-fullwiki test results.

Methods	Answer		Support		Joint	
	EM	F1	EM	F1	EM	F1
GoldEn Retriever (Qi et al., 2019)	37.9	48.6	30.7	64,2	18.9	39.1
Semantic Retrieval (Nie et al., 2019)	46.5	58.8	39.9	71.5	26.6	49.2
Transformer-XH (Zhao et al., 2020)	51.6	64.1	40.9	71.4	26.1	51.3
HGN (Fang et al., 2019)	56.7	69.2	50.0	76.4	35.6	59.9
DrKIT (Dhingra et al., 2020)	42.1	51.7	37.1	59.8	24.7	42.9
Graph Recurrent Retriever (Asai et al., 2020)	60.0	73.0	49.1	76.4	35.4	61.2
MDR (ELECTRA Reader)	**62.3**	**75.3**	**57.5**	**80.9**	**41.8**	**66.6**

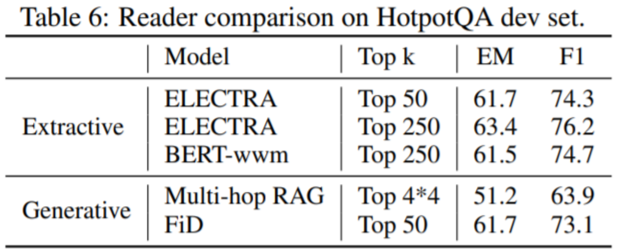


Table 6: Reader comparison on HotpotQA dev set.

	Model	Top k	EM	F1
Extractive	ELECTRA	Top 50	61.7	74.3
	ELECTRA	Top 250	63.4	76.2
	BERT-wwm	Top 250	61.5	74.7
Generative	Multi-hop RAG	Top 4*4	51.2	63.9
	FiD	Top 50	61.7	73.1

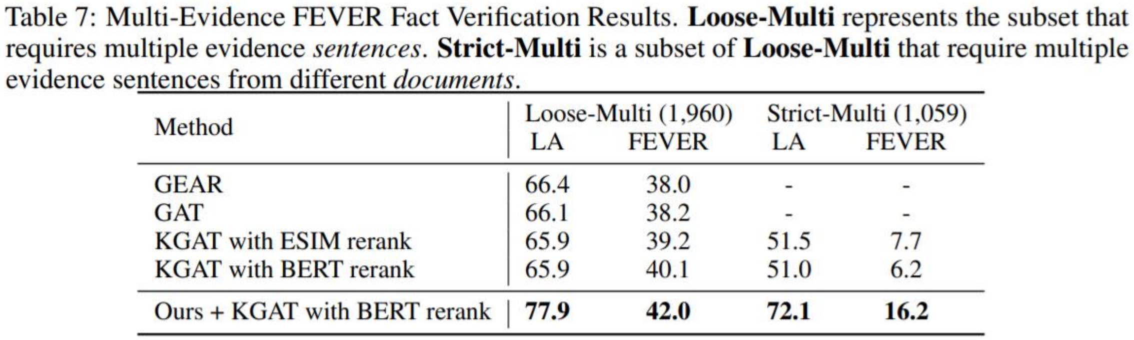


Table 7: Multi-Evidence FEVER Fact Verification Results. **Loose-Multi** represents the subset that requires multiple evidence *sentences*. **Strict-Multi** is a subset of **Loose-Multi** that require multiple evidence sentences from different *documents*.

Method	Loose-Multi (1,960)		Strict-Multi (1,059)	
	LA	FEVER	LA	FEVER
GEAR	66.4	38.0	-	-
GAT	66.1	38.2	-	-
KGAT with ESIM rerank	65.9	39.2	51.5	7.7
KGAT with BERT rerank	65.9	40.1	51.0	6.2
Ours + KGAT with BERT rerank	**77.9**	**42.0**	**72.1**	**16.2**

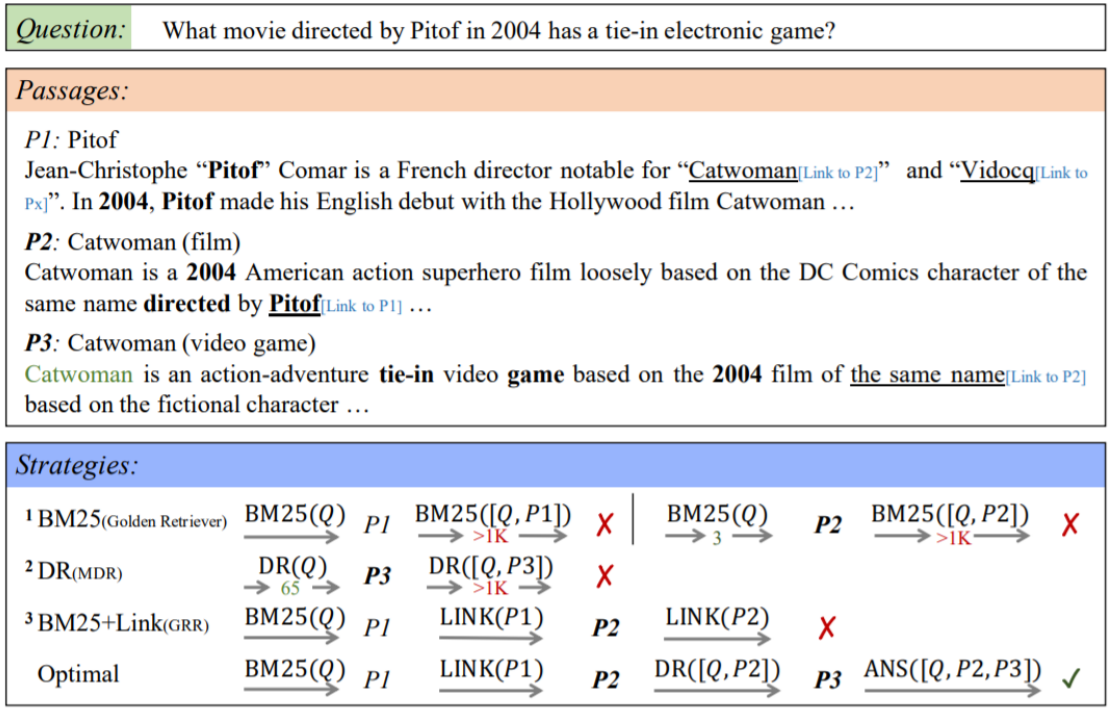
Adaptive Information Seeking for Open-Domain Question Answering

Yunchang Zhu[†§], Liang Pang[†*], Yanyan Lan[◇*], Huawei Shen[†§], Xueqi Cheng[‡§]
[†]Data Intelligence System Research Center
and [‡]CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences
[§]University of Chinese Academy of Sciences
[◇]Institute for AI Industry Research, Tsinghua University
{zhuyunchang17s, pangliang, shenhuawei, cxq}@ict.ac.cn
lanyanyan@tsinghua.edu.cn

EMNLP 2021

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Almost all existing iterative approaches use predefined strategies, either applying the **same retrieval function** multiple times or **fixing the order** of different retrieval functions.
  - **Traditional two-stage retriever-reader pipeline**(DRQA, DPR) have limitations in answering complex questions, which need multi-hop or logical reasoning.
  - To tackle this issue, **iterative approaches** have been proposed to recurrently retrieve passages and reformulate the query based on the original question and the previously collected passages.
  - However, the **fixed information-seeking strategies** cannot meet the **diversified requirements** of various problems.



Question: What movie directed by Pitof in 2004 has a tie-in electronic game?

Passages:

P1: Pitof
Jean-Christophe “**Pitof**” Comar is a French director notable for “Catwoman[Link to P2]” and “Vidocq[Link to Px]”. In **2004**, **Pitof** made his English debut with the Hollywood film Catwoman …

***P2*:** Catwoman (film)
Catwoman is a **2004** American action superhero film loosely based on the DC Comics character of the same name **directed** by **Pitof**[Link to P1] …

***P3*:** Catwoman (video game)
Catwoman is an action-adventure **tie-in** video **game** based on the **2004** film of the same name[Link to P2] based on the fictional character …

Strategies:

[1] BM25(Golden Retriever) $\xrightarrow{\text{BM25}(Q)}$ *P1* $\xrightarrow[>1K]{\text{BM25}([Q,P1])}$ ✗ | $\xrightarrow[3]{\text{BM25}(Q)}$ ***P2*** $\xrightarrow[>1K]{\text{BM25}([Q,P2])}$ ✗

[2] DR(MDR) $\xrightarrow[65]{\text{DR}(Q)}$ ***P3*** $\xrightarrow[>1K]{\text{DR}([Q,P3])}$ ✗

[3] BM25+Link(GRR) $\xrightarrow{\text{BM25}(Q)}$ *P1* $\xrightarrow{\text{LINK}(P1)}$ ***P2*** $\xrightarrow{\text{LINK}(P2)}$ ✗

Optimal $\xrightarrow{\text{BM25}(Q)}$ *P1* $\xrightarrow{\text{LINK}(P1)}$ ***P2*** $\xrightarrow{\text{DR}([Q,P2])}$ ***P3*** $\xrightarrow{\text{ANS}([Q,P2,P3])}$ ✓

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Adaptive Information Seeking approach for Open-domain QA (AISO)
    - The whole retrieval and answer process is modeled as a partially observed **Markov decision process(POMDP)** to reflect the interactive characteristics between the **QA model(Agent)** and the **interactable large scale corpus(Environment)**.
    - The agent is asked to perform an action according to its **state (belief module)** and the **policy** it learned **(policy module)**.
    - The **belief module** of the agent maintains a set of evidence to form its state.
    - In each step, the agent emits an action to the environment, which returns a passage as the observation back to the agent.
    - The agent updates the evidence set and generates the next action, step by step, until the evidence set is sufficient to trigger the **answer action** to answer the question.
    - To learn the strategy, the authors train the **policy module** in **imitation learning** by cloning the behavior of an oracle online, which avoids the hassle of designing reward functions and **solves the POMDP in the fashion of supervised learning**.

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Adaptive Information Seeking approach for Open-domain QA (AISO)

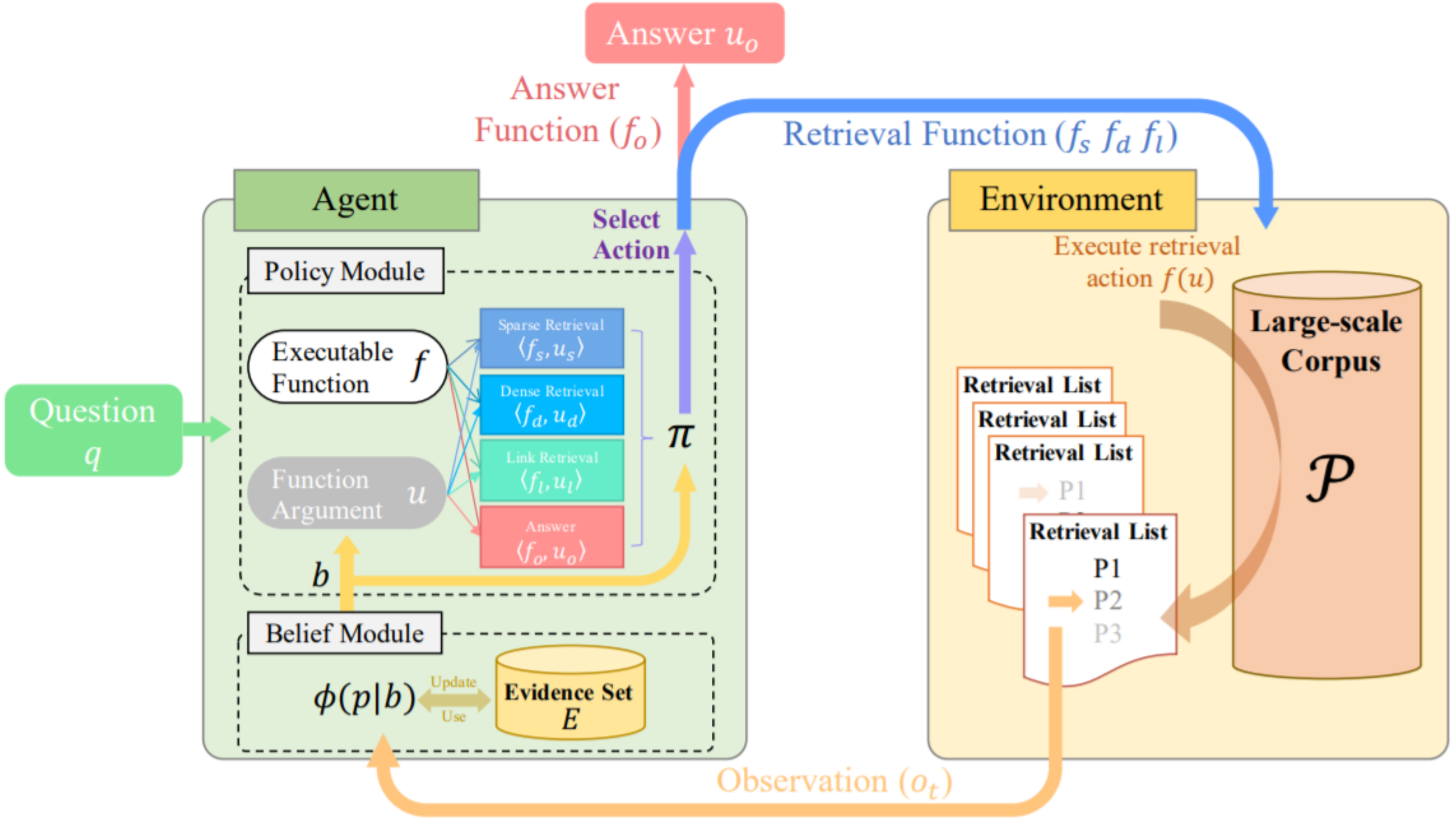


Figure 2: The overview of the AISO.

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Methodology
  - The **space of executable functions** includes two groups of functions : **retrieval function** and **answer function**.
  - The **state space** contains revealing states of retrieval lists of all history retrieval actions.
  - On reaching a new **environment state**, the environment will return an observation from the **observation space**, which is the last passage retrieved.
  - The **agent** consists of two modules : **belief module** that generates a belief state from experience, and **policy module** that prescribes action to take for current belief.
  - Both belief and policy modules are constructed based on pretrained Transformer encoders, which encode each inputted token into a d-dimensional representation.
  - *[CLS] [YES] [NO] [NONE] question [SEP] title1 [SOP] content1 [SEP] title2 [SOP] content2 [SEP] … content-|E| [SEP].*

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Methodology
  - The **belief module Φ** transforms the agent's experience h_t into belief state b_t by maintaining a set of evidence E_{t-1}.
  - At the end of the process, the **evidence set E** is expected to contain sufficient evidence necessary to answer the question and no irrelevant passage.

$$b_t = \Pi(h_t) = \langle q, C_t \rangle = \langle q, E_{t-1} \cup \{o_t\} \rangle. \quad (1)$$

$$E_t = \{p_i | \phi(p_i|b_t) > \phi(p_0|b_t), p_i \in C_t\}. \quad (2)$$

  - The **policy module Π** decides the next action to be taken based on the current belief state b_t.

  - The space of **executable functions** is defined as **sparse RF(f_s)**, **dense RF(f_d)**, **link RF(f_l)** and **answer function(f_o)**.
  - The **space of function arguments**, composed of textual queries and answers is too large to perform an exhaustive search due to the complexity of natural language.
  - To reduce the search complexity, the authors employ **four argument generators** to generate the most plausible query/answer for the equipped functions.

  - g_o uses the contextual representations to calculate the start and end positions of the most plausible answer u_o.
  - g_s is a query reformulation model for f_s, which takes the belief state b_t as input and outputs a span of the input sequence as the sparse query u_s.
  - g_d is a query reformulation model for f_d, which concatenates the question and the passage with the highest score in the evidence set E_t to make u_d.
  - g_l is a multi-class classifier for f_l, which selects the most promising anchor test from the belief state b_f.
  - In this way, the **action space** is narrowed down to $\tilde{A} = \{< f_s, u_s >, < f_d, u_d >, < f_l, u_l >, < f_o, u_o >\}$.

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Methodology
  - The **action scoring function** $\boldsymbol{\Pi}$ is built upon the output of the Ψ^{policy}.
  - To score an action $< f, u >$ for current belief state, an MLP-ReLU network projects the concatenated representation of b_t, f, and function argument u.

$$a_t = \Pi(b_t) = \arg\max_{a \in \breve{A}} \pi(a|b_t). \qquad (3)$$

- Training
  - In the agent, in addition to the encoders $\Psi^{belief}, \Psi^{policy}$, the **evidence scoring function** $\boldsymbol{\Phi}$, **link classifier** $\boldsymbol{g_l}$, **answer extractor** $\boldsymbol{g_o}$ and **action scoring function** $\boldsymbol{\Pi}$ needs to be trained along.

$$L = L_\phi + L_l + L_o + L_\pi. \qquad (4)$$

  - The authors explore the use of **imitation learning** by querying a model-based oracle online and imitating the action chose by the oracle, which avoids the hassle of designing R and solves **POMDP** in the fashion of supervised learning.

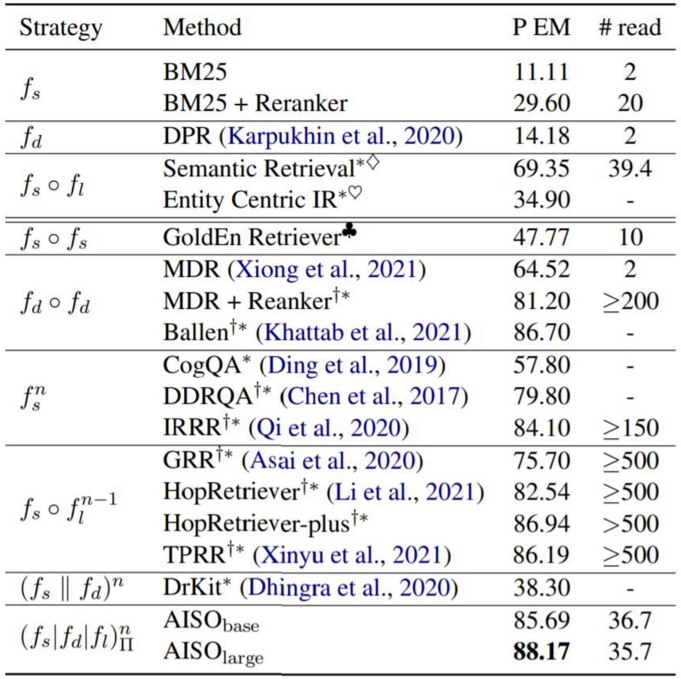
$$L_\pi = -\log \frac{e^{\pi(a^\star|b)}}{\sum_{a \in \breve{A}} e^{\pi(a|b)}}, \qquad (5)$$

$$L_\phi(\boldsymbol{y}, b) = -\log P(\tau_{\boldsymbol{y}} | \{\phi(p_i|b)\}_{i=0}^{|C|}), \qquad (6)$$

  - Loss of **evidence scoring function** is defined as negative log likelihood, loss of **action scoring function** is designed as cross entropy, and the other two losses(**link classifier**, **answer extractor**) are multi-class cross entropy.

Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

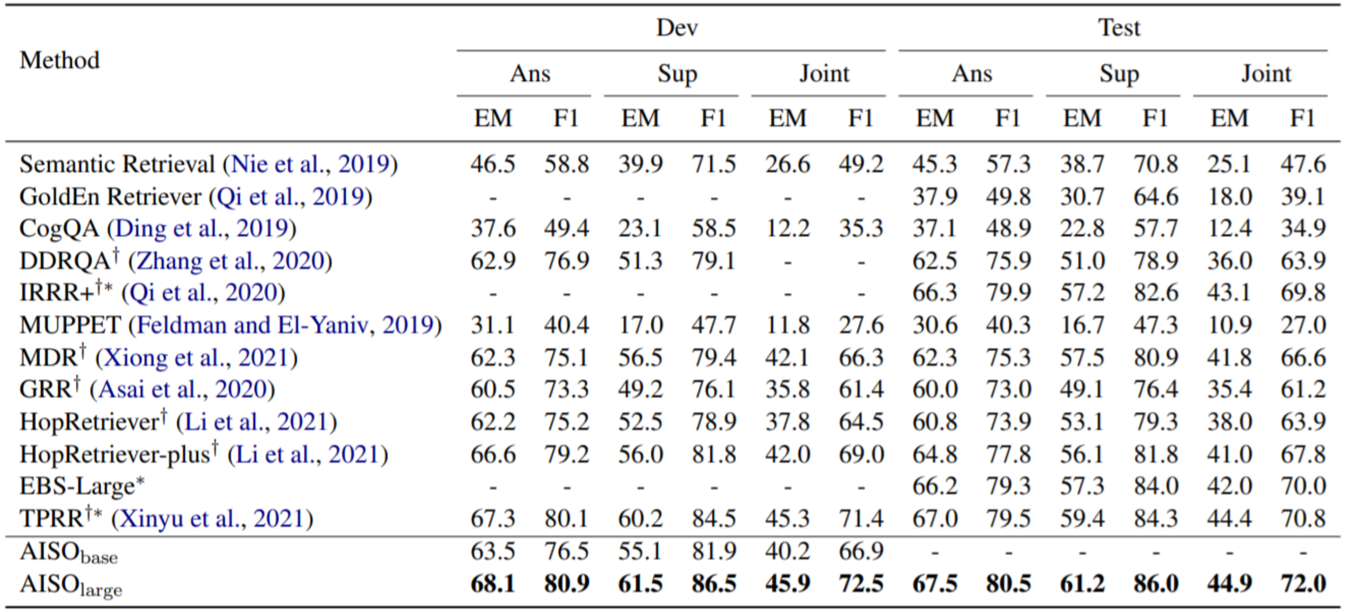
- Experiments
  - The authors use **HotpotQA** and **SQuAD Open** as their QA benchmark to test the framework.
  - For **sparse retrieval**, the authors index all passages in the corpus with **Elasticsearch** and implement BM25.
  - For **dense retrieval**, the authors leverage the trained passage encoder and query encoder and index all passage vectors using FAISS.
  - During training, the authors use the **HNSW-based index** for efficient low-latency retrieval; while in test time, they use **exact inner product search index**.
  - For **link retrieval**, the filtered hyperlinks are used, whose targets have to be another article from the dump.



Strategy	Method	P EM	# read
f_s	BM25	11.11	2
	BM25 + Reranker	29.60	20
f_d	DPR (Karpukhin et al., 2020)	14.18	2
$f_s \circ f_l$	Semantic Retrieval$^{*\diamondsuit}$	69.35	39.4
	Entity Centric IR$^{*\heartsuit}$	34.90	-
$f_s \circ f_s$	GoldEn Retriever$^{\clubsuit}$	47.77	10
$f_d \circ f_d$	MDR (Xiong et al., 2021)	64.52	2
	MDR + Reanker$^{\dagger *}$	81.20	≥200
	Ballen$^{\dagger *}$ (Khattab et al., 2021)	86.70	-
f_s^n	CogQA* (Ding et al., 2019)	57.80	-
	DDRQA$^{\dagger *}$ (Chen et al., 2017)	79.80	-
	IRRR$^{\dagger *}$ (Qi et al., 2020)	84.10	≥150
$f_s \circ f_l^{n-1}$	GRR$^{\dagger *}$ (Asai et al., 2020)	75.70	≥500
	HopRetriever$^{\dagger *}$ (Li et al., 2021)	82.54	≥500
	HopRetriever-plus$^{\dagger *}$	86.94	>500
	TPRR$^{\dagger *}$ (Xinyu et al., 2021)	86.19	≥500
$(f_s \parallel f_d)^n$	DrKit* (Dhingra et al., 2020)	38.30	-
$(f_s \mid f_d \mid f_l)^n_{\Pi}$	AISO$_{\text{base}}$	85.69	36.7
	AISO$_{\text{large}}$	**88.17**	35.7

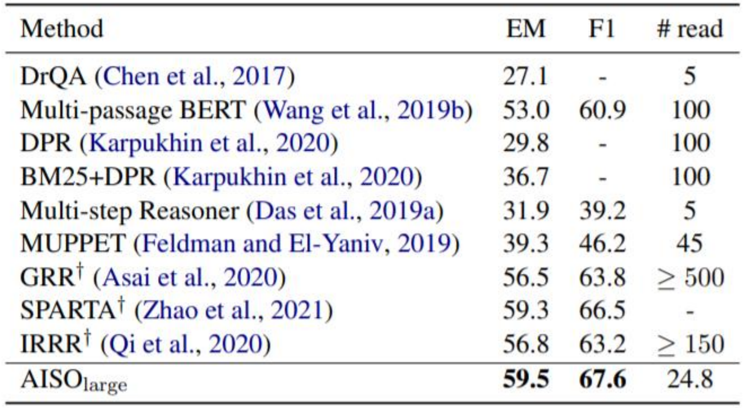
Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Experiments



Method	Dev						Test					
	Ans		Sup		Joint		Ans		Sup		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Semantic Retrieval (Nie et al., 2019)	46.5	58.8	39.9	71.5	26.6	49.2	45.3	57.3	38.7	70.8	25.1	47.6
GoldEn Retriever (Qi et al., 2019)	-	-	-	-	-	-	37.9	49.8	30.7	64.6	18.0	39.1
CogQA (Ding et al., 2019)	37.6	49.4	23.1	58.5	12.2	35.3	37.1	48.9	22.8	57.7	12.4	34.9
DDRQA† (Zhang et al., 2020)	62.9	76.9	51.3	79.1	-	-	62.5	75.9	51.0	78.9	36.0	63.9
IRRR+$^{\dagger *}$ (Qi et al., 2020)	-	-	-	-	-	-	66.3	79.9	57.2	82.6	43.1	69.8
MUPPET (Feldman and El-Yaniv, 2019)	31.1	40.4	17.0	47.7	11.8	27.6	30.6	40.3	16.7	47.3	10.9	27.0
MDR† (Xiong et al., 2021)	62.3	75.1	56.5	79.4	42.1	66.3	62.3	75.3	57.5	80.9	41.8	66.6
GRR† (Asai et al., 2020)	60.5	73.3	49.2	76.1	35.8	61.4	60.0	73.0	49.1	76.4	35.4	61.2
HopRetriever† (Li et al., 2021)	62.2	75.2	52.5	78.9	37.8	64.5	60.8	73.9	53.1	79.3	38.0	63.9
HopRetriever-plus† (Li et al., 2021)	66.6	79.2	56.0	81.8	42.0	69.0	64.8	77.8	56.1	81.8	41.0	67.8
EBS-Large*	-	-	-	-	-	-	66.2	79.3	57.3	84.0	42.0	70.0
TPRR$^{\dagger *}$ (Xinyu et al., 2021)	67.3	80.1	60.2	84.5	45.3	71.4	67.0	79.5	59.4	84.3	44.4	70.8
AISO$_{\text{base}}$	63.5	76.5	55.1	81.9	40.2	66.9	-	-	-	-	-	-
AISO$_{\text{large}}$	**68.1**	**80.9**	**61.5**	**86.5**	**45.9**	**72.5**	**67.5**	**80.5**	**61.2**	**86.0**	**44.9**	**72.0**

Table 2: Answer extraction and supporting sentence identification performance on HotpotQA fullwiki. The methods with † use the large version of pretrained language models comparable to AISO$_{\text{large}}$. The results marked with $*$ are from the official leaderboard otherwise originated from published papers.

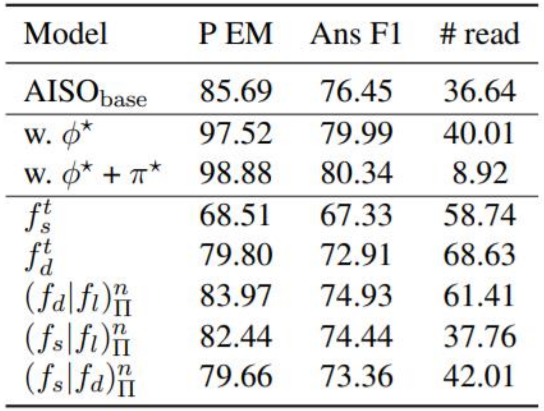


Method	EM	F1	# read
DrQA (Chen et al., 2017)	27.1	-	5
Multi-passage BERT (Wang et al., 2019b)	53.0	60.9	100
DPR (Karpukhin et al., 2020)	29.8	-	100
BM25+DPR (Karpukhin et al., 2020)	36.7	-	100
Multi-step Reasoner (Das et al., 2019a)	31.9	39.2	5
MUPPET (Feldman and El-Yaniv, 2019)	39.3	46.2	45
GRR† (Asai et al., 2020)	56.5	63.8	≥ 500
SPARTA† (Zhao et al., 2021)	59.3	66.5	-
IRRR† (Qi et al., 2020)	56.8	63.2	≥ 150
AISO$_{\text{large}}$	**59.5**	**67.6**	24.8

Table 3: Question answering performance on SQuAD Open benchmark. † denotes the methods use the large pretrained language models comparable to AISO$_{\text{large}}$.

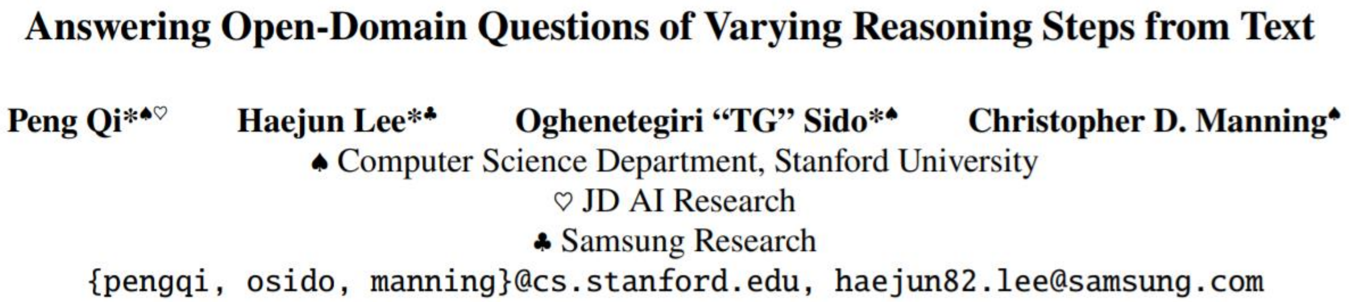
Adaptive Information Seeking for Open-Domain Question Answering [EMNLP 2021]

- Analysis
  - Belief module has a more impact on the performance than the cost.
  - A Good policy can greatly improve efficiency.
  - The lack of any RF will degrade performance, which illustrates that all RFs contribute to performance.



Model	P EM	Ans F1	# read
$\text{AISO}_{\text{base}}$	85.69	76.45	36.64
w. $\phi^\star$	97.52	79.99	40.01
w. $\phi^\star + \pi^\star$	98.88	80.34	8.92
f_s^t	68.51	67.33	58.74
f_d^t	79.80	72.91	68.63
$(f_d\|f_l)^n_\Pi$	83.97	74.93	61.41
$(f_s\|f_l)^n_\Pi$	82.44	74.44	37.76
$(f_s\|f_d)^n_\Pi$	79.66	73.36	42.01

Table 4: Analysis experiments on HotpotQA fullwiki.



Answering Open-Domain Questions of Varying Reasoning Steps from Text

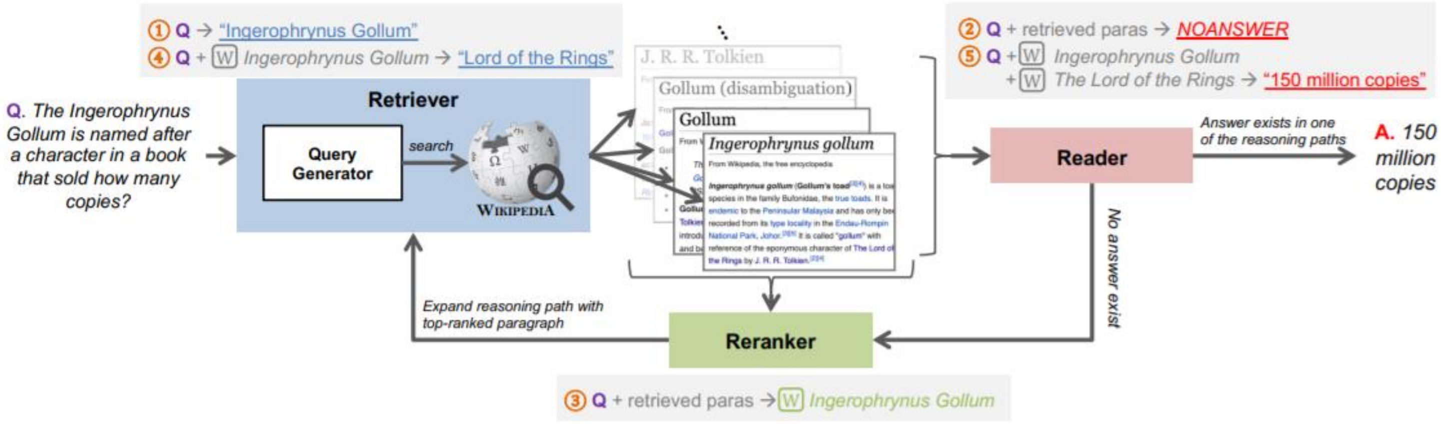
Peng Qi[*♠♡] Haejun Lee[*♣] Oghenetegiri “TG” Sido[*♠] Christopher D. Manning[♠]
[♠] Computer Science Department, Stanford University
[♡] JD AI Research
[♣] Samsung Research
{pengqi, osido, manning}@cs.stanford.edu, haejun82.lee@samsung.com

EMNLP 2021

Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]

- Despite the success of multi-hop reasoning systems over multiple pieces of evidence, most previous systems are developed with datasets that contain **exclusively** single-hop questions or two-hop ones.
    - In practice , not only can we expect open-domain QA systems to receive exclusively single or multi-hop questions from uses, but it is also **non-trivial to judge** reliably whether a question requires one or multiple pieces of evidence to answer a priori.
    - Besides the **impractical assumption about reasoning hops**, previous work often also assumes **access to non-textual metadata** such as knowledge bases, entity linking, and Wikipedia hyperlinks when retrieving supporting facts, especially in answering complex questions.
    - Although this information is helpful, it is not always available in text collections we might be interested in getting answers from, such as news or academic research articles, besides being labor-intensive and time-consuming to collect and maintain.

    - To address these limitations, the authors propose **Iterative Retriever, Reader, and Reranker (IRRR)**, which features a single neural network model that performs all of the subtasks required to answer questions from a large collection of text.
    - Also, the authors propose a new open-domain QA benchmark, **BEERQA**, that features questions requiring variable steps of reasoning to answer on Wikipedia corpus.


① Q → "Ingerophrynus Gollum"
④ Q + W Ingerophrynus Gollum → "Lord of the Rings"
Q. The Ingerophrynus Gollum is named after a character in a book that sold how many copies?
Retriever
Query Generator
search
Wikipedia
J. R. R. Tolkien
Gollum (disambiguation)
Gollum
Ingerophrynus gollum
② Q + retrieved paras → NOANSWER
⑤ Q + W Ingerophrynus Gollum
+ W The Lord of the Rings → "150 million copies"
Reader
Answer exists in one of the reasoning paths
A. 150 million copies
No answer exist
Expand reasoning path with top-ranked paragraph
Reranker
③ Q + retrieved paras → W Ingerophrynus Gollum

Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]

- Methodology (Iterative Retriever, Reader, Reranker)
  - IRRR aims at building a **reasoning path** p from the question q, through all the necessary supporting documents or paragraphs $d \in \mathcal{D}_{gold}$ to the answer a.
  - IRRR operates in a **loop of retrieval, reading, and reranking** to expand the reasoning path with new documents from $d \in \mathcal{D}$.
  - More specifically, once a set of relevant documents are retrieved, they might either help answer the question, or reveal clues about the next piece of evidence.
  - The **reader model** then attempts to read each of the documents in $\mathcal{D}_1$ to answer the question combined with the current reasoning path p.
  - If answers are found, the answer with the **highest answerability score** is predicted as the answer, and if no answer is found, then IRRR's **reranker scores each retrieved paragraph** against the current reasoning path and appends the top-ranked paragraph to the current reasoning path.
  - *[CLS] question [SEP] title-1 [CONT] para-1 [SEP] … title-t [CONT] para-t [SEP].*

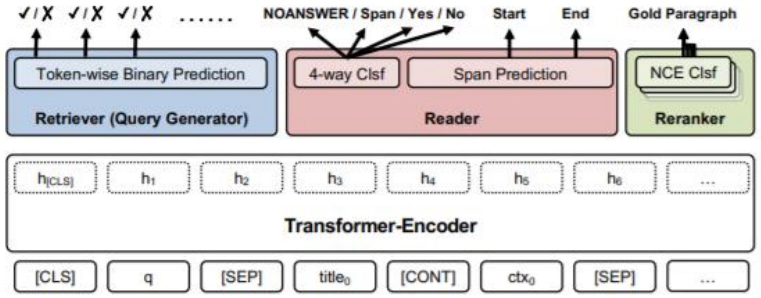


Figure 2: The overall architecture of our IRRR model, which uses a shared Transformer encoder to perform all subtasks of open-domain question answering.

Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]

- Methodology (Iterative Retriever, Reader, Reranker)
  - The goal of the **retriever** is to generate natural language queries to retrieve relevant documents from an off-the-shelf text-based retrieval engine(ElasticSearch).
  - Like **GOLDEN Retriever**, the authors **extract search queries** from the current reasoning path,
    because there is usually a strong **semantic overlap** between reasoning path and the next paragraph to retrieve.
  - To predict these search queries from the reasoning path, the authors apply a **token-wise binary classifier** on top of the encoder model,
    to decide whether each token is included in the final query.

  - The **reader** model attempts to find the answer given a reasoning path comprised of the question and retrieved paragraphs.
  - The reader is trained to predict one of 4 classes SPAN / YES / NO / NOANSWER, and span answers are predicted from the context using start, end classifier.
  - The authors further utilize NOANSWER to decide whether to continue the iterative process or not.

  - The **reranker** model selects one of the retrieved paragraphs to expand it, so that the retriever can generate new search queries to retrieve new context.
  - At training time, the reranker scores linearly transformed from CLS token is normalized with softmax,
    and is trained to maximize the log likelihood of selecting the gold supporting paragraphs from retrieved ones.

Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]

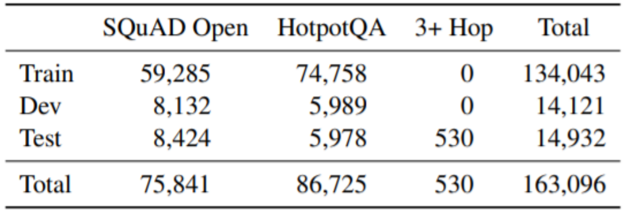
- Dynamic Oracle for Query Generation / Reducing Exposure Bias with Data Augmentation
    - Open Domain QA datasets do not include human-annotated search queries, so the authors derive **supervision signal** to train the retriever with a dynamic oracle.
    - Like **GOLDEN Retreiver**, the authors derive search queries from **overlapping terms** between the reasoning path and the target paragraph with the goal of maximizing retrieval performance.
    - Instead of enumerating all possible 2^N combinations, the authors use an **"importance" metric** as follows.

$$\mathrm{Imp}(s_i) = \mathrm{Rank}(t, \{s_j\}_{j=1, j\neq i}^{N}) - \mathrm{Rank}(t, \{s_i\}),$$

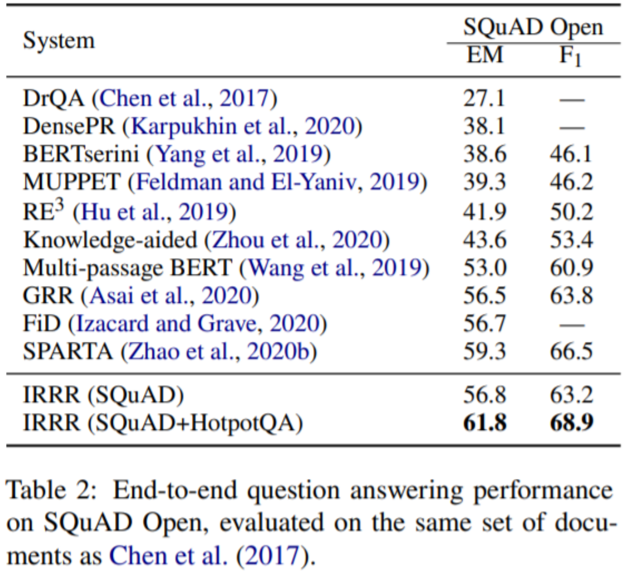
-
    - Intuitively, **the first term** captures its importance when combined with all other overlapping spans, while **the second term** captures the importance of the search term when used alone, which helps to capture query terms that are only effective when combined.
    - After estimating importance of each overlapping span, the authors determine the final oracle query by sorting all spans by descending importance. => O(N)
    - Additionally, to address the **exposure bias issue** (the model fails to generalize in cases where model behavior deviates from the oracle), the authors **augment training data** by occasionally selecting non-golden paragraphs to expand reasoning paths, and use the **dynamic oracle** to generate queries for model to recover from mistakes.

Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]

- Experiments
  - The authors test IRRR on SQuAD Open and HotpotQA along with the newly proposed benchmark BEERQA.
  - The authors use pretrained ELECTRA-Large model and train on a combined dataset of SQuAD Open and HotpotQA.

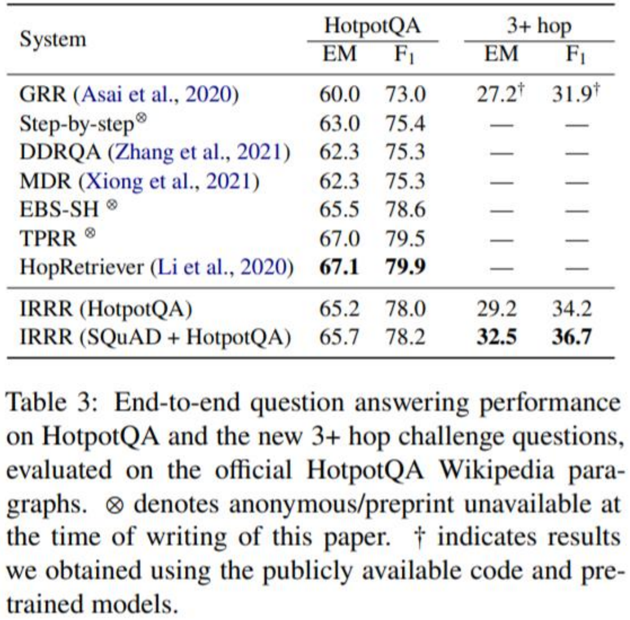


	SQuAD Open	HotpotQA	3+ Hop	Total
Train	59,285	74,758	0	134,043
Dev	8,132	5,989	0	14,121
Test	8,424	5,978	530	14,932
Total	75,841	86,725	530	163,096



System	SQuAD Open	
	EM	F$_1$
DrQA (Chen et al., 2017)	27.1	—
DensePR (Karpukhin et al., 2020)	38.1	—
BERTserini (Yang et al., 2019)	38.6	46.1
MUPPET (Feldman and El-Yaniv, 2019)	39.3	46.2
RE^3 (Hu et al., 2019)	41.9	50.2
Knowledge-aided (Zhou et al., 2020)	43.6	53.4
Multi-passage BERT (Wang et al., 2019)	53.0	60.9
GRR (Asai et al., 2020)	56.5	63.8
FiD (Izacard and Grave, 2020)	56.7	—
SPARTA (Zhao et al., 2020b)	59.3	66.5
IRRR (SQuAD)	56.8	63.2
IRRR (SQuAD+HotpotQA)	**61.8**	**68.9**

Table 2: End-to-end question answering performance on SQuAD Open, evaluated on the same set of documents as Chen et al. (2017).



System	HotpotQA		3+ hop	
	EM	F$_1$	EM	F$_1$
GRR (Asai et al., 2020)	60.0	73.0	$27.2^{\dagger}$	$31.9^{\dagger}$
Step-by-step$^{\otimes}$	63.0	75.4	—	—
DDRQA (Zhang et al., 2021)	62.3	75.3	—	—
MDR (Xiong et al., 2021)	62.3	75.3	—	—
EBS-SH $^{\otimes}$	65.5	78.6	—	—
TPRR $^{\otimes}$	67.0	79.5	—	—
HopRetriever (Li et al., 2020)	**67.1**	**79.9**	—	—
IRRR (HotpotQA)	65.2	78.0	29.2	34.2
IRRR (SQuAD + HotpotQA)	65.7	78.2	**32.5**	**36.7**

Table 3: End-to-end question answering performance on HotpotQA and the new 3+ hop challenge questions, evaluated on the official HotpotQA Wikipedia paragraphs. $\otimes$ denotes anonymous/preprint unavailable at the time of writing of this paper. † indicates results we obtained using the publicly available code and pretrained models.

Answering Open-Domain Questions of Varying Reasoning Steps from Text [EMNLP 2021]

- Experiments
  - IRRR stops its iterative process as soon as all necessary paragraphs to answer the question have been retrieved, effectively reducing the total number of paragraphs retrieved and read by the model compared to always retrieving a fixed number of paragraphs for each question.
  - Although it is still effective to increase the number of reasoning steps, IRRR is more effective than that of previous work.

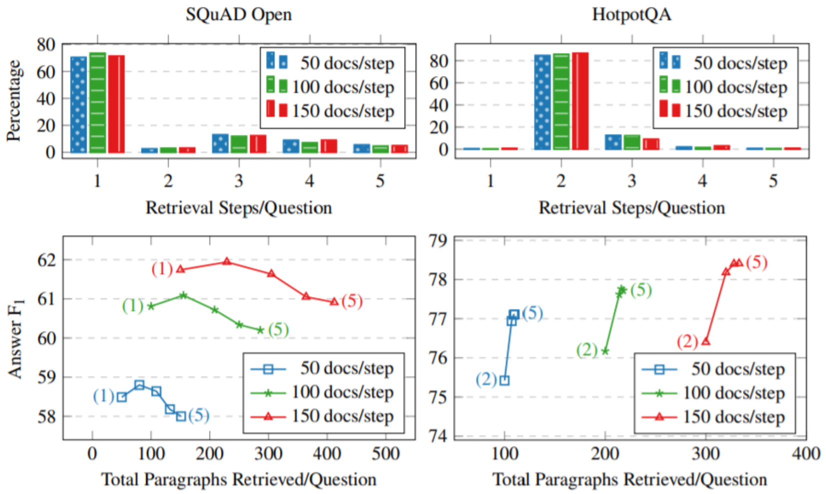


Figure 5: The retrieval behavior of IRRR and its relation to the performance of end-to-end question answering. Top: The distribution of reasoning path lengths as determined by IRRR. Bottom: Total number of paragraphs retrieved by IRRR vs. the end-to-end question answering performance as measured by answer F_1.

ANY QUESTIONS?