FastFiD: Improve Inference Efficiency of Open Domain Question Answering via Sentence Selection

Anonymous ACL submission

Abstract

Open Domain Question Answering (ODQA) has been advancing rapidly in recent times, driven by significant developments in dense passage retrieval and pretrained language models. State-of-the-art models typically incorporate the FiD framework, which is composed by a neural retriever alongside an encoder-decoder neural reader. In the answer generation process, the retriever will retrieve numerous passages (around 100 for instance), each of which is then individually encoded by the encoder. Subsequently, the decoder makes predictions based on these encoded passages. Nevertheless, this framework can be relatively time-consuming, particularly due to the extensive length of the gathered passages. To address this, we introduce FastFiD in this paper, a novel approach that executes sentence selection on the encoded passages. This aids in retaining valuable sentences while reducing the context length required for generating answers. Experiments on three commonly used datasets (Natural Questions, TriviaQA and ASQA) demonstrate that our method can enhance the inference speed by **2.3X-5.7X**, while simultaneously maintaining the model's performance. Moreover, an in-depth analysis of the model's attention reveals that the selected sentences indeed hold a substantial contribution towards the final answer.

1 Introduction

Open Domain Question Answering(ODQA) is a longstanding task in Natural Language Processing that involves generating an answer solely based on a given question. Recent advancements in this field have typically adopted the Retriever-Reader framework (Chen et al., 2017; Karpukhin et al., 2020; Lewis et al., 2020; Izacard and Grave, 2021b), which breaks down the task into two distinct stages. Initially, a retriever retrieves a set of relevant passages from a high-quality collection of open domain documents, such as Wikipedia. Subsequently, a reader model generates an answer by considering the question and the retrieved passages. Thanks to advancements in neural models, the retriever has transitioned from traditional search methods like TF-IDF (Chen et al., 2017) to dense passage retrieval (Karpukhin et al., 2020), resulting in improved retrieval performance. Furthermore, driven by the progress of Pretrained Language Models (PLMs) (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020), the reader has evolved from extracting answers from a single passage to generating answers from multiple passages (Izacard and Grave, 2021b). This approach enables the model to leverage information from various passages more effectively, thereby producing more accurate answers.

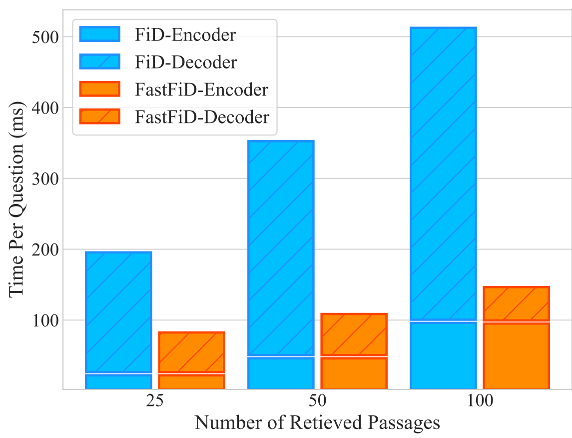

Figure 1: Inference Time for FiD (base) and FastFiD (base) with varying numbers of retrieved passages. As the number of retrieved passages increases, FiD encounters increasingly severe efficiency issues. Our FastFiD significantly accelerates the process by greatly reducing decoding time.

A recently successful model is Fuse-in-Decoder (FiD) (Izacard and Grave, 2021b), which utilizes Dense Passage Retrieval and a generative reader based on T5 (Raffel et al., 2020), an encoder-decoder model. FiD is capable of encoding each retrieved passage independently and subsequently concatenating these encoded passages to form an extensive context. The concatenated context is then used by the decoder to generate a response. Owing to its straightforward and extensible architecture, numerous subsequent works have introduced modifications based on this framework (Sachan et al., 2021b; Yu et al., 2022; Wen et al., 2022). However, as the decoder must generate a response based on all retrieved passages, it can be time-consuming to enhance performance through the retrieval of additional passages. Moreover, in real-world scenarios, the latency in generating an answer is a significant factor. As larger language models continue to be developed and demonstrate superior performance, this issue may become more pronounced.

To address this issue, we introduce FastFiD, a novel approach that performs sentence selection post the encoder's output and maintains only the essential sentences as references for the decoder, thereby significantly reducing the inference time for each query.

To demonstrate the effectiveness of our approach, we first carry out experiments to ascertain that the multi-task training, which involves sentence selection and answer generation, does not conflict with one another during the model's learning process. This is achieved by seamlessly incorporating a selection loss on the encoder outputs with a language modelling loss on answer generation, enabling the model to simultaneously handle both sentence selection and answer generation tasks. An in-depth analysis of the decoder's cross-attention reveals that tokens from the chosen sentences yield a higher average attention score compared to those unchosen. This finding provides compelling evidence that the selected sentences significantly contribute more to the model's predictions. Guided by this insight, we execute a secondary training phase, obliging the model to solely anchor to the selected encoder outputs when making the final prediction.

The experimental results obtained from two widely used ODQA datasets, namely Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017), along with a long-form QA dataset called ASQA (Hofstätter et al., 2023), demonstrate that FastFiD can achieve performance metrics comparable to the original FiD. Notably, it can reduce the context length by up to **38X** and accelerate the inference time by **2.3X-5.7X** on different datasets. To validate the effectiveness of sentence selection, we also compare its performance with passage reranking after the encoder outputs. The results show that sentence selection yields better performance while maintaining a similar context length. This comparison indicates that sentence selection is a more effective strategy for compressing information across multiple passages.

In summary, our contributions can be encapsulated within the following three key points:

- We implement a multi-task training approach, demonstrating that a singular reader model can concurrently perform sentence selection and answer generation.
- We introduce a novel technique to enhance the inference efficiency of FiD while preserving its question-answering capabilities.
- We carry out plenty of experiments to validate and analyze the effectiveness of our method.

2 Related Work

Open Domain Question Answering serves a crucial role in natural language processing, with its primary function being to respond to factoid questions. Followed by Chen et al. (2017), current ODQA systems usually use a large collection of documents like Wikipedia as the knowledge source to answer questions. Since the document collection usually contains millions of documents, the system always adds a retriever to retrieve some most relevant passages for the reader to make predictions. To get better retriever performance, Karpukhin et al. (2020) proposed a shift from sparse retrieval systems like TF-IDF to dense retrieval to enhance the efficiency of the retriever. Subsequent research (Lewis et al., 2020; Sachan et al., 2021b; Jiang et al., 2022; Lee et al., 2022) has investigated the use of end-to-end training methodologies to further boost the performance of the retriever, bypassing the need for pair-wise question-document data. Izacard and Grave (2021a) demonstrated an improvement in performance through the distillation of knowledge from the reader to the retriever. The idea of pretraining both the retriever and the reader on a vast, unlabeled corpus has been explored by Guu et al. (2020) and Sachan et al. (2021a). A different research trajectory has aimed to augment the reader's capacity to better utilize retrieved passages. With the advancement of PLMs, the reader has evolved from RNN-based

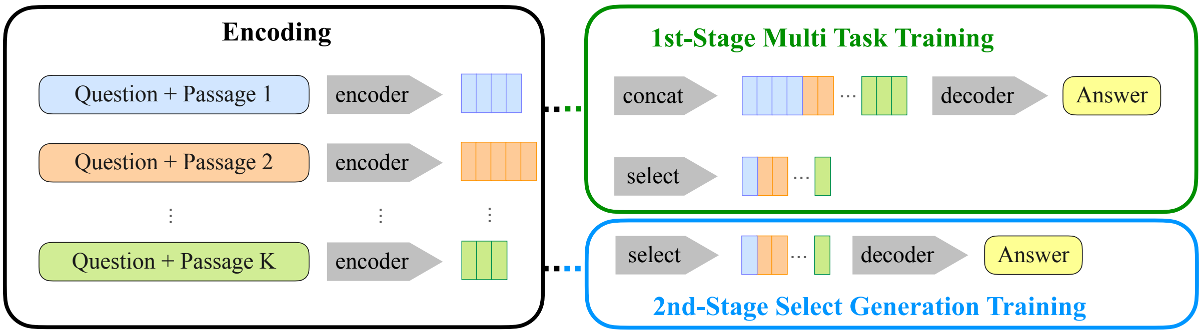

Figure 2: An overview of our FastFiD training pipeline. The pipeline undergoes two stages of training to empower the model with the capacity to generate answers based on the selected sentences, thereby minimizing inference time.

models (Chen et al., 2017) to BERT-based extractive readers (Karpukhin et al., 2020) and T5 or BART-based generative readers (Lewis et al., 2020; Izacard and Grave, 2021b). Recent studies (Cheng et al., 2021; Fajcik et al., 2021; Wen et al., 2022) have pivoted towards a hybrid approach, exploring the integration of both generative and extractive readers to further enhance system performance.

Efficient ODQA The majority of contemporary Open-Domain Question Answering (ODQA) systems face efficiency challenges, primarily due to the large-scale document processing and the use of sizable pre-trained language models. These efficiency challenges arise in two stages.

The first stage is retrieval efficiency. Given the potentially massive number of passages, dense retrieval can be extremely slow. Instead of relying solely on brute force search methods, alternative algorithms such as Approximate Nearest Neighbor (ANN) (Johnson et al., 2021) and Hierarchical Navigable Small World (HNSW) (Malkov and Yashunin, 2020) can be employed to expedite the retrieval process.

The second efficiency challenge lies in the reading process, which involves handling multiple passages for each query. To address this, Hofstätter et al. (2023) propose FiD-Light, which limits the decoder's attention to the first k tokens of each passage to reduce the context length. FiDO (de Jong et al., 2023) explores reducing the number of cross attention layers in FiD's decoder to increase efficiency, but this comes at the cost of re-pretraining the base model. Other complementary strategies explore to identify and stop processing less relevant passages early on by utilizing adaptive computation (Wu et al., 2020, 2021) or knowledge graph with GNN network (Yu et al., 2022). Additionally, some research has focused on directly retrieving answers to questions without the need for passage processing (Seo et al., 2019; Lee et al., 2021; Lewis et al., 2021), or using language models to generate answers directly by finetuning and few-shot prompting (Roberts et al., 2020; Brown et al., 2020).

3 Methods

In this section, we propose FastFiD, which is based on FiD (Izacard and Grave, 2021b) to reduce its inference time and make it more efficient. FastFiD contains a two-stage training procedure. Initially, in the first stage, we introduce a multi-task training objective that allows for simultaneous training of sentence selection and answer generation (Section 3.1). Then, in the second stage, we use the model trained in the first stage as the base model and perform continuous training on generating answers with reference to the selected tokens. (Section 3.2). Finally, in the inference stage, the encoder transcodes each passage into context embeddings and curates a selection of valuable sentences, which are then employed in the decoder generation process to expedite inference time (Section 3.3). The overall framework is shown in Figure 2.

3.1 Multi-Task Training

In this section, we present our multi-task training approach. Following FiD, we utilize T5, an encoder-decoder based PLM, as our base model. Given a question-answer pair (q, a), we initially retrieve K relevant passages $p^1, p^2, ..., p^K$, with their respective titles $t^1, t^2, ..., t^K$ from an extensive knowledge base, predicated on the question q. Subsequently, the question q and each corresponding passage p^k are combined to generate a comprehensive input in the following structure:

$$I^k = \textbf{Question: } q \ \textbf{Title: } t^k \ \textbf{Context: } p^k \quad (1)$$

After this, the model's encoder transcodes each input I^k into context embeddings $h_1^k, h_2^k, ..., h_N^k \in \mathbb{R}^d$, where N represents the max sequence length of the input text. Our multi-task training objective, which encompasses sentence selection and answer generation, is built upon these encoded context embeddings.

3.1.1 Sentence Selection

In the context of a given retrieved passage p^k, there exist M_k key sentences, represented as $\mathcal{S}^k = s_1^k, s_2^k, ..., s_{M_k}^k$, that are crucial for answering the question. As established in prior extractive reader works (Chen et al., 2017; Kwiatkowski et al., 2019; Min et al., 2019; Cheng et al., 2021), we implement a classification head to anticipate the begin and end positions of each key sentence. Taking into account the conclusions of Cheng et al. (2020) and Cheng et al. (2021), we employ a multi-objective approach to enhance sentence selection performance.

In formal terms, the probability of a span (i^k, j^k) being a selected sentence can be broken down into the product of the probabilities of the i^k-th token being the start token and the j^k-th token being the end token. We integrate some learned parameters, namely w_b, w_e, b_b, b_e, to calculate the start and end score:

$$S_b(i^k) = w_b^T h_i^k + b_b; \quad S_e(j^k) = w_e^T h_j^k + b_e \tag{2}$$

By calculating the probability based on different normalizing factors, we can derive the local passage-level probability and the global multi-passage-level probability. With local probability, the probability of each token in different retrieved passages will not affect one another. By normalizing the start and end probabilities by the total scores of all tokens in input I^k, we derive the probability as follows:

$$P_b^L(i^k) = \frac{\exp(S_b(i^k))}{\sum_n \exp(S_b(n^k))}; \quad P_e^L(j^k) = \frac{\exp(S_e(j^k))}{\sum_n \exp(S_e(n^k))} \tag{3}$$

In the case of global probability, we calculate the probability taking into account all the tokens in the top-K passages from the retriever. Therefore, the probability of each token being the start or end of the selected sentence will be jointly optimized across different passages:

$$P_b^G(i^k) = \frac{\exp(S_b(i^k))}{\sum_k \sum_n \exp(S_b(n^k))}; \quad P_e^G(j^k) = \frac{\exp(S_e(j^k))}{\sum_k \sum_n \exp(S_e(n^k))} \tag{4}$$

We then obtain the local and global probabilities of a span being the supported sentence as follows:

$$P_s^{\{L,G\}}(i^k, j^k) = P_b^{\{L,G\}}(i^k) \times P_e^{\{L,G\}}(j^k) \tag{5}$$

Following the methodology of Cheng et al. (2021), we utilize a multi-objective formulation to merge the HardEM (Min et al., 2019) and MML (Karpukhin et al., 2020) objectives for more efficient training. In the multi-objective formulation, we calculate the HardEM loss on global probability and the MML loss on local probability. The final sentence selection loss is calculated as follows:

$$\mathcal{L}_S = -\log \max_{(i,j)\in\mathcal{S}} P_s^G(i,j) - \frac{1}{K}\sum_k^K \log \sum_{(i^k,j^k)\in\mathcal{S}^k} P_s^L(i^k, j^k) \tag{6}$$

where $\mathcal{S} = \mathcal{S}^1 \cup \mathcal{S}^2 \cup ... \cup \mathcal{S}^K$ is the set of all crucial sentences in the top-K retrieved passages. Since ODQA datasets usually only contain question-answer pairs without annotated valuable sentences, we consider the sentences that include the short span answer in each retrieved passage as the crucial sentences.

3.1.2 Answer Generation

As the pipeline in FiD, we employ the decoder to fuse the information of retrieved passages and make a prediction. More specifically, we first concatenate the context embeddings of all inputs:

$$H = (H^1; H^2; ...; H^K) \tag{7}$$

where H^k represents the context embeddings for input I^k, therefore H have an overall length of $N \times K$. Subsequently, the decoder conducts cross-attention over the concatenated context embeddings to make generation.

For the training objective, it optimizes the language modelling loss of generating the golden answer a, a sequence of tokens represented as $\{a_1, a_2, ..., a_{N_a}\}$:

$$\mathcal{L}_G = -\log \sum_i^{N_a} P_{\theta_d}(a_i | H, a_{1:i-1}) \tag{8}$$

where θ_d is parameters of the decoder.

Finally, in the first-stage multi-task training, we integrate the sentence selection objective and answer generation objective in the following manner to simultaneously equip the model with these two capabilities. The variable λ is a hyper-parameter that balances these two objectives:

$$\mathcal{L}^1 = \mathcal{L}_G + \lambda \mathcal{L}_S \tag{9}$$

3.2 Select Generation Training

After completing the initial stage of training as outlined in Section 3.1, our preliminary experiments reveal that while the model possesses the capacity to select valuable sentences and make predictions at the same time, directly requiring the decoder to form predictions solely based on these selected sentences significantly hampers the performance of the model. We hypothesise that this is because of the gap in context length for decoder between training and inference. Therefore, we introduce a second stage of continuous training aimed at minimizing this discrepancy linked with context length.

More specifically, we initially obtain the context embeddings of the selected sentences, and this is done by the global multi-passage-level selection probability.

$$\begin{aligned} H_s &= \bigcup h_{i^k:j^k}; \\ (i^k, j^k) &\in \mathrm{TopK}(P_s^G(i,j)) \end{aligned} \tag{10}$$

The resultant loss for answer generation can then be expressed as follows:

$$\mathcal{L}_G^s = -\log \sum_i^{N_a} P_{\theta_d}(a_i | H_s, a_{1:i-1}) \tag{11}$$

Throughout the second stage of training, we maintain the use of a multi-task training objective to keep both the sentence selection ability and answer generation ability, thereby facilitating better performance.

$$\mathcal{L}^2 = \mathcal{L}_G^s + \lambda \mathcal{L}_S \tag{12}$$

3.3 Select Generation Inference

Following the two-stage training process, we acquire a model that is capable of dynamically selecting valuable sentences for the decoder to make generation. The inference process closely mirrors the second stage of training described in Section 3.2. Initially, valuable context embeddings are selected based on global selection probability. Subsequently, a greedy decoding strategy is employed to generate the answer based on the selected context embeddings denoted as H_s.

	#Train	#Dev	#Test	#Sent.
NQ	79,168	8,757	3,610	14.84
TriviaQA	76,423	8,837	11,313	30.58
ASQA	4,353	968	1,015	22.32

Table 1: Statistics of two ODQA datasets. #Train/#Dev/#Test imply the number of train/dev/test samples. #Sent. means the average number of valuable sentences recognized in top-100 retrived passages.

4 Experiments

4.1 Experimental Setup

Same as FiD (Izacard and Grave, 2021b), we utilize T5 (Raffel et al., 2020) as our base model. For passage retrieval, we utilize the retriever demonstrated by Izacard and Grave (2021a) which has superior retrieval performance. Following previous work (Lee et al., 2019; Karpukhin et al., 2020), we use the preprocessed English Wikipedia Snapshot on 12-30-2018 as our knowledge source. And we use average time per question (TPQ) to measure model's inference efficiency. We conduct experiments on two commonly used ODQA datasets and one long-form QA dataset. Their statistics are shown in Table 1. We use the original train/dev/test split to conduct our experiments.

Natural Questions (Kwiatkowski et al., 2019) is a large ODQA dataset where all questions are mined from Google Search real queries. The annotated answers are all created by human annotators based on Wikipedia documents. Lee et al. (2019) further filter out questions with short answers to construct the open domain version of NQ, which we used in our experiment. We evaluate the performance of our model on NQ using the Exact Match (EM) metric.

TriviaQA (Joshi et al., 2017) is collected from 14 trivia and quiz-league websites with human-annotated answers and a set of answer aliases gathered from Wikipedia. We use the unfiltered question-answer pairs and discard the distantly supervised documents as our open domain version. Similar to NQ, we assess our model's performance on TriviaQA using the Exact Match (EM) metric.

ASQA (Stelmakh et al., 2023) is a long-form question answering dataset that builds upon the AmbigQA (Min et al., 2020) dataset. It consists of ambiguous questions with multiple short span answers and long-form answers from human annotators that coverage all possible short span answers. In line with Stelmakh et al. (2023), we evaluate the performance of our model on this dataset using the STR-EM (String Exact Match) metric. STR-EM measures the proportion of disambiguated short answers that are correctly identified within the long answer. Since the test set of ASQA is not publicly available, our evaluation is conducted solely on the development set of ASQA.

Baselines We mainly compare our method with vanilla FiD, aiming at enhancing its inference efficiency. Additionally, we contrast our approach with the model resulting from our first training stage, referred to as HybridFiD, a model that is capable of simultaneously performing answer generation and sentence selection. Besides, we also compare with FiD-Light (Hofstätter et al., 2023), which propose to select the first-k tokens from each passage as the context for decoder and improve efficiency.

Implementation Our method is implemented using PyTorch (Paszke et al., 2019) and Huggingface Transformers (Wolf et al., 2020), with training efficiency enhanced by DeepSpeed ZeRO-2 (Rajbhandari et al., 2020). Due to GPU limitations, we conduct experiments using T5-Base, which has 345M parameters. We employ the AdamW (Loshchilov and Hutter, 2019) optimizer for stable training. More implementation details are shown in Appendix A.

4.2 Main Results

Answer Generation The performance and inference speed of our FastFiD and other baselines are presented in Table 2. Unlike FiD-Light, which sacrifices QA performance to accelerate the inference process, FastFiD achieves substantial acceleration while maintaining similar or even superior QA performance compared to vanilla FiD. Additionally, FastFiD demonstrates significantly greater inference speedup than FiD-Light on NQ and ASQA, and comparable acceleration on TriviaQA. This can be attributed to our context-aware compression methods, which extract more essential information with fewer tokens compared to the static method employed in FiD-Light. Among the three datasets, FastFiD achieves the highest acceleration

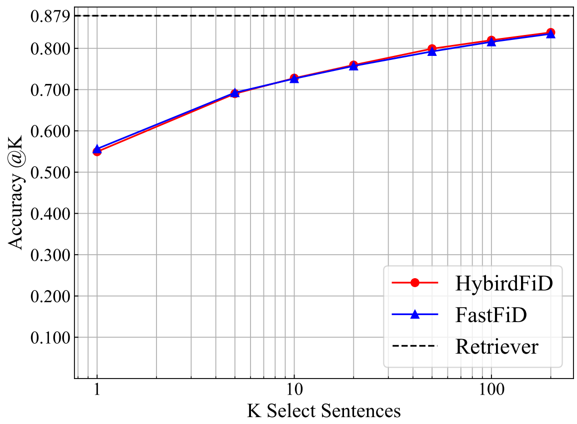

Figure 3: Sentence selection performance on NQ-Dev for HybirdFiD and FastFiD with 100 retrieved passages. Retriever means the accuracy of our retriever when retrieving 100 passages, which can be seen as an upper bound.

on ASQA due to the longer answer format. This showcases the effectiveness of FastFiD in long-form QA, which is a widely utilized task by modern LLM system like New Bing[1] and ChatGPT[2].

We also conducted experiments with varying numbers of retrieved passages on NQ, and the results are presented in Table 3. As observed, regardless of the number of retrieved passages, our FastFiD consistently matches or even surpasses FiD and HybridFiD in terms of EM, while significantly reducing the context length and inference time. Moreover, as the number of retrieved passages increases, the speedup rate also expands. This evidence underscores the potential of our method for effective implementation with a larger number of passages or lengthy documents.

Sentence Selection Similar to the metrics employed in the retriever, we measure the performance of sentence selection utilizing the accuracy@k, which assesses whether the correct answer appears within the top-k sentences. As depicted in Figure 3, there is a positive correlation between the increase in selected sentence numbers and accuracy, eventually surpassing 95% of the retriever's accuracy for both HybridFiD and FastFiD. This demonstrates their substantial capability to select valuable sentences. A comparative evaluation of FastFiD and HybridFiD indicates that the second-stage training has a minimal impact on the sentence selection performance. Its main contribution is to

[1] https://www.bing.com/
[2] https://chat.openai.com/

Model	NQ			TriviaQA			ASQA		
	EM	TPQ	Speed	EM	TPQ	Speed	STR-EM	TPQ	Speed
FiD	50.06	514	1.0X	**69.79**	550	1.0X	33.35	3,323	1.0X
FiD-Light	40.91	201	2.6X	63.15	**218**	**2.5X**	27.34	867	3.8X
HybridFiD	50.14	513	1.0X	69.77	540	1.0X	35.13	3,330	1.0X
FastFiD	**50.17**	**148**	**3.5X**	69.34	241	2.3X	**37.22**	**586**	**5.7X**

Table 2: Performance of vanilla FiD, FiD-Light, HybridFiD, FastFiD with 100 retrieved passages on test set (development set for ASQA). We select 200 sentences for NQ and ASQA, 400 sentences for TriviaQA. For FiD-Light, we utilize a value of 64 for k, which as demonstrated by Hofstätter et al. (2023), yields the best performance. TPQ is measured by milliseconds.

Model	# Doc	NQ-Dev	NQ-Test	Context Length	TPQ	Speed
FiD	25	47.33	47.23	9,600	197	1.0X
HybridFiD	25	**47.71**	**48.42**	9,600	194	1.0X
FastFiD	25	47.52	48.06	920	84	**2.4X**
FiD	50	47.79	47.89	19,200	354	1.0X
HybirdFiD	50	**48.12**	**49.09**	19,200	354	1.0X
FastFiD	50	47.96	48.89	1,035	110	**3.2X**
FiD	100	**49.10**	50.06	38,400	514	1.0X
HybirdFiD	100	48.65	50.14	38,400	513	1.0X
FastFiD	100	48.98	**50.17**	1,008	148	**3.5X**

Table 3: Detailed performance of vanilla FiD, HybridFiD and FastFiD on NQ with different number of passages.

adapt the model to the reduced context length, as we anticipated.

Discussion The performance of HybridFiD, as presented in Table 2 and Figure 3, highlights that answer generation and sentence selection are not mutually exclusive, and a multi-task training objective enables both capabilities. To further explore the relationship between sentence selection and answer generation, we examined the average cross-attention scores for tokens within the top 200 sentences and the non-selected segments. This analysis was conducted using HybridFiD with 100 retrieved passages on NQ. Following the approach of Izacard and Grave (2021a), we calculated the cross-attention score of each token in the inputs by averaging across all decoder layers, attention heads per layer, and all generated tokens.

Table 4 shows that the selected sentences have significantly higher average cross-attention scores compared to the non-selected segments, indicating that they contribute more significantly to the final answer generation. Conversely, this suggests that the non-selected segments largely contain irrelevant information, contributing less to answer generation despite being present in the context, and can therefore be disregarded during the decoding process. This insight also served as a motivation for our second-stage training, as described in Section 3.2. Furthermore, for a more comprehensive understanding of the effectiveness of our FastFiD approach, we provide a detailed case study in Appendix B.

	NQ-Dev	NQ-Test
Selected	**5.28E-4**	**5.32E-4**
Non-Selected	3.46E-5	3.43E-5

Table 4: Average cross-attention score for tokens in top-200 selected sentences and non-selected sentences for HybirdFiD with 100 retrieved passages.

5 Further Analysis

In this section, we conduct more experiments to show the effectiveness of our method. First, we compare the sentence selection method with passage reranking method in Section 5.1. Second, we compare our method with different number of selected sentences in Section 5.2. Finally, we make an ablation study to verify the importance of our

Model	# Doc	NQ-Dev	NQ-Test	Context Length
FastFiD	25	**47.52**	**48.06**	920
RerankFiD	25	46.42	47.20	1,152
FastFiD	50	**47.96**	**48.89**	1,035
RerankFiD	50	46.64	47.23	1,152
FastFiD	100	**48.98**	**50.17**	1,008
RerankFiD	100	46.45	48.09	1,152

Table 5: Comparison between FastFiD and RerankFiD among different number of retrieved passages. FastFiD consistently outperforms RerankFiD within similar context length.

two-stage training in Section 5.3.

5.1 Sentence Selection vs Passage Rerank

Similar to conducting sentence selection after the encoder, another method is to conduct passage rerank after encoder's outputs and thus reducing context length and inference time. In alignment with our two-stage training pipeline, we substitute the sentence selection loss with a passage reranking loss as utilized by Nogueira and Cho (2020), leading to a model we name RerankFiD. We evaluate the performance of FastFiD and RerankFiD under comparable context lengths, with the findings presented in Table 5. Consistently, our FastFiD method outperforms RerankFiD across a range of retrieved passage quantities. We hypothesize that this is due to the higher density of related information in the selected sentences compared to the reranked passages, as a passage often includes numerous irrelevant sentences even if it contains the correct answer.

5.2 Number of Selected Sentences

To evaluate the impact of varying the number of selected sentences, we conducted experiments on NQ with 100 retrieved passages. The results in Table 6 show that increasing the number of selected sentences leads to a nearly linear increase in the context length for the decoder. In terms of answer generation effectiveness, FastFiD performs well even with only 50 selected sentences and improves gradually with more sentences selected. It is worth noting that performance reaches a plateau after a certain number of sentences, such as 200. Beyond this point, selecting additional sentences does not yield further improvement but only increases context length and inference time.

Model	# Select Sentence	NQ-Dev	NQ-Test	Context Length
FiD	-	**49.10**	50.06	38,400
FastFiD	50	48.25	49.11	378
FastFiD	100	48.29	49.28	639
FastFiD	200	48.98	**50.17**	1,008
FastFiD	400	49.05	49.83	1,661

Table 6: Experiments on the number of selected sentences.

5.3 Two-Stage Training

To corroborate the efficacy of our two-stage training approach, we undertake experiments wherein each training stage is separately removed, with the outcomes displayed in Table 7. It is evident that the removal of either training stage results in a decrement in the final performance. Moreover, the second stage of training appears to be more consequential than the first stage, as demonstrated by the nearly 10-point drop in performance when the second stage is removed, compared to a decrease of less than 1-point when only the second stage is implemented.

Model	# Doc	# Select Sentence	NQ-Dev	NQ-Test
FastFiD	50	200	**47.96**	**48.89**
- 2nd-stage	50	200	36.61	37.67
- 1st-stage	50	200	47.62	48.03
FastFiD	100	200	**48.98**	**50.17**
- 2nd-stage	100	200	38.62	39.25
- 1st-stage	100	200	48.25	49.17

Table 7: Ablation study on two-stage training method.

6 Conclusion

In this paper, we present FastFiD, a model based on the FiD framework, designed to accelerate the inference process for ODQA tasks. FastFiD utilizes a two-stage training technique to enable the selection of valuable sentences and focus its predictions exclusively on these sentences. Experimental results demonstrate that FastFiD substantially improves inference speed while maintaining its original answer generation performance. And our ablation study confirms the effectiveness of the two-stage training approach, showing a decrease in final performance when any single training stage is omitted.

Limitations

The limitations of our FastFiD approach can be primarily summarized into the following two points:

- Firstly, due to constraints related to GPU resources, our experiments are performed using the T5-Base model, a comparatively modest model when compared with larger language models. While our method is theoretically adaptable to larger models, further experimentation using such models is anticipated to substantiate this claim.

- Secondly, with the widespread use of ChatGPT, most large language models currently in use are built on a decoder-only architecture. Unfortunately, this architecture is not compatible with the FiD framework utilized by our FastFiD method. As a result, FastFiD cannot be directly applied to the majority of existing large language models. Therefore, it becomes crucial and promising to explore acceleration techniques specifically designed for decoder-only models, as they face more significant challenges in terms of inference speed. And we believe that attempting to remove irrelevant information and reduce the context length will be a promising approach to address this challenge.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

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, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2020. Probabilistic assumptions matter: Improved models for distantly-supervised document-level question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5657–5667, Online. 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.

Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, Nicholas FitzGerald, Sumit Sanghai, Fei Sha, and William Cohen. 2023. FiDO: Fusion-in-decoder optimized for stronger performance and faster inference. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11534–11547, Toronto, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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*, pages 854–870, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Sebastian Hofstätter, Jiecao Chen, Karthik Raman, and Hamed Zamani. 2023. Fid-light: Efficient and effective retrieval-augmented text generation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1437–1447.

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

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.

Zhengbao Jiang, Luyu Gao, Zhiruo Wang, Jun Araki, Haibo Ding, Jamie Callan, and Graham Neubig. 2022. Retrieval as attention: End-to-end learning of retrieval and reading within a single transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2336–2349, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):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.

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.

Haejun Lee, Akhil Kedia, Jongwon Lee, Ashwin Paranjape, Christopher Manning, and Kyoung-Gu Woo. 2022. You only need one model for open-domain question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3047–3060, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021. Learning dense representations of phrases at scale. 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 6634–6647, Online. 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.

Patrick 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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Patrick 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. *Transactions of the Association for Computational Linguistics*, 9:1098–1115.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Yu A. Malkov and D. A. Yashunin. 2020. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard EM approach for weakly supervised question answering. 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 2851–2864, Hong Kong, China. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Rodrigo Nogueira and Kyunghyun Cho. 2020. Passage re-ranking with bert.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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. *Journal of Machine Learning Research*, 21(140):1–67.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. ArXiv.

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.

Devendra Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021a. 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.

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

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2023. Asqa: Factoid questions meet long-form answers.

Liang Wen, Houfeng Wang, Yingwei Luo, and Xiaolin Wang. 2022. M3: A multi-view fusion and multi-decoding network for multi-document reading comprehension. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1450–1461, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Yuxiang Wu, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2021. Training adaptive computation for open-domain question answering with computational constraints. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 447–453, Online. Association for Computational Linguistics.

Yuxiang Wu, Sebastian Riedel, Pasquale Minervini, and Pontus Stenetorp. 2020. Don't read too much into it: Adaptive computation for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3029–3039, Online. Association for Computational Linguistics.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Appendices

A Implementation Details

In the first stage of training, we employ a linear scheduler with a warmup ratio of 0.1 and a maximum learning rate of $1e-4$ for 10 epochs. The selection of the best checkpoint for the second-stage training is based on performance evaluation on the development set. In the second training stage, we use a constant learning rate of $5e-5$ for 5 epochs. We evaluate the performance of the hyperparameter λ in the training objective using values of 0.1 and 0.05, and select the one that yields better results for each dataset. Specifically, we use 0.1 for NQ and ASQA, and 0.05 for TriviaQA, considering its higher number of annotated sentences as indicated in Table 1.

During inference, we follow the approach of previous work (Hofstätter et al., 2023) by utilizing beam search with a beam size of 4. The maximum decoding length is set to 32 for NQ and TriviaQA, while it is set to 128 for ASQA due to the longer answer lengths in that dataset.

B Case Study

To demonstrate the effectiveness of our FastFiD approach, we present an example using the test set of NQ, as depicted in Figure 4. In this figure, the text highlighted in yellow represents the valuable sentences identified by FastFiD, which are subsequently utilized in the decoding process. It is evident that FastFiD possesses the capability to recognize valuable sentences that often contain the correct answer, even if they are not in the highly-ranked documents. Additionally, these valuable sentences only constitute a small portion of all the

Question: When is the next Deadpool movie being released?
Answer: May 18, 2018
Document [1] (Deadpool 2): Deadpool 2 is a 2018 American superhero film based on the Marve …
Document [2] (Deadpool 2): integrate him into the PG-13 MCU. Deadpool 2 is …
Document [3] (Deadpool 2): The film's score is the first to receive a parental advisory warning for explicit content, and the soundtrack also includes the original song "Ashes" by Céline Dion. "Deadpool 2" was released in the United States on May 18, 2018. It has grossed over $738 million worldwide, becoming the …
…
Document [15] (Deadpool (film)): "Deadpool 2" was released on May 18, 2018, with Baccarin, T. J. Miller, Uggams, Hildebrand, and Kapičić all returning. Josh Brolin joined them as Cable. The film explores the team X-Force, which includes Deadpool and Cable …
Document [16] (Deadpool 2): January, the film's release was moved up to May 18, 2018. In February 2018, Terry Crews was revealed to have a role in the film, the character Shatterstar was confirmed to be appearing, and the production returned to Vancouver for six days of reshoots under a new working title, "Daisy". Some reports emerged by mid-March claiming that these reshoots …
…
Document [99] (Josh Brolin): Summers / Cable in the "X-Men" film series. 2018's "Deadpool 2" is his first installment within that contract. He is set to reprise his role in ...

Figure 4: An example from the test set of NQ with 100 retrieved passages. The text highlighted in yellow represents the valuable sentences identified by our FastFiD.

retrieved passages which is important for us to accelerate inference. However, it is important to note that not all selected sentences are necessarily relevant to the given question. For instance, the second selected sentence in DOCUMENT [16] may not carry any meaningful information. Consequently, we need to select a specific number of sentences to retain all the pertinent information for achieving satisfactory performance, as demonstrated in Section 5.2.