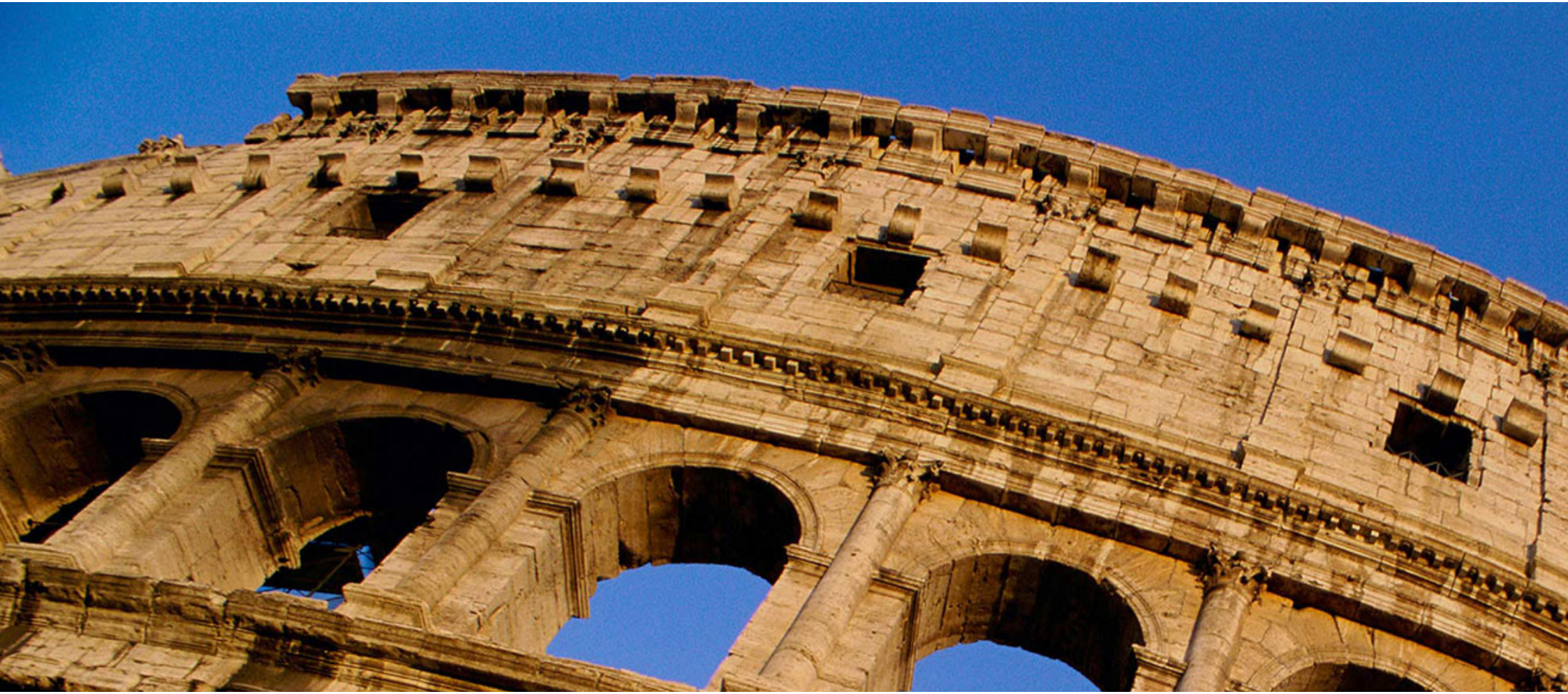
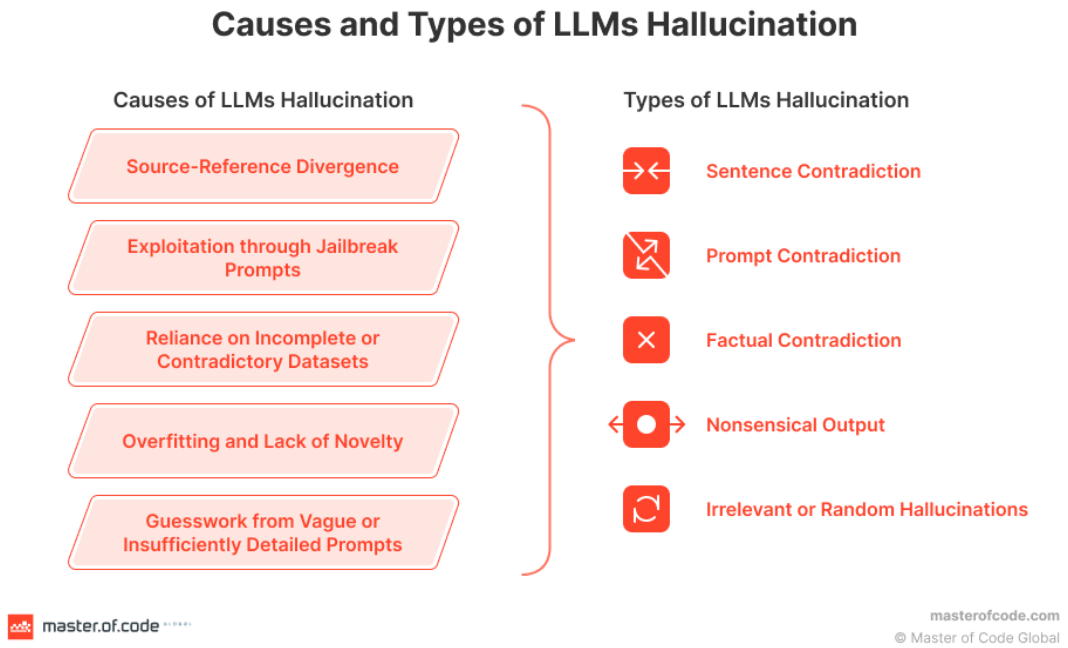


Retrieval Augmented LM - Attempts to improve retrieval

HCC Lab
Heo Dong Seok

Retrieval Augmented Generation

- LM 's big problem : Hallucination / Imaginary content



https://masterofcode.com/blog/hallucinations-in-llms-what-you-need-to-know-before-integration

Retrieval Augmented Generation

- RAG : Uses external docs/knowledge to generate output

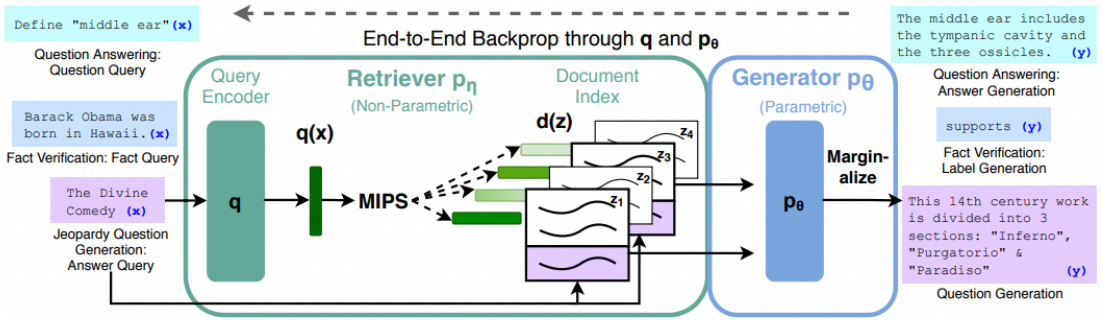


Figure 1: Overview of our approach. We combine a pre-trained retriever (*Query Encoder + Document Index*) with a pre-trained seq2seq model (*Generator*) and fine-tune end-to-end. For query x, we use Maximum Inner Product Search (MIPS) to find the top-K documents z_i. For final prediction y, we treat z as a latent variable and marginalize over seq2seq predictions given different documents.

Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks (2020)

https://arxiv.org/abs/2005.11401

Retrieval Augmented Generation

- **Step 1 : Retrieval**
  - Retrieve top k passages for given query from external knowledge base (Wiki, Bing, etc)
  - (ex) MIPS / BM25 / DPR
- **Step 2 : Generation**
  - Generate answer output with (query + passages) input
  - (ex) LLM (BART / GPT / LoRa ...)

Retrieval Augmented Generation - Step 1

- BM25 : sparse retriever

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{f(q_i,D) * (k_1+1)}{f(q_i,D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})}$$

문서 D에서 q_i의 term frequency

문서 D 의 길이

파라메터

문서 집합의 평균 문서 길이

- DPR : dense retriever

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right) \qquad \mathbf{d}(z) = \mathrm{BERT}_d(z), \quad \mathbf{q}(x) = \mathrm{BERT}_q(x)$$

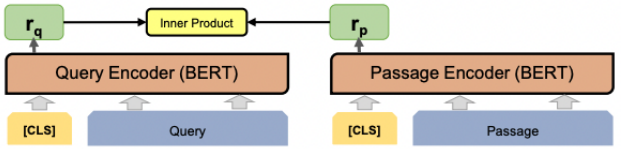


Figure 5: Representation Model for Initial Retrieval

q: query encoder / d: DPR
두 임베딩의 곱으로 부터 similarity 계산

Re2G (2022)

Re2G

- **Re**trieve, **Re**rank, **G**enerate
- RAG + Reranker Network

Re2G: Retrieve, Rerank, Generate (2022)

https://arxiv.org/abs/2207.06300

Re2G

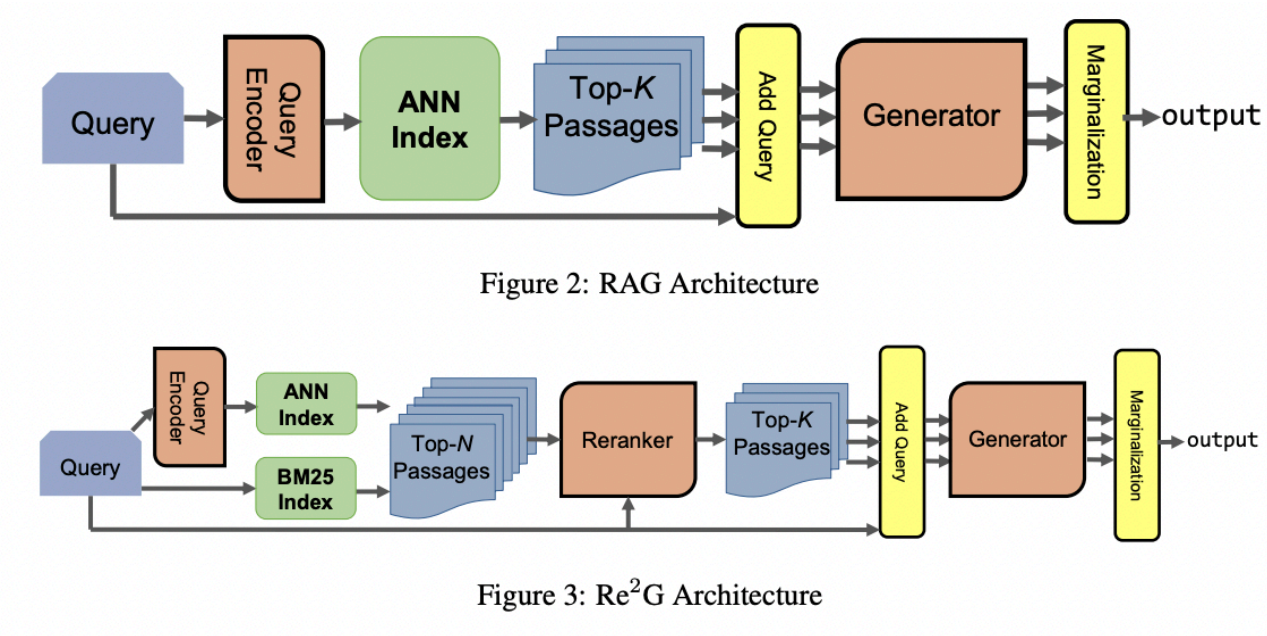


Figure 2: RAG Architecture

Figure 3: Re^{2}G Architecture

Rerank (BM25 passage ; ANN(DPR) passage)

Retrieval Augmented Generation - Step 1

- BM25 : sparse retriever ; good at out of domain data

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) * \frac{f(q_i,D) * (k_1+1)}{f(q_i,D) + k_1 * (1 - b + b * \frac{|D|}{avgdl})}$$

문서 D에서 q_i의 term frequency

문서 D 의 길이

파라메터

문서 집합의 평균 문서 길이

- DPR : dense retriever ; good at semantic info

$$p_\eta(z|x) \propto \exp\left(\mathbf{d}(z)^\top \mathbf{q}(x)\right) \qquad \mathbf{d}(z) = \text{BERT}_d(z), \quad \mathbf{q}(x) = \text{BERT}_q(x)$$

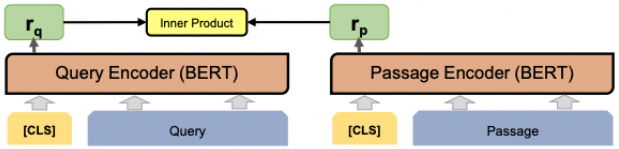


Figure 5: Representation Model for Initial Retrieval

Re2G

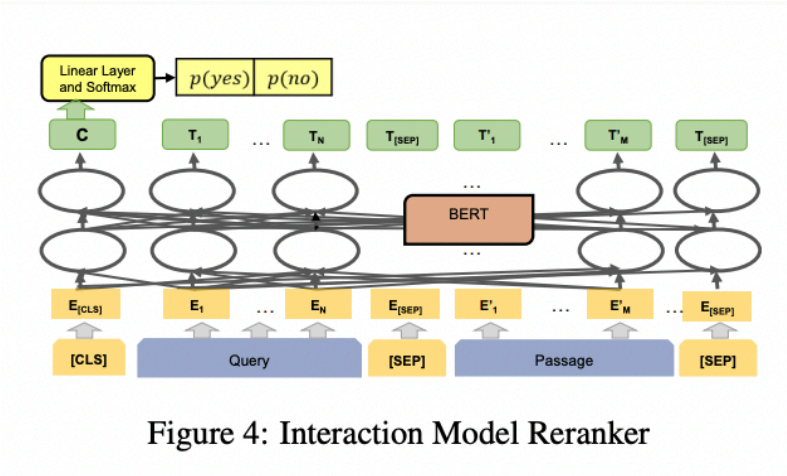


Figure 4: Interaction Model Reranker

Reranker architecture

Re2G - Training Scheme

- Training with Data <**q, t, Prov**> : query, target output, Provenance
  - 1. Training DPR : <**q, p+, p-**> triplet (**p+**: positive **Prov**, **p-**: negative **Prov**)
  - 2. Training Generator (BART) : <**q, t**>
  - 3. Training Reranker : <**q, P, Prov**> (P = BM25(q) U DPR(q))
  - 4. Training end-to-end : <**q,t**>

$$loss = -\sum_{i \in \mathbf{Prov}} log(softmax(\mathbf{z_r})_i)$$

z_r: reranker logit

Re2G - Training Scheme

- Training end-to-end
  - Using reranker passage instead of DPR passage : can't train query encoder
    - (does not use query encoder embedding in re-ranker)
  - Fix : Online knowledge distilation / Freezing query encoder

Re2G - Result

KILT IR benchmarks

	T-REx					(Slot Filling)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^2G (ours)	**80.70**	**89.00**	**87.68**	**89.93**	**75.84**	**77.05**
KGI_1 [Glass et al., 2021]	74.36	83.14	84.36	87.24	69.14	70.58
KILT-WEB 2 [Piktus et al., 2021]	75.64	87.57	81.34	84.46	64.64	66.64
SEAL [Bevilacqua et al., 2022]	67.80	81.52	83.72	86.53	60.08	61.72
KGI_0 [Glass et al., 2021]	59.70	70.38	77.90	81.31	55.54	56.79

	Natural Questions					(Question Answering)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^2G (ours)	**70.78**	**76.63**	51.73	60.97	**43.56**	**49.80**
SEAL [Bevilacqua et al., 2022]	63.16	68.19	**53.74**	**62.24**	38.78	44.40
KGI_0 [Glass et al., 2021]	63.71	70.17	45.22	53.38	36.36	41.83
KILT-WEB 2 [Piktus et al., 2021]	59.83	71.17	51.59	60.83	35.32	40.73
RAG [Petroni et al., 2021]	59.49	67.06	44.39	52.35	32.69	37.91

	TriviaQA					(Question Answering)
	R-Prec	**Recall@5**	**Accuracy**	**F1**	**KILT-AC**	**KILT-F1**
Re^2G (ours)	**72.68**	74.23	**76.27**	**81.40**	**57.91**	**61.78**
SEAL [Bevilacqua et al., 2022]	68.36	**76.36**	70.86	77.29	50.56	54.99
KILT-WEB 2 [Piktus et al., 2021]	58.85	71.55	72.73	79.54	45.55	49.57
KGI_0 [Glass et al., 2021]	60.49	63.54	60.99	66.55	42.85	46.08
MultiDPR [Maillard et al., 2021]	61.49	68.33	59.60	66.53	42.36	46.19

	FEVER				(Fact Checking)
	R-Prec	**Recall@5**	**Accuracy**		**KILT-AC**
Re^2G (ours)	**88.92**	**92.52**	**89.55**		**78.53**
SEAL [Bevilacqua et al., 2022]	81.45	89.56	89.54		71.28
KILT-WEB 2 [Piktus et al., 2021]	74.77	87.89	88.99		65.68
KGI_0 [Glass et al., 2021]	75.60	84.95	85.58		64.41
MultiDPR [Maillard et al., 2021]	74.48	87.52	86.32		63.94

	Wizard of Wikipedia					(Dialog)
	R-Prec	**Recall@5**	**Rouge-L**	**F1**	**KILT-RL**	**KILT-F1**
Hindsight [Paranjape et al., 2021]	56.08	74.27	**17.06**	**19.19**	**11.92**	**13.39**
Re^2G (ours)	**60.10**	**79.98**	16.76	18.90	11.39	12.98
SEAL [Bevilacqua et al., 2022]	57.55	78.96	16.65	18.34	10.45	11.63
KGI_0 [Glass et al., 2021]	55.37	78.45	16.36	18.57	10.36	11.79
RAG [Petroni et al., 2021]	57.75	74.61	11.57	13.11	7.59	8.75
KILT-WEB 2 [Piktus et al., 2021]	41.54	68.25	13.94	15.66	6.55	7.57

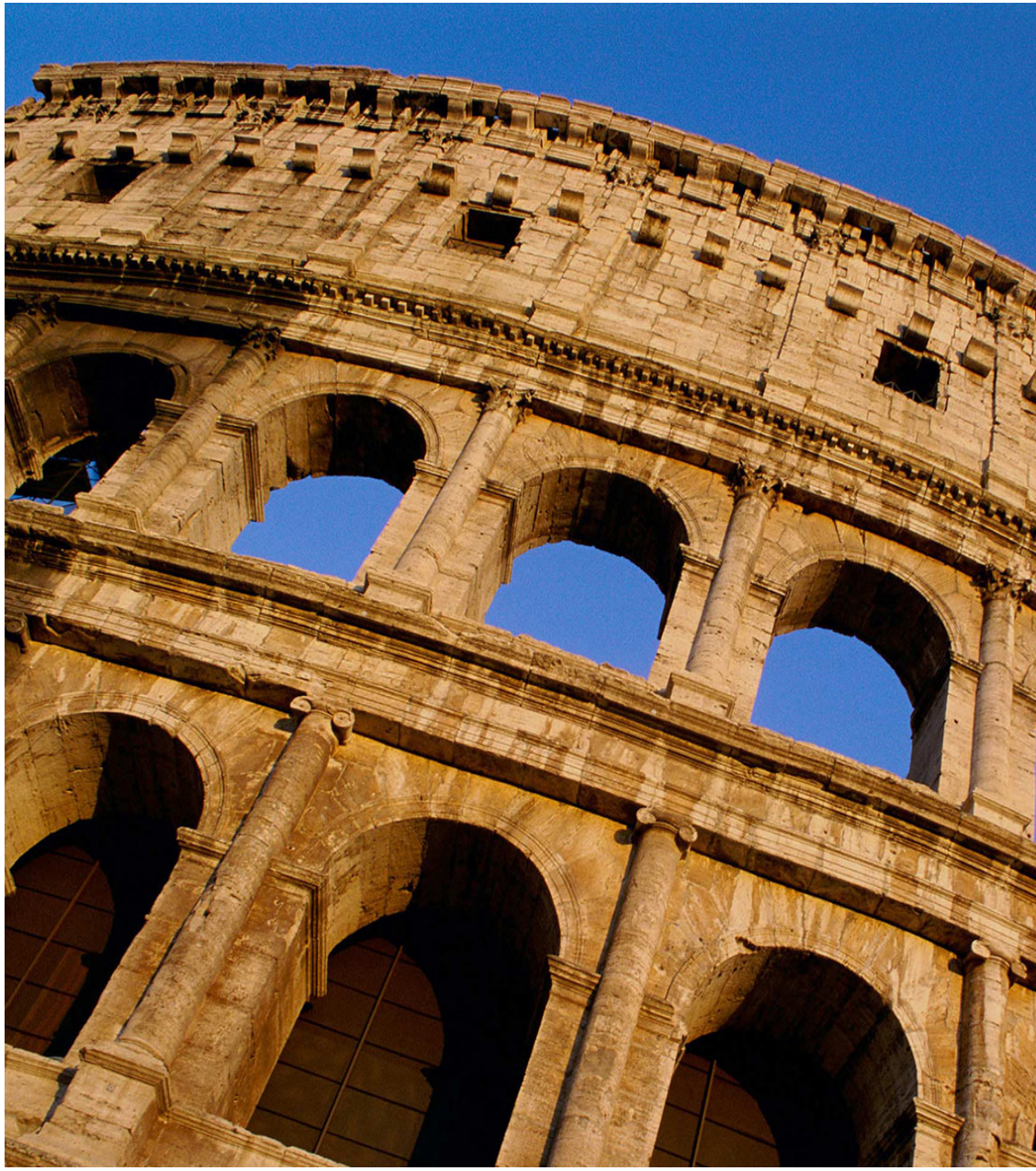
Table 1: KILT leaderboard top systems

R-Prec: Precision for R retrieved passages

Re2G - Conclusion

- Improvement
  - BM25, DPR 등 retrieval method들의 ensemble 방법 제시 및 성능 향상
- Limitation
  - Very long training scheme
  - Simple re-ranker network architecture

FLARE (2023)



FLARE

- **F**oward-**L**ooking **A**ware **Re**trieval augmented generation
- In short-term knowledge intensive task, augmented LM works great
- However, long-term knowledge intensive task augmented LM does not work well as original LM.

Active Retrieval Augmented Generation (2023)

https://arxiv.org/abs/2305.06983

FLARE

- RAG retrieve passage only **once**.
- In long-term complex task, one retrieval is not enough.
- Repeat (Retrieve - Generate) and model stops itself actively.

- Assumption : LLM tend to be well-calibrated (필요한 정보만 사용) / low term probability means lack of information. (단어 확률이 낮음 => 정보 부족 의미)

FLARE - Model

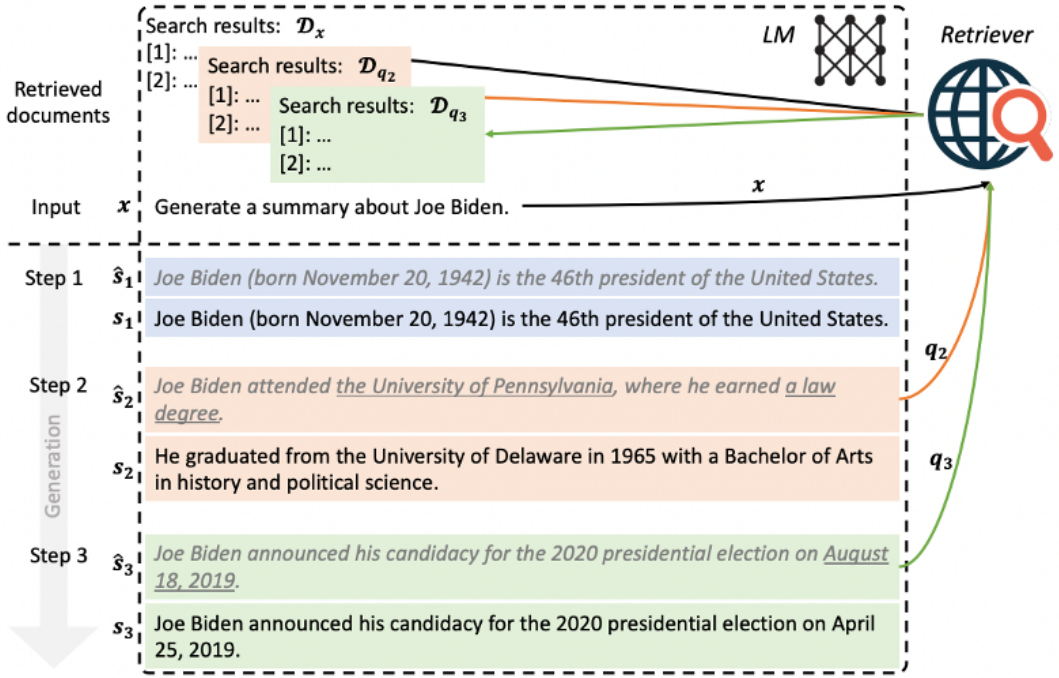


Figure 1: An illustration of forward-looking active retrieval augmented generation (FLARE). Starting with the user input $\boldsymbol{x}$ and initial retrieval results $\mathcal{D}_{\boldsymbol{x}}$, FLARE iteratively generates a temporary next sentence (shown in *gray italic*) and check whether it contains low-probability tokens (indicated with underline). If so (step 2 and 3), the system retrieves relevant documents and regenerates the sentence.

q_t = qry(x, y(<t))
D_t = ret(q_t)
y_t = LLM(D_t, x, y(<t))

if all y_t's term prob > threshold:
    end retrieval, use y_t as final output
else:
    t += 1
    repeat

FLARE - Model

FLARE_instruct : generator가 [Search (*query)] 를 출력 시 이를 그대로 retrieve에 사용.
(tend to not search than necessary : increase logit of "["
(excessive queries : after [Search (*query)], decrease logit of "[" for next few tokens)

FLARE_direct : 앞서 설명한대로 단순히 y_t를 q에 붙임.
대신 이경우 잘못된 y_t 정보가 들어갈 수 있어서, masking / self-ask QG with chatGPT로써 극복하고자함

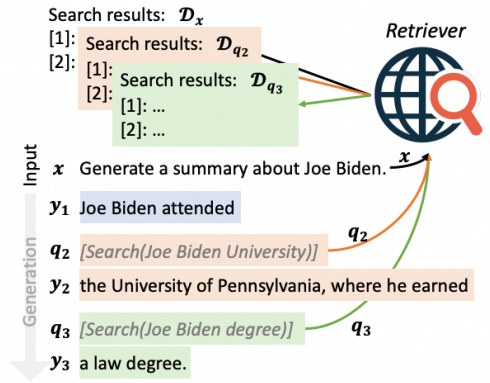


Figure 2: An illustration of forward-looking active retrieval augmented generation with retrieval instructions (FLARE$_{\text{instruct}}$). It iteratively generates search queries (shown in *gray italic*) to retrieve relevant information to aid future generations.

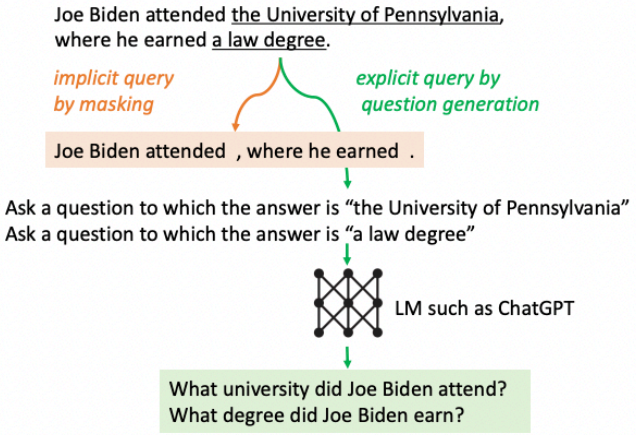


Figure 3: Implicit and explicit query formulation. Tokens with low probabilities are marked with underlines.

FLARE - Result

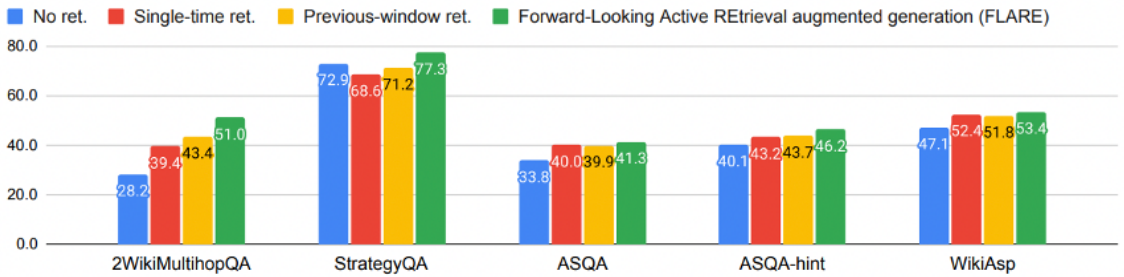


Figure 4: Comparision between FLARE and baselines across all tasks/datasets. We report the primary metric fo each dataset: EM for 2WikiMultihopQA, StrategyQA, and ASQA, and UniEval for WikiAsp.

MultihopQA / Commonsense reasoning / long-term QA (x2) / open domain summarization

EM: exact match UniEval : measure factual consistency.
Used generator baseline : GPT3.5

Methods	EM	F_1	Prec.	Rec.
No retrieval	28.2	36.8	36.5	38.6
Single-time retrieval	39.4	48.8	48.6	51.5
Multi-time retrieval				
Previous-window	43.2	52.3	51.7	54.5
Previous-sentence	39.0	49.2	48.9	51.8
Question decomposition	47.8	56.4	56.1	58.6
$\text{FLARE}_{\text{instruct}}$ (ours)	42.4	49.8	49.1	52.5
$\text{FLARE}_{\text{direct}}$ (ours)	**51.0**	**59.7**	**59.1**	**62.6**

Table 1: FLARE and baselines on 2WikiMultihopQA. Previous-window (Borgeaud et al., 2022; Ram et al., 2023), previous-sentence (Trivedi et al., 2022), and question decomposition (Press et al., 2022; Yao et al., 2022) methods are reimplemented for fair comparisons.

Baseline : window ret / sentence ret / question decomposition

FLARE - Result

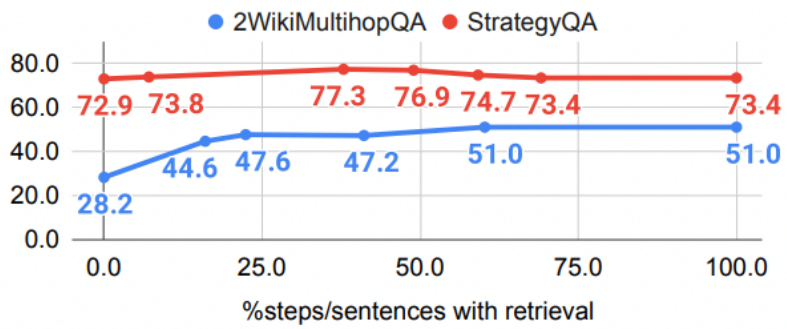


Figure 5: Performance (EM) of FLARE with respect to the percentage of steps/sentences with retrieval on 2WikiMultihopQA and StrategyQA.

- FLARE outperforms others.
- MultihopQA, long-term QA(w. hint) show big improvements
- FLARE_direct > FLARE_instruct
- too many retrieval is worse.

FLARE - Conclusion

- Improvement
  - 단어 확률을 바탕으로 스스로 retrieval 횟수 조정, 여러번의 retrieval로 성능 향상
- Limitation
  - 비정교한 threshold 설정
  - 다회 retrieval - generate의 반복으로 인한 cost 증가, overhead
  - 다회 retrieval passage 단순 합성 => 정보 손실 우려

Q&A