You Only Need One Model for Open-domain Question Answering

Haejun Lee[♠] **Akhil Kedia**[♠] **Jongwon Lee**[♠] **Ashwin Paranjape**[♣]
Christopher D. Manning[♣] **Kyoung-Gu Woo**[♡*]
♠ Samsung Research ♣ Stanford University ♡ Growdle Corporation
{haejun82.lee, akhil.kedia, jay722.lee}@samsung.com,
{ashwinp, manning}@cs.stanford.edu, epigramwoo@growdle.com

Abstract

Recent works for Open-domain Question Answering refer to an external knowledge base using a retriever model, optionally rerank the passages with a separate reranker model and generate an answer using an another reader model. Despite performing related tasks, the models have separate parameters and are weakly-coupled during training. In this work, we propose casting the retriever and the reranker as hard-attention mechanisms applied sequentially within the transformer architecture and feeding the resulting computed representations to the reader. In this singular model architecture the hidden representations are progressively refined from the retriever to the reranker to the reader, which is more efficient use of model capacity and also leads to better gradient flow when we train it in an end-to-end manner. We also propose a pre-training methodology to effectively train this architecture. We evaluate our model on Natural Questions and TriviaQA open datasets and for a fixed parameter budget, our model outperforms the previous state-of-the-art model by 1.0 and 0.7 exact match scores.

1 Introduction

Open-domain Question Answering (Open QA) is a knowledge-intensive task that finds the answer for the given question from a large-scale knowledge corpus that can easily surpass millions of documents. Thus, how to store and refer to the knowledge at such scales is important in terms of both performance and scalability for Open QA systems. Traditional systems rely on information retrieval engines such as Elastic Search. These score the relevance of knowledge to a given query by symbolic overlaps between them in a sparse representation space based on TF-IDF or BM25 (Chen et al., 2017; Wang et al., 2018; Yang et al., 2019). However, recent advances in neural language modeling have enabled two new lines of approach; 1) referring to internal knowledge parameterized in the model (Brown et al., 2020; Petroni et al., 2019; Roberts et al., 2020), and 2) referring to external knowledge retrieved by matching query and knowledge in dense representation spaces (Karpukhin et al., 2020; Lee et al., 2019; Guu et al., 2020; Lewis et al., 2020b; Izacard and Grave, 2021b).

Despite the simplicity of the approach, parametric models have limitations such as a large number of model parameters that require large compute for both training and inference and non-expandable knowledge without re-training. Their implicit knowledge reference also makes it hard to find supporting knowledge and often results in hallucinations (Shuster et al., 2021). On the other hand, the current dense retrieval models have advantages over parametric models on these issues (Karpukhin et al., 2020; Guu et al., 2020; Lewis et al., 2020b). But most retrieval models only have a weak coupling between and separate parameters for reader, reranker (if any), and retriever that limits these models from achieving optimal end-to-end training and efficient use of the total model capacity.

In this paper, we propose a single language model YONO (You Only Need One model) that can refer to external knowledge via its internal attention functions, which are trainable in a fully end-to-end manner. We achieve this by generalizing the retrieval and reranking as internal passage-wise attentions. At the lower retrieval layers, the query and passages are separately encoded allowing pre-computation of all the passage representations. Then passage-wise hard attention is applied to retrieve initial relevant passages from the entire knowledge base. While it would be optimal to retrieve passages based on soft-attention between the query and all the passages (Khattab and Zaharia, 2020), it is computationally intractable. Hence, we approximate this soft-attention by a passage-

[*] This work was done when the author was affiliated with Samsung Research.

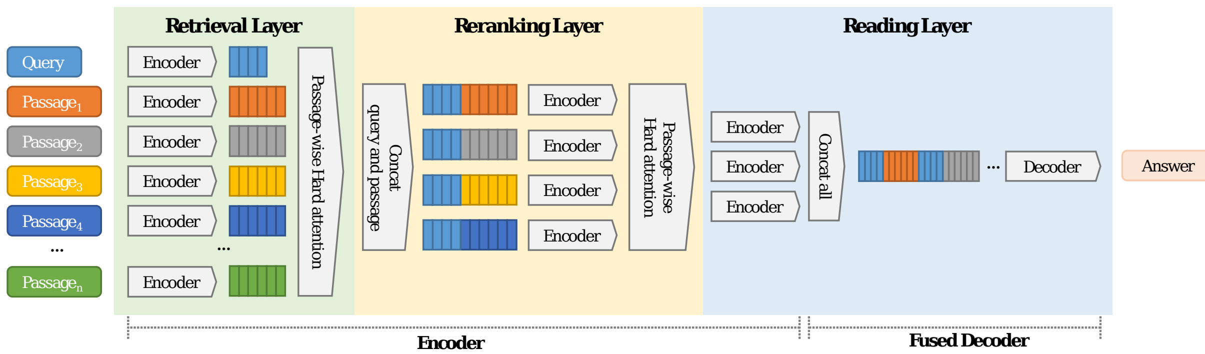

Figure 1: The overall architecture of our proposed model YONO.

wise hard-attention layer using decoupled query and passage representations. The representations of the initial relevant passages are further encoded jointly with the query representation to compute more expressive coupled representations. These are used to select only the more relevant passages using another hard-attention in the reranking layer. The representations of the final set of passages are then encoded by transformer encoders to get deeper representations that are fused in the decoder to generate the answers.

We train this architecture fully end-to-end by leveraging a training signal from the final answer generation decoder. By combining the retrieving, reranking, and reading into a single model, we achieve better gradient flow across these functions for fully end-to-end training and more efficient use of the model capacity.

Our contributions are twofold;

- A single model that generalizes retrieval, reranking, and reading as internal attention functions that achieves better utilization of the model parameters by sharing representations.
- A methodology to train this architecture in a fully end-to-end manner that allows better gradient flow across the whole model.

2 YONO Architecture

As depicted in Figure 1, we propose a single encoder-decoder language model architecture consisting of 3 components: Retrieval Layer, Reranking Layer, and Reading Layer.

2.1 Retrieval Layer

This layer retrieves the top-N relevant passages for a given query from the knowledge corpus. We first encode query and passage independently with the first K transformer encoder layers. The query is encoded with '*query*:' prefix while passages are encoded with '*title*:' and '*context*:' prefixes.

Passage-wise Hard-attention via Dot-product similarity (PHD): We apply passage-wise hard-attention to get activated passages from all the passages. Let q_0 and P_0 be the first tokens' representations of the encoded query and all the passages respectively. The attention scores $score_{phd}(q, P)$ are calculated by the dot-product attention function between q_0 and P_0 as:

$$\begin{aligned} Q_q &= LayerNorm(q_0 W_q) \\ K_P &= LayerNorm(P_0 W_p) \\ score_{phd}(q, P) &= \frac{Q_q K_P^T}{\sqrt{d_k}} \end{aligned} \tag{1}$$

where $W_q, W_p \in \mathrm{R}^{\mathrm{d \times d}}$ are learned linear projections. The scaling factor $1/\sqrt{d_k}$ does not affect the ordering of the retrieved passages, and is added to align the scale of this score with the soft-attention scores.

At training time, it is intractable to calculate the representations of all the passages P_0. Thus we stochastically approximate P_0 with a combination of the random negative passages P^N and the retrieved passages P^R. We obtain P^R from the indexed passage representations using Maximum Inner Product Search(MIPS) tools such as FAISS (Johnson et al., 2021) and iteratively updating P^R following previous approaches (Guu et al., 2020; Singh et al., 2021).

2.2 Reranking Layer

We further narrow down the passages by applying an additional hard-attention based on more expressive representations from the joint encoding of query and passages using the next L transformer encoder layers.

Passage-wise hard-attention via Cross-attention (PHC): First, we encode concatenated pairs of query and passage representations for P^R with cross-attention for more expressive representations, H. Then we apply another hard-attention using H_0, the representations of the first tokens in H, linearly projected by a learnt $W_{qp} \in \mathrm{R}^{\mathrm{d}\times 1}$ to calculate the score $score_{phc}(q, P^R)$:

$$
\begin{aligned}
H &= Transformer(q \oplus P^R) \\
score_{phc}(q, P^R) &= LayerNorm(H_0)W_{qp} \quad (2)
\end{aligned}
$$

2.3 Reading Layer

After the retrieval and reranking layers, the final representations are fed to the reading layer. These are further encoded using the remaining transformer encoder layers and fused in the decoder for multi-passage reading, following the approach of FiD (Izacard and Grave, 2021b).

Critically, our method uses a single model with internal attention functions, instead of having independent models in a pipeline. The advantages are 1) gradient level end-to-end optimization that will train the whole model from the final training signal from the decoder; 2) deeper model for a given number of total parameters by stacking retriever, reranker, and reader layers while sharing representations.

3 Training YONO

3.1 Training Objective

The whole model is always trained end-to-end by leveraging a training signal from the final answer generation. Due to the non-differentiability of hard attentions, attention approximation losses for training hard attentions are combined to the answer generation loss. The final loss is a sum of losses for generation $\mathcal{L}_{gen.}$ and attention approximations $\mathcal{L}^{phd}_{approx.}$ and $\mathcal{L}^{phc}_{approx.}$ as below:

$$\mathcal{L} = \mathcal{L}_{gen.} + \mathcal{L}^{phd}_{approx.} + \mathcal{L}^{phc}_{approx.} \quad (3)$$

Generation loss: We use a conventional auto-regressive language modeling loss for generating an answer a given a query q and retrieved passages P^R:

$$\mathcal{L}_{gen.} = -\log \prod_{t=1}^{T_A} p(a_t \mid a_{<t}, q, P^R) \quad (4)$$

Attention Approximation loss: The hard-attentions are trained to approximate the soft-attentions using KL-divergence between the soft-attention scores G_{soft} obtained from the decoder's attentions and hard-attention scores S_{hard}.

$$\mathcal{L}_{approx.} = D_{KL}(G_{soft} \| S_{hard}) \quad (5)$$

The soft-attention scores for each retrieved passage G^R are derived by accumulating decoder's attention scores using Formula 6. att_{dec}, 0, N_l, N_h, $N_{p,t}$, and SG denote decoder attention matrices toward encoded outputs, output token index, number of layers, attention heads, tokens in a given passage p, and stop gradient function respectively. This is similar to the cross-attention score proposed by FiD-KD (Izacard and Grave, 2021a), except that we only accumulate attention scores of passage tokens excluding query tokens. Since the decoder is trained simultaneously, the gradient flow back to the decoder's soft-attention scores is blocked by SG.

$$
\begin{aligned}
&score_{soft}(P) = \\
&\quad (\sum_{l=0}^{N_l}\sum_{h=0}^{N_h}\sum_{t_p=0}^{N_{p,t}} \frac{SG(att_{dec}(0, l, h, t_p))}{N_l N_h N_t} \mid p \in P)
\end{aligned} \quad (6)
$$

For the retrieval layer, soft attention scores of the random negative passages G^N are set to $-\infty$ to get zero probability after softmax normalization. Note that decoder's soft-attention scores for P^N are not available because these negative passages are never passed to the decoder. The final soft-attention scores G_{soft} are obtained by concatenating scores for P^R and P^N followed by the softmax normalization.

$$
\begin{aligned}
G^R &= score_{soft}(P^R) \\
G^N &= (-\infty \mid b \in P^N) \\
G_{soft} &= \sigma(G^R \oplus G^N) \quad (7)
\end{aligned}
$$

Hard-attention scores S^R and S^N for P^R and P^N are calculated by Formula 1, with a constant penalty γ to the scores of the random negative passages S^N. This penalty is an inductive bias to decrease scores of the negatives passages lower than the lowest retrieved passages.

$$S^R = score_{phd/phc}(q, P^R)$$
$$S^N = score_{phd}(q, P^N) + \gamma$$
$$S_{hard} = \sigma(S^R \oplus S^N) \quad (8)$$

Note that we do not use random negative passages for the reranking layer.

3.2 Pre-training corpus

We first pre-train our model to adapt the pre-trained encoder-decoder architecture to the YONO architecture and provide initial retrieval performance for fine-tuning without passage labels. Inverse Cloze Task (ICT) (Lee et al., 2019) and Masked Salient Span (MSS) (Roberts et al., 2020; Guu et al., 2020) are widely used tasks for pre-training. ICT uses '*input-passage*' pairs that have explicit supervision for training passage-wise attention, but has no supervision for the answer generation. On the other hand, MSS trains the model by '*input-output*' pairs that have strong supervision for the answer generation, but no supervision for retriever, requiring additional warm-up training for retrieval such as ICT. Thus, these tasks are sequentially applied to pre-train the pipeline models to overcome their limitations (Guu et al., 2020; Singh et al., 2021). However, because of our single model architecture, all the retrieval, reranking, and reading layers need to be trained at the same time, requiring triples of '*input-passage-output*' for pre-training. To provide such supervisions, we extend a masked salient span task with explicit passage labels, which we call Explicit Masked Salient Span.

Explicit Masked Salient Span (eMSS): We first pick one named entity and mask all instances of this entity from the sentence. We explicitly add a ground truth passage that contains the masked named entity from nearby passages except its original passage. We refine the data by simple heuristics such as using only pairs of sentence and target passage that contain at least 1 common named entity other than the masked span, and selecting the target passage with the highest number of common named entities when there are multiple passages containing the masked span. In this way, we generate 53M triples from the whole Wikipedia passages in total.

3.3 Training Procedure

The model is pre-trained and fine-tuned iteratively to refresh retrieved passages for a better approximation of the distribution over all the passages.

We start the first pre-training iteration using the initial pre-training data, extracted by the eMSS method that has one ground-truth passage for each query sentence. Note that with one positive passage per query, $\mathcal{L}^{phd}_{approx.}$ is equivalent to the negative log-likelihood loss of predicting the positive passage along with negative passages. However, $\mathcal{L}^{phc}_{approx.}$ does not yield any training signal at the first pre-training iteration because it can only learn from contrasting multiple retrieved passages.

From the second iteration, the model is trained with 100 passages fetched from the retrieval layer. We do not filter more passages at reranking layer during training to compute $score_{soft}$ to allow the reader to learn from the maximum number of passages. We pre-train the model for several iterations until the performance of the retrieval converges based on the recall metric.

After the pre-training, the model is then fine-tuned following the same procedure as that after the first iteration. To prevent an over-fitting of the reader due to limited size of the fine-tuning data, we once reinitialize the model after retrieval performance converges. Note that the model is fine-tuned by only weak-supervision of question-answer pairs without gold passages. Detailed hyper-parameters are described in Section 4.

4 Experiments

4.1 Model configurations

We primarily compare our model with baselines that use 440M parameters in total. To get a single language model with 440M parameters, we initialize our model from the pre-trained T5-large (Raffel et al., 2020) discarding 18 decoder layers. This results in our model with 24 encoder and 6 decoder layers. The retrieval layer uses the first 12 encoder layers that uses 25% less parameters than baselines' bi-encoders retrievers (165M vs 220M). Since our reranking layer works on the representations of the retrieval layer, we only allocate 4 encoder layers for reranking. The total number of parameters used for retrieval and reranking is 220M. The remaining 220M parameters are allocated for the reading layer.

4.2 Training details

At the first pre-training iteration, the model is trained with a batch of 800 *question-passage-answer* triples for 100K steps. From the second

Model	Passage Label	Aug. data	Retriever # Params	Natural Questions			TriviaQA		
				R@5	R@20	R@100	R@5	R@20	R@100
BM25 (Mao et al., 2021a)			-	43.6	62.9	78.1	67.7	77.3	83.9
DPR (Karpukhin et al., 2020)	√		220M	68.1	80.0	85.9	-	79.4	85.0
DPR^{new} (Karpukhin et al., 2020)	√		220M	72.2	81.3	87.3	-	-	-
GAR (Mao et al., 2021a)	√	√	220M	60.9	74.4	85.3	73.1	80.4	85.7
GAR^{+} (Mao et al., 2021a)	√	√	220M	70.7	81.6	88.9	76.0	82.1	86.6
PAIR (Ren et al., 2021)	√	√	220M	74.9	84.0	89.1	-	-	-
coCondenser (Gao and Callan, 2021)	√		220M	**75.8**	84.3	89.0	**76.8**	83.2	87.3
DPR-PAQ (Oguz et al., 2021)	√	√	220M	74.2	84.0	89.2	-	-	-
ANCE (Xiong et al., 2021a)	√		220M	-	81.9	87.5	-	80.3	85.2
FiD-KD (Izacard and Grave, 2021a)			220M	-	80.4	86.7	-	81.6	86.6
E2NR (Sachan et al., 2021)			220M	75.0	84.0	89.2	**76.8**	83.1	87.0
R2-D2$_{Retrieval}$ (Fajcik et al., 2021)	√		220M	68.6	80.6	86.7	69.8	78.9	84.7
Larger models									
E2NR (Sachan et al., 2021)			660M	76.2	84.8	89.8	78.7	84.1	87.8
DPR-PAQ (Oguz et al., 2021)	√	√	660M	76.9	84.7	89.2	-	-	-
YONO$_{Retrieval}$			165M	75.3	**85.2**	**90.2**	**76.8**	**83.5**	**87.4**
Reranking models									
GAR^{+}-BART (Mao et al., 2021b)	√		330M	73.5	82.2	-	-	-	-
GAR^{+}-RIDER (Mao et al., 2021b)	√		330M	75.2	83.2	88.9	77.9	82.8	85.7
R2-D2$_{Reranking200}$ (Fajcik et al., 2021)	√		330M	76.8	84.5	88.0	78.9	83.5	86.0
YONO$_{Reranking200}$			220M	79.1	**86.7**	90.7	82.1	86.0	88.1
YONO$_{Reranking800}$			220M	**79.1**	86.6	**91.1**	**82.3**	**86.4**	**88.7**

Table 1: Recall@N results on Natural Questions and TriviaQA test sets. The best retrieval and reranking scores except larger models are indicated in bold. Reranking200/800 refer to reranking the 200/800 retrieved passages.

iteration, we train the model for 1,250 steps per iteration using a batch with 64 *question-passages-answer* triplets where each triplet is packed with 100 retrieved passages. The maximum batch sizes given the available GPU memory were used. In total, we run 42 additional iterations after the first iteration for pre-training. After pre-training, the model is fine-tuned the same way as pre-training except it is trained for 1 epoch at every iteration. The model is optimized with Adam optimizer with a fixed learning rate 1^{-4}. Using 8 A100 GPUs, the first iteration takes 24 hours, and other iterations take around 5 hours each including MIPS index building and passage refresh. The penalty γ for hard-attention scores of the random negatives is set to 5 in all our experiments.

We found that answer generation more easily over-fits compared to the retrieval. To prevent this over-fitting, the model is once reinitialized from pre-trained YONO model after the model achieves acceptable recall on the training data at the 6^{th} iteration. We show the effect of the re-initialization in subsection 7.3.

4.3 Datasets

We evaluate our model with two standard open-domain question answering datasets, Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) following short answer subsets processed by Lee et al. (2019). Our external knowledge base is built using the Wikipedia dump from Dec. 20, 2018 where articles are split into passages of 200 tokens without overlap which is the same as datasets used in Izacard and Grave (2021b) for fair comparison.

5 Evaluation

5.1 Retrieval Performance

Table 1 and 2 show overall performances of our model and other baselines on Natural Questions and TriviaQA test sets. Our retrieval layer achieves the state-of-the-art recall@20/100 on Natural Questions regardless of model size, and the state-of-the-art recall@5/20/100 on TriviaQA among the models of the same size.

5.2 Reranking Performance

The reranking layer further improves the recall by more than 2 absolute points over the previous

Model	# Params	NQ	TQA
Discriminative models			
OrQA (Lee et al., 2019)	330M	33.3	45.0
REALM (Guu et al., 2020)	330M	40.4	-
ANCE (Xiong et al., 2021a)	330M	46.0	57.5
Generative models			
RAG (Lewis et al., 2020b)	440M	44.5	56.8
FiD (Izacard and Grave, 2021b)	440M	48.2	65.0
FiD-KD (Izacard and Grave, 2021a)	440M	49.6	68.8
E2NR (Sachan et al., 2021)	440M	45.9	56.3
EMDR^2 (Singh et al., 2021)	440M	52.5	71.4
Larger models			
FiD (Izacard and Grave, 2021b)	990M	51.4	67.6
FiD-KD (Izacard and Grave, 2021a)	990M	53.7	72.1
E2NR (Sachan et al., 2021)	1.4B	48.1	59.6
UnitedQA (Cheng et al., 2021)	1.87B	54.7	70.5
R2-D2 (Fajcik et al., 2021)	1.29B	55.9	69.9
YONO$_{Retrieval}$	440M	53.2	71.3
YONO$_{Reranking200}$	440M	53.2	71.5
YONO$_{Reranking800}$	440M	**53.2**	**71.9**

Table 2: End-to-end Open QA Exact-Match results on Natural Questions and TriviaQA test sets. Our model uses top 100 retrieved or reranked passages to generate answers. The best EM scores except larger models are indicated in bold.

state-of-the-art reranker model at Recall@5 in both datasets, achieving the best recall performance.

5.3 End-to-end Performance

Our model also results in the best end-to-end performance among the models of the same size on both datasets as shown in Table 2. Our best scores improve EM scores by 0.7 and 0.5 points on NQ and TQA respectively over the previously best performing model of the same size, EMDR^2 (Singh et al., 2021).

6 Ablation Studies

6.1 Shared representations on reader performance

Our reading layer uses 220M parameters but shares representations encoded by its preceding retrieval and reranking layers which use another 220M parameters. To measure gains of the shared representations, we compare our reader performance with that of a stand-alone reader model that uses the same number of parameters as our reading layers. For fair comparison, the stand-alone reader model is pre-trained and fine-tuned for the same amount of training tokens using the same retrieved data. Table 3 shows that the reader model sharing representations outperforms the stand-alone reader by 7.1% and 3.2% on NQ and TQA respectively.

Model	Natural Questions	TriviaQA
YONO Reader	51.4	70.0
Stand-Alone Reader	48.0	67.8
Δ	+3.4 (7.1%)	+2.2 (3.2%)

Table 3: Effect of sharing of retrieval and reranking representations on exact match scores of reader models that use 220M parameters on NQ and TQA development sets.

6.2 The generation loss on retrieval performance

The retrieval layer is trained by signals from both attention approximation and answer generation losses. While the attention approximation loss directly trains the retrieval scores, the answer generation loss is also a useful indirect training signal for the retriever. This signal is a key advantage of our approach over similar works such as Izacard and Grave (2021a); Singh et al. (2021).We evaluate the performance gain from the additional generation loss at the first pre-training iteration as shown in Table 4. The model trained with both losses shows significant improvement over the model trained with only the attention approximation loss. These relative gains are larger the fewer the number of retrieved passages. This result shows that answer generation loss is very effective for training the retriever.

Loss	R@5	R@20	R@100
$\mathcal{L}^{phd}_{approx.} + \mathcal{L}_{gen.}$	28.8	48.1	67.0
$\mathcal{L}^{phd}_{approx.}$	18.0	32.1	49.7
Δ	+10.8 (60.0%)	+16.0 (49.8%)	+18.7 (34.8%)

Table 4: Effect of generation loss on zero shot retrieval performance after the first iteration of pre-training on Natural Questions development set.

6.3 Effectiveness of eMSS pre-training

In this section, we show the effectiveness of our eMSS pre-training method. We evaluate this by fine-tuning our architecture without any pre-training and fine-tuning after the 1^{st} pre-training iteration. We also compare our pre-training to that of the data augmentation (Mao et al., 2021a; Ren et al., 2021; Oguz et al., 2021). We generate 'question-answer' pairs from a Wikipedia dump using a question and answer generation model trained on the

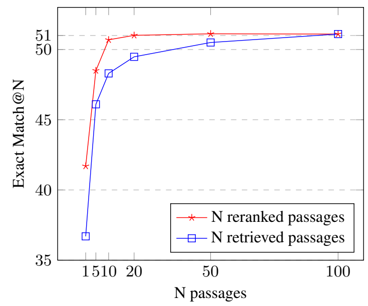

Figure 2: Exact Match scores for given N retrieved or reranked passages on NQ development set. Rerank EM scores are from reranking only 100 retrieved passages.

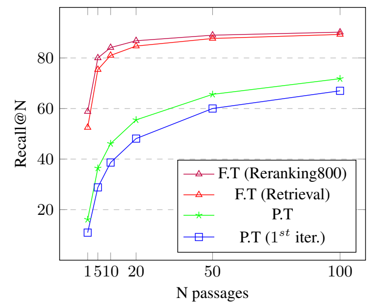

Figure 3: Recall@N at each training stage on NQ development set. P.T denotes pre-training, F.T denotes fine-tuning.

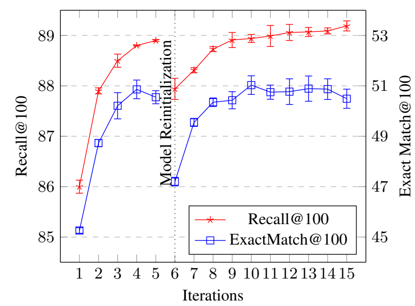

Figure 4: Average Recall and Exact Match scores at each fine-tuning iteration with error bars from 3 runs on NQ development set. The model is once reinitialized at the 6^{th} iteration.

NQ dataset using ASGen approach (Back et al., 2021). The model is further trained after the pre-training by this augmented data for 15,000 steps before fine-tuning.

Table 5 shows retrieval, reranking, and reading performance on Natural Questions and TriviaQA test sets. Our eMSS pre-training dramatically boosts the performance of our architecture by 9 EM points on NQ even with only the first pre-training iteration. Further iterations of our pre-training improves EM by 1 EM point. The further data augmentation pre-training improves performance on NQ consistently but only slightly, while the improvements on TQA is inconsistent, as the data was generated by the model trained on NQ dataset. These results clearly demonstrate that our simple self-supervised eMSS pre-training is strong enough to compete favourably against sophisticated approaches and yields good performance with only 1 pre-training iteration.

	Natural Questions			TriviaQA		
Pre-training	**R@20**	**R@100**	**EM**	**R@20**	**R@100**	**EM**
Retrieval						
No-pretrain	72.3	82.2	42.4	-	-	-
eMSS(1^{st} it.)	84.5	89.8	52.1	-	-	-
eMSS	85.2	90.2	53.2	**83.5**	87.4	**71.3**
eMSS+ASGen	**85.5**	**90.3**	**53.5**	**83.5**	**87.5**	70.9
Reranking 200						
No-pretrain	79.9	84.5	43.5	-	-	-
eMSS(1^{st})	86.3	90.3	52.1	-	-	-
eMSS	86.7	90.7	53.1	86.0	88.1	**71.5**
eMSS+ASGen	**87.2**	**90.9**	**53.2**	**86.2**	**88.2**	71.2

Table 5: Effect of further pre-training using augmented data on Natural Questions and TriviaQA test sets.

7 Analysis

7.1 Computational Efficiency of Reranking

In many dense retrieval systems, a reranker is often omitted due to functional overlaps with the reader and computational overhead (Guu et al., 2020; Lewis et al., 2020b; Singh et al., 2021). Thanks to the shared representations across the reader and reranker, our model can incorporate a reranking function without significantly more parameters or computation. By dropping irrelevant passages early at the reranking layer, we can achieve better computational efficiency. Figure 2 shows exact match scores for given N retrieved or reranked passages. The model still achieves optimal EM performance with only the top 20 reranked passages reranked from 100 retrieved passages. Reranking 100 to 20 passages can reduce the inference computation by 27.4% without pre-computed passage representations, and 54.0% with pre-computed passage representations without losing end-to-end performance.

7.2 Effect of pre-training iterations on retrieval performance

Figure 3 shows recall@N at each training stage across the pre-training and fine-tuning using Natural Question development set. The first iteration of the pre-training results in zero-shot recall@100 of 67.0%, that is further improved by additional pre-training iterations to 71.8% recall@100. These zero-shot recall scores enable us to fine-tune our model without passage labels result in the state-of-the-art retrieval performance.

7.3 Model re-initialization during fine-tuning

One caveat of sharing representation for retrieval and answer generation is that these show different over-fitting tendencies during fine-tuning where the training data is limited. We found that answer generation over-fits more easily compared to the retrieval. Answer generation relies on more expressive representation via cross attention, which may make it easier to memorize the output and hence more vulnerable to over-fitting. Furthermore, at the first fine-tuning iteration, the model is trained by zero-shot retrieval results from the pre-trained model that have relatively low recall rate and can harm the answer generation training. To refresh the over-fitted answer generation parameters, and to start from training data with high recall rate, we simply re-initialize the model with the pre-trained YONO model at the 6^{th} fine-tuning iteration.

Figure 4 shows retrieval and end-to-end performance at each fine-tuning iteration. Exact Match score drops after the 4^{th} iteration while retrieval score keeps increasing. After re-initializing the model at 6^{th} iteration, the model starts with higher recall and EM score. However, the EM score drops again from the 10^{th} iteration after achieving the best end-to-end performance, while the retrieval performance continues to improve. We leave further approaches for preventing over-fitting of our model such as freezing the model partially as future work.

8 Related Works

Parametric models for Knowledge Intensive Tasks: Large Scale Language Models trained on large corpora such as GPT (Brown et al., 2020), T5 (Raffel et al., 2020), and BART (Lewis et al., 2020a) store knowledge implicitly in their parameters. They have been used to generate correct responses by referring to their internal knowledge, such as question-answering or conversation, as shown in Brown et al. (2020), Petroni et al. (2019), Roberts et al. (2020), Adiwardana et al. (2020) and Roller et al. (2021).

These models often need a large number of model parameters to internalize this knowledge, requiring large compute and storage. The knowledge stored in these models also cannot be updated easily without an expensive re-training of the model. More importantly however, these language models often suffer from Hallucinations (Shuster et al., 2021; Maynez et al., 2020; Zhou et al., 2021; Zellers et al., 2019; Roller et al., 2021; Mielke et al., 2020), generating factually incorrect but plausible-sounding outputs.

Neural Retriever augmented Language Modeling: Augmenting language models with neural retrieval has been shown to be very effective, such as by retrieving nearest neighbor words for LM tasks (Khandelwal et al., 2020; Yogatama et al., 2021) or Machine Translation (Khandelwal et al., 2021). Dinan et al. (2019) proposed a decomposed transformer for conversation tasks, which enabled pre-computation of the embeddings over the large external knowledge.

ORQA (Lee et al., 2019) proposed the ICT task to pre-train a decomposed retriever, and DPR (Karpukhin et al., 2020) enhanced this approach with in-batch negatives and hard negatives to eliminate the pre-training. Synthetic Data Augmentation is also commonly used, such as in DPR-PAQ (Oguz et al., 2021), PAIR (Ren et al., 2021), Hu et al. (2021). Per-token embeddings, or multiple embeddings per passage were used in ColBERT (Khattab et al., 2020), ME-BERT (Luan et al., 2021), Lin et al. (2021), Lee et al. (2021).

Similar to our approach of re-ranker on top of a shared retriever, PreTTR (MacAvaney et al., 2020) pre-computed term representations for all documents, and used these to run only the upper layers of a transformer reranker model. Decoupled Transformer (Elfdaeel and Peshterliev, 2021) also shares the lower layers of a transformer encoder to serve as a reranker, using the upper layers as a reader, but lacks retriever and decoder, and focuses on computationally efficient reranking.

E2E optimization of NRALM: It is intractable to re-compute the embeddings of the knowledge for every weight update. REALM (Guu et al., 2020) and ANCE (Xiong et al., 2021a) proposed async index refresh to propagate updates to the index to yield better negatives. TAS (Hofstätter et al., 2021) and Xiong et al. (2021b) used clustering of embeddings for the same. RAG (Lewis et al., 2020b) used DPR retriever with BART generator to marginalize over generated tokens, which is back-propagated to the retriever. REALM++ (Balachandran et al., 2021) added a re-ranker to REALM.

Similar to our work, Bruyn et al. (2020) and TREAD (Shuster et al., 2021) utilize BART and T5 reader's encoders as a retriever, but unlike our work they do not have a unified pre-training method to train all the components of the model. Furthermore, these models also lack our integrated re-ranker, and the query and passage are also not cross-encoded for more expressive representations.

Multi-passage readers: Reading multiple passages at the same time is difficult, as concatenating multiple passages increases computation quadratically for transformers. Zhao et al. (2020) reduced multiple passages and sentences to few via a knowledge selector, which were then concatenated and passed on to GPT. FiD (Izacard and Grave, 2021b) concatenated the encoded representations of documents, which can then be attended by the decoder, achieving large performance gains. This approach was also applied in RocketQA (Qu et al., 2021). UnitedQA (Cheng et al., 2021) and R2D2 (Fajcik et al., 2021) combine results from an ensemble of extractive and generative readers, whereas PAQ (Lewis et al., 2021) directly retrieves possible answers with an FiD fallback.

Similar to our work, both FiD-KD (Izacard and Grave, 2021a) and EMDR^2 (Singh et al., 2021) trains the retriever end-to-end with a signal from the reader. Unlike these approach, our model has shared lower layers for more effective utilization of model parameters and better end-to-end gradient flow across the whole model. Furthermore, our training methodology results in propagating the answer generation loss the retriever, which has a large effect on improving performance as we show in Table 4.

9 Conclusion

In this paper, we propose a novel language model architecture that embeds the retriever and the reranker as hard-attention mechanisms and a training method to effectively train this model. In this singular model architecture efficiently uses a model capacity by cascading and sharing the representations from retriever to reranker to the reader that leads to better gradient flow for an end-to-end training. We evaluate our model on Natural Questions and TriviaQA open datasets and for a fixed parameter budget, our model outperforms the previous state-of-the-art model by 1.0 and 0.7 exact match scores. We show detailed ablations and analysis of each component of our approach. Our future work is to conduct more experiments on various knowledge intensive tasks and extend this model to match query and passage in multiple or hierarchical representation spaces.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977.

Seohyun Back, Akhil Kedia, Sai Chetan Chinthakindi, Haejun Lee, and Jaegul Choo. 2021. Learning to generate questions by learning to recover answer-containing sentences. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1516–1529, Online. Association for Computational Linguistics.

Vidhisha Balachandran, Ashish Vaswani, Yulia Tsvetkov, and Niki Parmar. 2021. Simple and efficient ways to improve REALM. *CoRR*, abs/2104.08710.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2020. BART for knowledge grounded conversations. In *Proceedings of the KDD 2020 Workshop on Conversational Systems Towards Mainstream Adoption co-located with the 26TH ACM SIGKDD Conference on Knowledge Discovery and Data Mining (SIGKDD 2020), Virtual Workshop, August 24, 2020*, volume 2666 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2021. UnitedQA: A hybrid approach for open domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3080–3090, Online. Association for Computational Linguistics.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Haytham Elfdaeel and Stanislav Peshterliev. 2021. Decoupled transformer for scalable inference in open-domain question answering. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Held Online, 1-3September, 2021*, pages 386–393. INCOMA Ltd.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-D2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 854–870. Association for Computational Linguistics.

Luyu Gao and Jamie Callan. 2021. Condenser: a pre-training architecture for dense retrieval. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Retrieval augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pages 113–122. ACM.

Ziniu Hu, Yizhou Sun, and Kai-Wei Chang. 2021. Relation-guided pre-training for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3431–3448. Association for Computational Linguistics.

Gautier Izacard and Edouard Grave. 2021a. Distilling knowledge from reader to retriever for question answering. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Gautier Izacard and Edouard Grave. 2021b. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2020. Relevance-guided supervision for openqa with colbert. *CoRR*, abs/2007.00814.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021. Phrase retrieval learns passage retrieval, too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 3661–3672. Association for Computational Linguistics.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Patrick S. H. Lewis, Yuxiang Wu, Linqing Liu, Pasquale Minervini, Heinrich Küttler, Aleksandra Piktus, Pontus Stenetorp, and Sebastian Riedel. 2021. PAQ: 65 million probably-asked questions and what you can do with them. *CoRR*, abs/2102.07033.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173, Online. Association for Computational Linguistics.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Trans. Assoc. Comput. Linguistics*, 9:329–345.

Sean MacAvaney, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto, Nazli Goharian, and Ophir Frieder. 2020. Efficient document re-ranking for transformers by precomputing term representations. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 49–58. ACM.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021a. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4089–4100, Online. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021b. Reader-guided passage reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 344–350, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sabrina J. Mielke, Arthur Szlam, Y-Lan Boureau, and Emily Dinan. 2020. Linguistic calibration through metacognition: aligning dialogue agent responses with expected correctness. *CoRR*, abs/2012.14983.

Barlas Oguz, Kushal Lakhotia, Anchit Gupta, Patrick S. H. Lewis, Vladimir Karpukhin, Aleksandra Piktus, Xilun Chen, Sebastian Riedel, Wen-tau Yih, Sonal Gupta, and Yashar Mehdad. 2021. Domain-matched pre-training tasks for dense retrieval. *CoRR*, abs/2107.13602.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. 2021. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Ruiyang Ren, Shangwen Lv, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. PAIR: Leveraging passage-centric similarity relation for improving dense passage retrieval. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2173–2183, Online. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the*

2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 5418–5426, Online. Association for Computational Linguistics.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6648–6662, Online. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 3784–3803. Association for Computational Linguistics.

Devendra Singh, Siva Reddy, Will Hamilton, Chris Dyer, and Dani Yogatama. 2021. End-to-end training of multi-document reader and retriever for open-domain question answering. *Advances in Neural Information Processing Systems*, 34.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018. Evidence aggregation for answer re-ranking in open-domain question answering. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021a. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Wenhan Xiong, Hong Wang, and William Yang Wang. 2021b. Progressively pretrained dense corpus index for open-domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2803–2815, Online. Association for Computational Linguistics.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Trans. Assoc. Comput. Linguistics*, 9:362–373.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9051–9062.

Xueliang Zhao, Wei Wu, Can Xu, Chongyang Tao, Dongyan Zhao, and Rui Yan. 2020. Knowledge-grounded dialogue generation with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3377–3390, Online. Association for Computational Linguistics.

Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. Detecting hallucinated content in conditional neural sequence generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.