Passage-Mask: A Learnable Regularization Strategy for Retriever-Reader Models

Shujian Zhang **Chengyue Gong** **Xingchao Liu**
The University of Texas at Austin
{szhang19, cygong, xcliu}@utexas.edu

Abstract

Retriever-reader models achieve competitive performance across many different NLP tasks such as open question answering and dialogue conversations. In this work, we notice these models easily overfit the top-rank retrieval passages and standard training fails to reason over the entire retrieval passages. We introduce a learnable passage mask mechanism which desensitizes the impact from the top-rank retrieval passages and prevents the model from overfitting. Controlling the gradient variance with fewer mask candidates and selecting the mask candidates with one-shot bi-level optimization, our learnable regularization strategy enforces the answer generation to focus on the entire retrieval passages. Experiments on different tasks across open question answering, dialogue conversation, and fact verification show that our method consistently outperforms its baselines. Extensive experiments and ablation studies demonstrate that our method can be general, effective, and beneficial for many NLP tasks.

1 Introduction

Retriever-reader based approaches are popularly considered in the knowledge-intensive tasks ($e.g.$, open Question Answering (QA), fact verification). It is designed to retrieve a set of support documents and extract the answer from these documents. Mostly adopted retrieve and read models ($e.g.$, Izacard and Grave, 2020) are trained to generate the annotated gold answers using the reader model, based on passages obtained by the retrievers ($e.g.$, Robertson and Zaragoza, 2009; Karpukhin et al., 2020). This training process of reader disregards the evidentiality of all retrieval passages and can easily overfit the top ranked passages (Xu et al., 2021; Lee et al., 2021). Even if the top-rank passages in the test setting do not have the correct answers, these models still tend to find the answer in the top-rank passages and yield worse performance (Xu et al., 2021). It happens to the reader model due to the overfitting and the memorization of outdated information (Longpre et al., 2021).

To what extent does the reader model quality depend on the retrieval passages? We analyze the ranking impact of the retrieval passages from masking ($e.g.$ mask out the top three passages), permuting, and removing. The overfitting, as well as the performance degradation, is observed. To desensitize the impact from the top-rank passages, we consider masking passages during training which serves as a desensitizer and can improve the reader model ability to reason over all retrieval passages.

However, the standard masking and dropout strategies are not designed for our focused tasks and also bring an increased gradient variance during training due to their randomness. In the meantime, each neuron plays the same role and has the same mask. However, in the reader model, intuitively, top-rank passages often have a higher chance to overfit during the training. To this end, we introduce our passage mask (PM), which encourages to mask top-rank passages. Reducing the gradient variance with fewer mask candidates and optimizing the mask candidates with bi-level optimization, the mask magnitude for each candidate can be learned. Overall, the proposed mask parameters are jointly optimized with the entire network.

We run extensive experiments across representative knowledge-intensive tasks: open-domain QA (Natural Questions Open (Kwiatkowski et al., 2019); TriviaQA unfiltered (Joshi et al., 2017)), fact verification (FaVIQ (Park et al., 2021)), and knowledge grounded dialogue (Wizard pf Wikipedia (Dinan et al., 2018)). Our method shows large performance improvements across different tasks and datasets. Furthermore, we provide extensive ablation studies on different design choices for the proposed method, including the designs of masking candidate space and efficiency. Our analysis shows the passage mask contributes the performance improvement, helping the reader learn to focus on

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3931–3943
December 7-11, 2022

the retrieval passages without being distracted by high-ranked passages with more lexical overlaps. With little modification, our regularization can be easily applied to other NLP tasks for a better answer generation strategy. To the best of our knowledge, we present the first mask regularization in the open retriever-reader setting by preventing the rank-related overfitting in Open QA, dialogue conversation, and fact verification. Our contributions are summarized as follows:

- Demonstrate that current models, *e.g.*, Fusion-in-Decoder (Izacard and Grave, 2020), tend to find answers in top-rank passages. These models are neither robust to passage drop nor able to utilize the entire retrieval passages.
- Present a passage mask mechanism for retrieval reader models. It improves the model generalization and encourages the model to extract answers from all the passages.
- Propose an efficient and effective way to train the model and the mask hyper-parameters jointly, which can one-shot search passage mask hyper-parameters. First, we use smaller number of mask candidates to reduce training gradient variance. Second, we jointly optimize the model parameters and mask candidate choices (*a.k.a.*, parameters) with theoretically-converged bi-level optimization.
- Verify the effectiveness and general applicability of the proposed method in knowledge intensive NLP tasks, $e.g.$, open question answering, fact verification, and dialogue tasks, and provide a rich analysis of this method with various design choices such as the masking position and efficiency. The proposed strategy can be easily incorporated or extended to many other NLP tasks.

2 Method

2.1 Knowledge-intensive Tasks

Knowledge-intensive tasks ($e.g.$, open QA, dialogue conversations) require to access a large body of retrieval information. A retrieval-augmented generation framework such as Fusion-in-Decoder (FiD) (Izacard and Grave, 2020) that consists of two components: a retriever model R and a generator model G has demonstrated the competitive performance and scalability to the large collection of retrieval evidence. FiD uses Dense Passage Retrieval (DPR) (Karpukhin et al., 2020) to retrieve a set of documents, and the decoder attends over the concatenation of all encoded document representations to generate the final answer. Specifically, the retriever model R is trained to retrieve a set of passages P with the highest top K relevance score for each training query. G is then trained to generate the final output $\hat{y}$ given an input query x and the top retrieved passages: $\hat{y} = G(x, P)$.

Mask Position	1st	2nd	3rd	4th	5th	FiD
Mask 1st	✓					44.5
Mask 2nd		✓				48.8
Mask 3rd			✓			48.3
Mask 4th				✓		49.1
Mask 5th					✓	49.6
Mask Top 5	✓	✓	✓	✓	✓	35.7
N/A						50.1

Table 1: Examples of the trained FiD (Izacard and Grave, 2021) reader model on Natural Questions Open (Kwiatkowski et al., 2019) where the top-rank retrieval passages are masked based on the mask position and the reader generates the answer from non-mask passages.

Although FiD does not use the unnormalized passage score as DPR, we still find out that FiD has a preference over passages with higher retrieval passage scores. Our analysis in Table 1[1] shows that G trained in this manner overfits the passages ranked high by the retriever. In this work, our goal is to prevent the overfitting, extract the answers in all given passages and improve the model generalization during the reader training.

2.2 Reader Model

The overall FiD reader model is composed of the encoder and the answer generator.

Encoder. Each retrieved passage and its title are concatenated with the question and processed independently by the encoder. We add tokens *question:*, *title:* and *context:* before the question, title and text of each passage. The input query x is prepended to each passage (Asai et al., 2021). The encoder is usually a pre-trained T5 (Raffel et al., 2020).

Answer Predictor. Mark $\mathbf{h}$ as a summary representation of the input, formed by concatenating the final-layer hidden state of passages. $\mathbf{h}$ is fed into the answer predictor and the final answer is autoregressively output.

Objective. In the encoder-decoder structure, we train the answer generator G given the originally

[1]Detailed discussions are in Section 4.1

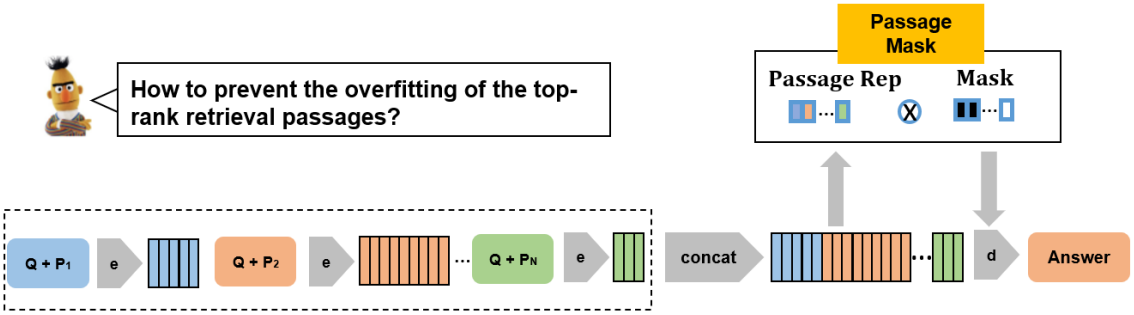

Figure 1: Overview of passage mask. Some notations are labeled along with corresponding components. 'Passage Rep' refers to the passage representation, 'Mask' refers to the mask, 'e' refers to the encoder, 'd' refers to the decoder, and 'Q + P_1' refers to the question and the first passage. In the Mask, the black color represents the mask and the white color represents the non-mask.

available data (x, y). In particular, our framework with the model parameter θ is defined as:

$$\mathcal{L}_{gen} = -\sum_{j}^{T} \log p_\theta \left(y_j \mid y_{<j}, x, \mathbf{h}\right), \quad (1)$$

where y_j denotes the jth token of the annotated gold answer y. The generator is based on the T5 architecture and uses cross attentions to model the interactions between retrieved passages (Izacard and Grave, 2021). This probability is normalized over T5 vocabulary.

2.3 Passage Mask

Since the over-parameterized neural networks are prone to overfitting, regularization methods such as mask and dropout (Srivastava et al., 2014; Tompson et al., 2015; DeVries and Taylor, 2017; Fan et al., 2021) are usually adopted during training to reduce the generalization error. Specifically, these methods randomly drop part of units in each neural network layer to avoid co-adapting and overfitting. Intuitively, mask and dropout approximately perform to combine exponentially many different neural network architectures efficiently (Srivastava et al., 2014; Ghiasi et al., 2018).

There are few studies of the mask about the reader model training in the retrieval-reader settings. In a standard training setting, each neuron plays the same role and has the same mask rate. In the reader model, intuitively, top-rank passages often contain the answer and are easy to overfit while the other passages have fewer chances to be fitted. Based on our above observations, we propose the Passage-Mask (PM) to regularize the top-rank passages which have larger probabilities to overfit as demonstrated in Figure 1. Briefly, we propose to drop top-rank passages during training.

Though simple and effective, masking increases the gradient variance during training due to its randomness. To reduce gradient variance and lead to stable training, we propose to downsize and select the candidate set of masking with one-shot bi-level optimization in this work.

2.4 Mask Candidates

Denote P passage each with len tokens as $\mathbf{t} = (\mathbf{t}_0, \cdots, \mathbf{t}_P)$ where $\mathbf{t}_i = (t_{i,0}, t_{i,1}, \cdots, t_{i,len})$. We pass the passages $\mathbf{t}$ through the reader model and get $\mathbf{h} = (\mathbf{h}_0, \mathbf{h}_1, \cdots, \mathbf{h}_P)$ where $\mathbf{h}_i = (h_{i,0}, h_{i,1}, \cdots, h_{i,len})$ is the corresponding final-layer hidden state of a passage. Let $\mathcal{DP}$ be a set of mask choice ($e.g.$, retrieval passages) with N candidates and each is denoted as o. For a typically selected mask candidate, we define the mask index set $\{i | i \leq P, i \in \mathbb{N}^+\}$ where P is the number of passages and mask all the corresponding $\mathbf{h}_i$.

To relieve the noisy gradient (large gradient variance), we reduce the size of candidate set. Numerous works ($e.g.$, Ge et al., 2015; Jin et al., 2017; Daneshmand et al., 2018; Chen et al., 2020) have shown that the strong noisy gradient in the backward pass caused by the dropout mask is detrimental to the model optimization. The gradient noise is highly related to the number of drop candidates. As only the top-rank passages play a huge impact during the reader model training, we reduce the size of mask candidates with preferences to mask top-rank passages.

2.5 Fast Search for Mask Candidate Set

To decide the final candidate subset, instead of manual search or grid search (Bergstra and Ben-

gio, 2012; Li and Talwalkar, 2020) all the possible candidates, we propose to do a one-shot fast search of mask candidate *with an almost negligible additional computation cost* compared to standard training schedule. First, we define the search space.

Discrete Search Space. To automatically choose candidates, we consider a set $\mathcal{DP}$ with N candidates and target at selecting S candidates for our Passage-Mask ($S < N$). Inspired by Zoph et al. (2018); Liu et al. (2018); Hong et al. (2022), we create S vectors, and each is a N-dimension vector representing the selected probability for all the N candidates. We denote the hyper-parameter as $w \in \mathbb{R}^{S \times N}$, and each mask candidate as a function $o(\mathbf{h})$ where $\mathbf{h}$ is hidden representations for P.

To make the search space continuous, during training, we relax the categorical choice of a particular operation to a SoftMax over all possible operations, and the output is defined as,

$$\bar{\mathbf{h}}^s = \sum_{o \in \mathcal{DP}} \frac{\exp\left(w_o^s\right)}{\sum_{o' \in \mathcal{DP}} \exp\left(w_{o'}^s\right)} o(\mathbf{h}), \quad (2)$$

where $s \sim \{1, \cdots, S\}$ is sampled with equal probability. Then we pass the $\bar{\mathbf{h}}^s$ to the answer generator. During inference or evaluation, a discrete architecture can be obtained by replacing each mixed operation $\hat{h}^s$ with the most likely operation. *i.e.*, $h^s = o^*(h), o^* = \text{argmax}_{o \in DP}\, w_o^s$.

Bi-level Optimization. To avoid grid-search over the mask schedule, we target at jointly learning the model parameter θ and the mask hyper-parameter w. Formally, the training and the validation sets are denoted by $\mathcal{D}_{tr}$ and $\mathcal{D}_{val}$, respectively. The goal for this optimization is to find θ^* that minimizes the train loss $\theta^* = \arg\min_\theta \mathcal{L}_{train}(\theta, w^*)$, where w^* is obtained by minimizing the validation loss $w^* = \arg\min_w \mathcal{L}_{val}(\theta, w)$. For simplicity, we write $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ as f and g, respectively.

This implies a bi-level optimization problem (Franceschi et al., 2017; Shaban et al., 2019; Grazzi et al., 2020, 2021) which has been shown the effectiveness in the many machine learning fields such as hyperpameter optimization and meta-learning (*e.g.* Yang et al., 2021; Guo et al., 2021; Khanduri et al., 2021). We optimize

$$\begin{aligned} &\min_{\theta \in \mathbb{R}^{d_{model}}} \ell(\theta) = f\left(\theta, w^*\right) := \mathbb{E}_\theta\left[f\left(\theta, w^*\right)\right] \\ &\text{s.t. } w^* = \arg\min_{w \in \mathbb{R}^{d_{mask}}} \{g(\theta, w) := \mathbb{E}_w[g(\theta, w)]\}, \end{aligned} \quad (3)$$

Algorithm 1: Passage Mask (PM)

1: **Input:** Passage P, query x. Model parameter θ with learning rate α_t, mask parameter w with learning rate β_t, update frequency u and time step t.
2: **for** $t = 0$ to final step **do**
3: $\quad \theta \longleftarrow \theta - \alpha_t \nabla_\ell(\theta)$,
4: $\quad$ **if** $t\%u == u - 1$ **then**
5: $\quad\quad w \longleftarrow w - \beta_t \text{grad}_t^g$ where grad_t^g is calculated by Eqn (4).
6: $\quad$ **end if**
7: **end for**

where f, g: $\mathbb{R}^{d_{model}} \times \mathbb{R}^{d_{mask}} \to \mathbb{R}$ with $\theta \in \mathbb{R}^{d_{model}}$ and $w \in \mathbb{R}^{d_{mask}}$; In practice, we do stochastic sample to estimate the expectation value $\mathbb{E}(\cdot)$. Note here that f depends on the minimizer of the mask hyper-parameter objective g, and we refer to $\ell(\theta)$ as the training objective function.

We adopt the recursive momentum techniques developed in (Cutkosky and Orabona, 2019; Tran-Dinh et al., 2019) which yield for-free one-shot training. In summary, our updated mask schedule can be summarized as the below. Define $\eta_t^g \in [0, 1]$, for the problem involving x, we utilize the following momentum-assisted gradient estimator, $\text{grad}_t^g \in \mathbb{R}^{d_{mask}}$, defined recursively as

$$\begin{aligned} &\text{grad}_t^g = \eta_t^g \nabla g\left(\theta_t, w_t\right) \\ &+ (1 - \eta_t^g)\left(\text{grad}_{t-1}^g + \nabla g\left(\theta_t, w_t\right) - \nabla g\left(\theta_{t-1}, w_t\right)\right). \end{aligned} \quad (4)$$

The gradient estimator grad_t^g are computed from the current and past gradient estimates $\nabla g\left(\theta_t, w_t\right)$ and $\nabla g\left(\theta_{t-1}, w_t\right)$. Recent theoretical works in (Khanduri et al., 2021; Ji et al., 2021; Yang et al., 2021) have provided the convergence analysis for the momentum-based recursive optimizer. Thus, PM takes benefits from the model-independent sample complexity and good convergence.

The Proposed Algorithm. Our passage mask with momentum-based recursive bi-level optimization is shown in Algorithm 1. We iteratively update the model parameter θ and mask parameter w in a single-loop manner. The model parameter θ is updated by standard gradient descent, while w is updated in a momentum recursive technique (Cutkosky and Orabona, 2019) with a given frequency u to save computation. We further show in the experiments that the proposed method can

effectively prevent overfitting, improve the model generalization and introduce little additional time cost.

3 Experimental Settings

Table 2 shows the experimental data configuration.

3.1 Task and Evaluation Metrics

Open Question Answering. We use Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017) to evaluate our method on open QA. Natural questions consists of 79,168 train, 8,757 dev, and 3,610 test question answer pairs. It contains questions corresponding to Google search queries. The open-domain version of this dataset is obtained by discarding answers with more than five tokens. TriviaQA (Joshi et al., 2017) contains questions gathered from trivia websites. The unfiltered version of TriviaQA is used for open-domain question answering. Following the open domain splits from (Lee et al., 2019), it contains 78,785 train, 8,837 dev, and 11,313 test question answer pairs. For both datasets, we use publicly available DPR retrieval results for training and inference data, and do not further fine-tune retrievers. Following prior work (Lee et al., 2019), we use Exact Match (EM) as our primary metric.

Dialogue Conversation. Wizard of Wikipedia (WoW) (Dinan et al., 2018) is a large dataset with conversations directly grounded with knowledge retrieved from Wikipedia. The utterances of the speaker should be based on a specific knowledge sentence from a Wikipedia page. We utilize the officially available KILT DPR (Petroni et al., 2020) to extract top passages and report F1 score for evaluation (Asai et al., 2021). **Pre-process to match our setting:** As PM prevents the model from overfitting the top-rank passages, we preprocess the existing development and test dataset by removing the examples with the answers in the top four passages. Evaluating such a dataset, a model cannot provide the true answers if it is overfitted on top 4 passages. This results in 974 dev and 989 test. We report both the preprocess results (Section 4.3) and the non-preprocess results (Section 5).

Fact Verification. FaVIQ (Park et al., 2021) represents fact verification derived from information seeking questions, where the model is given a natural language claim and predicts support or refute with respect to the English Wikipedia. FaVIQ Ambig (FaVIQ-A) is composed from Natural Questions (Kwiatkowski et al., 2019) and AmbigQA (Min et al., 2020). It is constructed from ambiguous questions and their disambiguation. We use the retrieved passages and baseline code provided by Park et al. (2021). Accuracy is adopted as our evaluation metric.

Task	Dataset	Train	Val	Test
Open QA	Natulral Question Open	79.2K	8.8K	3.6k
	TriviaQA unfiltered	78.8K	8.8K	11.3K
Dialogue	Wizard of Wikipedia	63.7K	3.1K	2.9K
Fact Verification	FaVIQ-Ambig (A)	17.0K	4.3K	4.7K

Table 2: Dataset Configuration. The top block is for the Open QA, the middle block is for the dialogue conversation, and the bottom block is for the fact verification.

3.2 Implementation Details

Due to the computational budget, we use the provided checkpoint for the reader model and continue the finetuning with our method. To have fair comparisons, we also finetune the checkpoint with standard training (Details are included in Section 5). For Open QA, following the setting in Izacard and Grave (2021), we utilize the provided checkpoint for the reader and use the top 100 passages during training and inference. We set the training steps as 30k and take the checkpoint that achieves the highest score on the development set. The batch size and the gradient accumulation step are both set to be 1. The learning rate is set to 5×10^{-5} and the number of warm-up steps is 3k. For dialogue conversation and fact verification, following the setting and the checkpoints in (Asai et al., 2021), we use the top 20 passages during training and inference. We set the gradient accumulation step to be 4, with learning rate 10^{-5} and 1k warm-up steps. The development set is used for bi-level optimization. **Search Space.** In all experiments, we use the top four retrieval passages to compose our candidate search space, $\{(1,2),(1,3),(1,4),(2,3),(2,4),(3,4)\}$, where $(1,3)$ is a candidate which indicates that the hidden representation of the 1st and 3rd passages are masked. More detailed experimental settings are included in Appendix A.

4 Experiments

We evaluate the performance of our mask and learning framework in this section. We bold the best result within each column block. The results of our method are obtained with three independent runs

to determine the variance. See Appendix A for full results with error bars.

Model	NQ		TriviaQA	
	dev	test	dev	test
DPR (Karpukhin et al., 2020)	-	41.5	-	57.9
RAG (Lewis et al., 2020)	-	44.5	-	56.1
ColBERT-QA (Khattab et al., 2021)	-	48.2	-	63.2
REALM (Guu et al., 2020)	-	40.4	-	-
FiD base (Izacard and Grave, 2021)	49.2	50.1	68.7	69.3
Ours base	**49.9**	**51.3**	**69.3**	**69.9**
FiD large (Izacard and Grave, 2021)	52.7	54.4	72.5	72.5
Ours large	**53.3**	**55.3**	**73.1**	**72.9**

Table 3: Comparison to models on Natural Questions and TriviaQA. Exact Match scores are reported for each model. 'FiD base' and ' FiD large' represent the base and large generator model (T5) sizes. RAG at here is with BART large.

4.1 Open-Domain QA Results

We first report the results in Table 1. We use the FiD (Izacard and Grave, 2021) base reader model on Natural Questions Open (Kwiatkowski et al., 2019). To verify that the model overfits the top-rank passages, we purposely mask top retrieval passage representations based on the mask position. We observe huge performance degradation ($e.g.$, 50.1 to 44.5) by masking the top one passage representation and even larger performance drop (50.1 to 35.7) by masking the top five retrieval passages.

Table 3 reports our results on two open question answering datasets. ❶ The top block displays the performance of baselines on the NQ and TriviaQA datasets, and the bottom block shows the results of incorporating the PM during the reader model training. We report the results on both base and large settings. With PM, it shows consistent performance gains and better model generalization on both development and test dataset (*e.g.*, 50.1 $\rightarrow$ 51.3 on NQ with FiD base, 54.4 $\rightarrow$ 55.3 on NQ with FiD large). ❷ Through these results, it further confirms that PM can work as an effective module to be incorporated into different-scale models to prevent the overfitting on the top retrieval passages and reason over the entire passages. ❸ PM on improving the reader model can be also seen as a complementary module to works focusing on improving retrieval components (Paranjape et al., 2021; Maillard et al., 2021).

4.2 Fact Verification

We further show the experimental results on FaVIQ-A in Table 4. We adopt several baselines from the existing literature. ① For TF-IDF + BART, following Park et al. (2021), it takes a concatenation of a claim and retrieved passages by TF-IDF from Chen et al. (2017). ② DPR + BART, the baseline, takes a concatenation from passages retrieved by DPR (Karpukhin et al., 2020). ③ For EQA, following Asai et al. (2021), it is built on FiD (Izacard and Grave, 2020) pipeline with T5 base and further incorporates evidentiality of passages into the training of the generator.

In Table 4: ❶ We observe sizable gains over all baselines with a clear margin (from FiD's 64.3, from EQA's 65.7 to ours 66.5), yielding SOTA performance on this dataset. ❷ PM demonstrates the strong capability of avoiding overfitting during the training and allowing the reader model to extract the information from all passages. Thus, it comes to the best performance in most of the settings.

Model	FaVIQ-A	
	dev	test
DPR+BART (Park et al., 2021)	66.9	64.9
TF-IDF + BART (Park et al., 2021)	65.1	63.0
FiD base (Izacard and Grave, 2021)	67.8	64.3
EQA base (Asai et al., 2021)	69.6	65.7
Ours base	**70.6**	**66.5**

Table 4: Performance on FaVIQ-A. We report the accuracy on the development and test dataset. Previous best model is EQA from Asai et al. (2021).

4.3 Dialogue Conversations

Table 5 shows the results on the Wizard of Wikipedia development dataset. We use the FiD (Izacard and Grave, 2021) as our primary baseline, and also include the recent generator model EQA (Asai et al., 2021). Following Asai et al. (2021) and Petroni et al. (2020), we load the official checkpoint from KILT[2] and pre-processed Wikipedia file using the DPR official implementation to retrieve top passages. On Wizard of Wikipedia, by desensitizing the impact from the top-retrieval candidate, our model improves the F1 score from the EQA by 0.7 and the base FiD model by 1.6. Although the input format is conversation and output format is long abstractive sentences, it is interesting to see the consistent improvement of our proposed mask in knowledge-enhanced dialogue. It further demonstrates that PM can be utilized for many ranking-related problems in general NLP tasks.

5 Analysis

What is the influence of the vanilla mask and Dropout? Here we verify whether PM is better

[2] https://github.com/facebookresearch/KILT

Model	F1
FiD base (Izacard and Grave, 2021)	17.1
EQA base (Asai et al., 2021)	18.0
Ours base	**18.7**

Table 5: Results across different strategies on Wizard of Wikipedia. The input format is conversation and output format is abstractive sentences.

than the standard dropout and masking out strategies. With the designed mask candidates, PM targets the top retrieval passages. We compare PM with two standard masking out setting - dimension-wise dropout and vanilla mask. Dimension-wise dropout represents the standard dropout while vanilla mask represents per-passage mask with a scaling factor $1/(1-p)$ where p denotes the mask rate. We set the dropout rate and masking as 0.5 and study whether the standard masking out is applicable to our focused tasks. As shown in Table 6, these two strategies only achieve marginal improvements ($e.g.$ 0.1) while PM yields better results with a clear margin. ***Training Loss Variance.*** To verify the small number of candidates coming to a smaller gradient variance, we investigate the training loss variance for vanilla mask with the different number of candidates. We notice that the vanilla mask with a smaller number of candidate set achieves smaller variance (for *s.t.d.*, 0.042 for six mask candidates vs. 0.046 for sixteen mask candidates). This gets along with our intuition.

Data	FiD base	Dimension Dropout	Vanilla Mask	Ours
NQ	50.1	50.1	50.2	51.3
TriviaQA	69.3	69.4	69.3	69.9

Table 6: Comparison of different masking on Natural Questions and TriviaQA.

More evidence for rank-related overfitting? ❶ We observe huge performance degradation by only masking the top retrieval passage representation during evaluation in Table 1. These results confirm our analysis and motivation for the rank-aware mask. ❷ However, would these results and observations still hold if we try different masking strategies? We use more masking strategies, such as permuting (*i.e.,* random permute the top-K retrieval passages) and removing (*i.e.,* remove the top one retrieval passage and only use the succeed passages), to give more evidences. Similar trend is observed in Table 7.

Efficiency and running time. We provide the parameter sizes, GPU peak memory, and per step time

Position	1st	2nd	3rd	4th	5th	FiD base
Permute Top 3	✓	✓	✓			50.0
Permute Top 5	✓	✓	✓	✓	✓	50.0
Remove 1st	✓					44.9
Remove 2nd		✓				48.7
Remove 3rd			✓			49.3

Table 7: Results of different masking strategies on Natural Questions. FiD (Izacard and Grave, 2021) base model is presented.

comparisons between the baseline and PM. Experiments in this part are performed on a Tesla V100 GPU during training with batch size as 1. ❶ Table 8 shows that PM keeps the parameter size at the same level as the FiD base. The GPU memory and running time of PM are slightly higher (2.7% for memory and 1.6% for running time) than FiD. PM gives the best Exact Match score outperforming FiD, while keeping the comparable efficiency and running time. ❷ Even with the momentum-based recursive optimizer, our passage-aware mask is still computational productive as the bi-level optimization (*e.g.,* applying mask operators and optimizing low-dimension w) has almost zero cost.

Model	EM ↑	Params ↓	GPU memory ↓	s/step ↓
FiD base	50.1	223M	10.9G	12.4
Ours base	51.3	223M	11.2G	12.6

Table 8: Results of parameter size, GPU memory, and step time for FiD base and our base on Natural Question. 's/step' represents step time (second/per step) with batch size as 1.

Ablation studies on the components in PM. We conduct the ablation study to exam the role of bi-level optimization and reduced mask candidate set. For ablation, instead of searching the mask probability for different mask candidates, we randomly sample a candidate in the search space. Through isolating performance of each components, our focus here is to identify the impact of the introduced mask parameter w and the reduced mask set. ❶ Table 9 shows that each component of our method brings benefits. ❷ We find that even without w, '-w' still shows a superior performance to the FiD across both base and large models, indicating that it is often beneficial to have the reduced mask candidate set and target the potential overfitting candidates. ❸ Optimizing w further increases the performance from 50.8 to 51.3 and from 55.0 to 55.3 for FiD base and Large, respectively. It demonstrates the necessity and effectiveness of the fast search for mask candidate set in PM structure.

Data	FiD base	Ours base	-w	FiD large	Ours large	-w
NQ	50.1	51.3	50.8	54.4	55.3	55.0

Table 9: Ablation study of the components in PM. '-w' refers to the removal of the mask parameter w and use a randomly-sampled set of candidates.

WoW additional results. We show the non-preprocessed development set results on the Wizard of Wikipedia in Table 5. We include the RAG (Lewis et al., 2020), DPR + BART (Petroni et al., 2020; Park et al., 2021), and EQA (Asai et al., 2021) as baselines. Even without removing the examples which has the answers in the top 4 passages, PM consistently yields better results than all the baselines. These results verify our conjecture in Section 4.3 that PM not only improves the model generalization for specific cases but also can serve as a plug-in module for general settings since it never hurts the performance in our case.

Model	F1
DPR+BART (Petroni et al., 2020)	15.5
RAG (Lewis et al., 2020)	13.8
FiD base (Asai et al., 2021)	16.9
EQA base (Asai et al., 2021)	17.6
Ours base	**18.4**

Table 10: Results on Wizard of Wikipedia development set for non-preprocessed dataset.

Would we see improvements if finetuning the given checkpoint with baselines? As discussed in Section 3.2, due to computation cost limitation, we use the provided checkpoint for the reader model and continue the finetuning with our method. However, if we continue finetuning the baseline checkpoint, would we still see the improvements? We conduct the experiments on open QA, dialogue and fact verification tasks. We adopt the best baseline models for each task such as FiD base for NQ and TriviaQA, and EQA base for dialogue conversations and fact verification. In Table 11, ours indicates strong improvements. This further proves that our selection method is capable of reasoning over the retrieval passages. By only finetuning the baselines, it keeps similar performance such as the baseline on WoW and FaVIQ-A.

6 Related Work

Retrieval Read Architecture Recent retriever models (*e.g.*, Lee et al., 2019; Karpukhin et al., 2020; Khattab et al., 2021) learn to encode the input query and large-scale passage collection to score their similarities. Readers (generators) aim

Model	NQ	TriviaQA	WoW	FaVIQ-A
Baseline	50.1	69.3	17.6	65.7
Baseline finetuning	50.2	69.4	17.5	65.5
Ours base	**51.3**	**69.8**	**18.4**	**66.5**

Table 11: Finetuning results on Natural Questions test dataset, TriviaQA test dataset, FaVIQ-A test dataset and Wow non-preprocessed development dataset. We report results of our mask strategy with baseline and baseline finetuning.

to generate answers condition on the question and the retrieved passages (Yang et al., 2019; Lewis et al., 2020; Mao et al., 2020). Our work relies on this architecture and further fine-grain the reader model to introduce the passage-aware masking and promote the reasoning over the entire passage set.

Rank-Related Studies Passage ranking has shown promising performance improvements. The most popular approach is combining the passage score and answer score together (Karpukhin et al., 2020; Xiong et al., 2020; Qu et al., 2020). Other works (*e.g.*, Nogueira et al., 2020; Fajcik et al., 2021; Zhang et al., 2021b) propose additional modules or operations to re-identify the passage rank. Nogueira et al. (2020) uses seq2seq model to identify the document's relevance to the query, Fajcik et al. (2021) introduces a passage re-ranking module, and Zhang et al. (2021b) proposes to use the calibrator as an answer reranker. There are some works that focus on the ranking efficiency. Luan et al. (2021) creates a simple neural model that combines the efficiency of dual encoders. Similarly, we also find out that directly taking the rank makes the model overfitting. Different from existing works, PM rethinks the impact of retrieval passage ranking from the regularization and generalization perspective. We focus on preventing the overfitting and improving the reasoning generalization during training. In the meantime, PM is also compatible with other previous ranking works with the potential to jointly improve the performance.

7 Conclusion

Our work demonstrates the benefits of introducing a passage mask mechanism. The proposed mask can desensitize the impact from the top-rank retrieval passages and prevent the model from overfitting. The proposed strategy shows noticeable gains in performance across open question answering, dialogue conversation, and fact verification. We further conduct the detailed study with the proposed masking strategy in different settings, *e.g.*,

comparing with vanilla masking, providing more evidence for rank-related overfitting, and verifying the impact of different components. To summarize, the proposed PM is effective and general, with the potential to be incorporated into existing models for various NLP tasks.

8 Limitations

In real practices or real-life scenarios, the data is often biased. The gap between the training and testing data might be large and unexpected. Thus, incautious implementation or vague understanding of model output might lead to unanticipated false consequences. In addition, with computational consumption, environmentally sustainability and users friendly should be considered.

Acknowledgments

The authors thank Eunsol Choi and Anqi Lou for helpful comments on the paper draft.

References

Akari Asai, Matt Gardner, and Hannaneh Hajishirzi. 2021. Evidentiality-guided generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2112.08688*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2).

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.

Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. 2020. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*.

Ashok Cutkosky and Francesco Orabona. 2019. Momentum-based variance reduction in non-convex sgd. *Advances in neural information processing systems*, 32.

Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. 2018. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pages 1155–1164. PMLR.

Terrance DeVries and Graham W Taylor. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2018. Wizard of wikipedia: Knowledge-powered conversational agents. *arXiv preprint arXiv:1811.01241*.

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-d2: A modular baseline for open-domain question answering. *arXiv preprint arXiv:2109.03502*.

Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. 2020. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 33:16362–16376.

Xinjie Fan, Shujian Zhang, Korawat Tanwisuth, Xiaoning Qian, and Mingyuan Zhou. 2021. Contextual dropout: An efficient sample-dependent dropout module. *arXiv preprint arXiv:2103.04181*.

Luca Franceschi, Michele Donini, Paolo Frasconi, and Massimiliano Pontil. 2017. Forward and reverse gradient-based hyperparameter optimization. In *International Conference on Machine Learning*, pages 1165–1173. PMLR.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. 2015. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. 2018. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31.

Riccardo Grazzi, Luca Franceschi, Massimiliano Pontil, and Saverio Salzo. 2020. On the iteration complexity of hypergradient computation. In *International Conference on Machine Learning*, pages 3748–3758. PMLR.

Riccardo Grazzi, Massimiliano Pontil, and Saverio Salzo. 2021. Convergence properties of stochastic hypergradients. In *International Conference on Artificial Intelligence and Statistics*, pages 3826–3834. PMLR.

Zhishuai Guo, Yi Xu, Wotao Yin, Rong Jin, and Tianbao Yang. 2021. On stochastic moving-average estimators for non-convex optimization. *arXiv preprint arXiv:2104.14840*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Weijun Hong, Guilin Li, Weinan Zhang, Ruiming Tang, Yunhe Wang, Zhenguo Li, and Yong Yu. 2022. Dropnas: Grouped operation dropout for differentiable architecture search. *arXiv preprint arXiv:2201.11679*.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Gautier Izacard and Edouard Grave. 2021. Distilling knowledge from reader to retriever for question answering. In *ICLR 2021, 9th International Conference on Learning Representations*.

Kaiyi Ji, Junjie Yang, and Yingbin Liang. 2021. Bilevel optimization: Convergence analysis and enhanced design. In *International Conference on Machine Learning*, pages 4882–4892. PMLR.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. 2017. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR.

Mandar Joshi, Eunsol Choi, Daniel S 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.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *Empirical Methods in Natural Language Processing (EMNLP)*.

Prashant Khanduri, Siliang Zeng, Mingyi Hong, Hoi-To Wai, Zhaoran Wang, and Zhuoran Yang. 2021. A near-optimal algorithm for stochastic bilevel optimization via double-momentum. *Advances in Neural Information Processing Systems*, 34.

Omar Khattab, Christopher Potts, and Matei Zaharia. 2021. Relevance-guided supervision for openqa with colbert. *Transactions of the Association for Computational Linguistics*, 9:929–944.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural Questions: a benchmark for question answering research. *TACL*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *International Conference on Machine Learning (ICML)*, abs/1906.00300.

Kyungjae Lee, Seung-won Hwang, Sang-eun Han, and Dohyeon Lee. 2021. Robustifying multi-hop qa through pseudo-evidentiality training. *arXiv preprint arXiv:2107.03242*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, F. Petroni, V. Karpukhin, Naman Goyal, Heinrich Kuttler, M. Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

Liam Li and Ameet Talwalkar. 2020. Random search and reproducibility for neural architecture search. In *Uncertainty in artificial intelligence*, pages 367–377. PMLR.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2021. Entity-based knowledge conflicts in question answering. *arXiv preprint arXiv:2109.05052*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. 2021. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oğuz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. *arXiv preprint arXiv:2101.00117*.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, abs/2004.10645.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. *arXiv preprint arXiv:2003.06713*.

Ashwin Paranjape, Omar Khattab, Christopher Potts, Matei Zaharia, and Christopher D Manning. 2021. Hindsight: Posterior-guided training of retrievers for improved open-ended generation. *arXiv preprint arXiv:2110.07752*.

Jungsoo Park, Sewon Min, Jaewoo Kang, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Faviq: Fact verification from information-seeking questions. *arXiv preprint arXiv:2107.02153*.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. 2020. Kilt: a benchmark for knowledge intensive language tasks. *arXiv preprint arXiv:2009.02252*.

Chen Qu, Liu Yang, Cen Chen, Minghui Qiu, W Bruce Croft, and Mohit Iyyer. 2020. Open-retrieval conversational question answering. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 539–548.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, M. Matena, Yanqi Zhou, W. 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.

Stephen Robertson and Hugo Zaragoza. 2009. *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. 2019. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1723–1732. PMLR.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Kai Sun, Seungwhan Moon, Paul Crook, Stephen Roller, Becka Silvert, Bing Liu, Zhiguang Wang, Honglei Liu, Eunjoon Cho, and Claire Cardie. 2020. Adding chit-chat to enhance task-oriented dialogues. *arXiv preprint arXiv:2010.12757*.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient object localization using convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 648–656.

Quoc Tran-Dinh, Nhan H Pham, Dzung T Phan, and Lam M Nguyen. 2019. Hybrid stochastic gradient descent algorithms for stochastic nonconvex optimization. *arXiv preprint arXiv:1905.05920*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. 2020. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*.

Peng Xu, Davis Liang, Zhiheng Huang, and Bing Xiang. 2021. Attention-guided generative models for extractive question answering. *arXiv preprint arXiv:2110.06393*.

Junjie Yang, Kaiyi Ji, and Yingbin Liang. 2021. Provably faster algorithms for bilevel optimization. *Advances in Neural Information Processing Systems*, 34.

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. *arXiv preprint arXiv:1902.01718*.

Shujian Zhang, Xinjie Fan, Bo Chen, and Mingyuan Zhou. 2021a. Bayesian attention belief networks. In *International Conference on Machine Learning*, pages 12413–12426. PMLR.

Shujian Zhang, Chengyue Gong, and Eunsol Choi. 2021b. Knowing more about questions can help: Improving calibration in question answering. *arXiv preprint arXiv:2106.01494*.

Shujian Zhang, Chengyue Gong, Xingchao Liu, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. 2022. Allsh: Active learning guided by local sensitivity and hardness. *arXiv preprint arXiv:2205.04980*.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.

A Experimental details

A.1 Full Results With Error Bar

We report the full results of our method with the error bar for open question answering and dialogue conversations in Table 12 and 13, respectively. The full result of fact verification is demonstrated in Table 14.

Model	NQ		TriviaQA	
	dev	test	dev	test
DPR (Karpukhin et al., 2020)	-	41.5	-	57.9
RAG (Lewis et al., 2020)	-	44.5	-	56.1
ColBERT-QA (Khattab et al., 2021)	-	48.2	-	63.2
REALM (Guu et al., 2020)	-	40.4	-	-
FiD base (Izacard and Grave, 2021)	49.2	50.1	68.7	69.3
Ours base	49.9±0.3	51.3±0.2	69.3±0.2	69.9±0.2
FiD large (Izacard and Grave, 2021)	52.7	54.4	72.5	72.5
Ours large	53.1±0.1	55.3±0.2	72.9±0.1	72.9±0.2

Table 12: Full results on Natural Questions and TriviaQA. Exact Match scores are reported for each model. 'FiD base' and ' FiD large' represents the base and large generator model (T5) sizes. RAG at here is with BART large.

Model	F1
FiD base (Izacard and Grave, 2021)	17.1
EQA base (Asai et al., 2021)	18.0
Ours base	18.7±0.2

Table 13: Full results across different strategies on dialogue conversations (Wizard of Wikipedia). The input format is conversation and the output format is abstractive sentences.

Model	FaVIQ-A	
	dev	test
DPR+BART (Park et al., 2021)	66.9	64.9
TF-IDF + BART (Park et al., 2021)	65.1	63.0
FiD base (Izacard and Grave, 2021)	67.8	64.3
EQA base (Asai et al., 2021)	69.6	65.7
Ours base	70.6±0.2	66.5±0.2

Table 14: Full performance on FaVIQ-A. We report the accuracy on the development and test dataset.

A.2 Experimental Datasets

Open Question Answering. Following the setting in Lee et al. (2019) and Karpukhin et al. (2020) for Natural Questions and TriviaQA, the original development set is used as the test set, and 10% of the training set is used as the development set. All questions with answers longer than five tokens are discarded for the Natural Questions. We use the Wikipedia dumps from Dec. 20, 2018 for NQ and TriviaQA and apply the same preprocessing as Chen et al. (2017).

Fact Verification. FAVIQ (Park et al., 2021) represents fact verification derived from information seeking questions, where the model is given a natural language claim and predicts support or refute with respect to the English Wikipedia. It consists of 188k claims derived from an existing corpus of ambiguous information-seeking questions. FaVIQ Ambig (FaVIQ-A) is composed from Natural Questions (Kwiatkowski et al., 2019) and AmbigQA (Min et al., 2020). AmbigQA provides disambiguated question-answer pairs for NQ questions, thereby highlighting the inherent ambiguity in information-seeking questions. FaVIQ-A uses the disambiguated question-answer pairs and generates support and refute claims from matching pairs (filmed–2000, released–2001) and crossover pairs (filmed–2001, released–2000), respectively (Park et al., 2021).

Dialogue Conversation. With the goal of making virtual assistant conversations more engaging and interactive, Sun et al. (2020) develops an engaging chatbot that can discuss a variety of topics with a user. The conversation history and the next utterance are used as input and output, respectively (Petroni et al., 2020). Wizard of Wikipedia (WoW) (Dinan et al., 2018) is a large dataset of conversation grounded with knowledge retrieved from Wikipedia. In the conversation, the utterances from the speaker should be relied on a specific knowledge sentence from a Wikipedia page.

A.3 Experimental Settings

For Open QA, we follow the setting in (Izacard and Grave, 2020, 2021) and initialize our models with the pretrained T5 model (Raffel et al., 2020) from the HuggingFace Transformer library[3](Fan et al., 2020; Zhang et al., 2021a). Two model sizes, base (220M parameters) and large (770M parameters), are considered. We finetune the models on each dataset independently and use provided checkpoints from (Izacard and Grave, 2021)[4]. Following Izacard and Grave (2021), we adopt the AdamW (Loshchilov and Hutter, 2017; Zhang et al., 2022) with the learning rate 5×10^{-5} and weight decay 0.25. The training step is 30k. The batch size and gradient accumulation step are both set to 1. The development dataset is used for bi-level optimiza-

[3] https://github.com/huggingface/transformers

[4] https://github.com/facebookresearch/FiD

tion and the warm-up steps is 3000. We evaluate models every 500 steps and select the best one on the validation set based on the Exact Match score. For Natural Question, we sample the target among the list of answers during the training. For TriviaQA, we use the unique human-generated answer. For both training and testing, we retrieve 100 passages and truncate them to 250 word pieces. The retrieval passages are from DPR (Karpukhin et al., 2020) for NQ and TriviaQA.

For fact verification and dialogue conversation, following Petroni et al. (2020) and Asai et al. (2021), we use the top 20 passages during training and inference. The batch size is set to 1. We set the gradient accumulation step to be 4 to keep the same batch size as previous works. The AdamW (Loshchilov and Hutter, 2017) with the learning rate 1×10^{-5} and weight decay 0.25 are utilized. The training steps are 30k and warm-up steps are 1k. Following (Asai et al., 2021)[5], for fact verification, we report the accuracy as evaluation metric and report the results on FaVIQ-A test set in Table 4. For dialogue, we evaluate model based on the F1 score and report the results on WoW development set in Table 5.

[5] https://github.com/AkariAsai/evidentiality_qa