M3: A Multi-View Fusion and Multi-Decoding Network for Multi-Document Reading Comprehension

Liang Wen[†,‡] **Houfeng Wang**[†] **Yingwei Luo**[†,‡] **Xiaolin Wang**[†,‡]
[†]School of Computer Science, Peking University, China
[‡]Peng Cheng Laboratory, Shenzhen, China
{yuco,wanghf,lyw,wxl}@pku.edu.cn

Abstract

Multi-document reading comprehension task requires collecting evidences from different documents for answering questions. Previous research works either use the extractive modeling method to naively integrate the scores from different documents on the encoder side or use the generative modeling method to collect the clues from different documents on the decoder side individually. However, any single modeling method cannot make full of the advantages of both. In this work, we propose a novel method that tries to employ a multi-view fusion and multi-decoding mechanism to achieve it. For one thing, our approach leverages question-centered fusion mechanism and cross-attention mechanism to gather fine-grained fusion of evidence clues from different documents in the encoder and decoder concurrently. For another, our method simultaneously employs both the extractive decoding approach and the generative decoding method to effectively guide the training process. Compared with existing methods, our method can perform both extractive decoding and generative decoding independently and optionally. Our experiments on two mainstream multi-document reading comprehension datasets (Natural Questions and TriviaQA) demonstrate that our method can provide consistent improvements over previous state-of-the-art methods.

1 Introduction

Open domain question answering (QA) aims to produce an answer for a given question using a large text corpus source such as Wikipedia. One of the typical approaches to open domain QA follows the retriever-reader framework (Chen et al., 2017; Karpukhin et al., 2020; Izacard and Grave, 2021b), where a retriever first identifies the most relevant documents, then a reader understands the retrieved documents and produces an answer. In this work, we focus on improving the effectiveness of the reader, whose goal is to efficiently aggregate and combine evidence from multiple documents for better answering questions, which is also known as multi-document reading comprehension (Hu et al., 2019).

Depending on the difference of decoding approaches, recent works on multi-document reading comprehension could be mainly divided into two categories: extractive approaches (Lee et al., 2019; Karpukhin et al., 2020; Guu et al., 2020) and generative approaches (Lewis et al., 2020b; Izacard and Grave, 2021b; Yu et al., 2022). To produce an answer, extractive approaches make predictions by extracting a contiguous span from the given evidence documents as answer. Since it is not straightforward for extractive models to aggregate and combine evidence from multiple documents (or paragraphs), various techniques have been put forward to address this limitation (Clark and Gardner, 2018; Wang et al., 2018; Min et al., 2019a). Besides, another drawback of extractive models is that they can't produce an answer string if it does not contained in the given evidence documents. Izacard and Grave (2021b) seek to address these issues in a generative way. In their method, they use a pre-trained generative model (Raffel et al., 2020) to perform evidence fusion in the decoder and sequentially generate an answer string, which is later wildly adopt in (Izacard and Grave, 2021a; Sachan et al., 2021; Cheng et al., 2021; Yu et al., 2022).

Recently, on multi-domain dialogue state tracking, Wu et al. (2019) achieve impressive performance by utilizing a generative model to *generate* slot values while using a classifier to *discriminate* whether the corresponding (domain, slot) pair is actually triggered. On open domain QA, Fajcik et al. (2021) show that ensembling the complementary results of *generate* and *extractive* approaches yields significant performance improvement. These hint that discriminative and generative objectives

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1450–1461
December 7-11, 2022 ©2022 Association for Computational Linguistics

could cooperate with each other and work well. Intuitively, to answer a question based on given evidence documents, people could effortless *extract* all the correct answer spans from the given documents and *generate* a valid answer if the evidence documents do not contain one. These inspire us to explore the joint advantage of using both the extractive modeling method and generative modeling method in multi-document reading comprehension.

In this paper, to have the best of both the extractive modeling method and the generative modeling method, we present M3: a **M**ulti-view fusion and **M**ulti-decoding network for **M**ulti-document reading comprehension. Unlike previous approaches, M3 not only performs evidence information fusion in the encoder and decoder simultaneously (multi-view fusion), but also is able to extract an answer or generate an answer at the same time (multi-decoding). More precisely, we take advantage of a question-centered fusion mechanism to gather question-related clues from different documents on the encoder side and meanwhile make use of a cross-attention mechanism (Vaswani et al., 2017) to aggregate and combine evidence from multiple passages in the decoder. To aggregate useful training signals from both the encoder side and the decoder side, we equip M3 with an extractive reader in the encoder and a generative reader on the decoder side concurrently.

We evaluate our proposed approach by experimenting on two commonly used open domain QA datasets: Natural Questions (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). Experimental results show that M3 exhibits a consistent better generative performance for various generative models of comparable size on all datasets, outperforming recent generative approaches by about 2 exact match points. We also find that the extraction performance of M3 are superior to those of many typical extractive models that have similar or more model parameters. Last but not least, based on careful ablation studies, we demonstrate that the proposed multi-view information integration and the proposed multi-decoding mechanism are the key elements that lead to performance improvement.

2 Model

Our proposed model M3 consists of three components: a document content extraction module, a cross-document information integration module and a multivariate heterogeneous decoding module. Figure 1 gives an overview of the architecture of our model. In the following sections, we will describe each component and our training objective in detail.

2.1 Document Content Extraction

Let the set of evidence documents be denoted by $D = \left\{D^1, D^2, \ldots, D^k, \ldots, D^K\right\}$. Given a question $Q = \{q_1, q_2, \ldots, q_m\}$, the goal of the document content extraction module is to extract question-related clues from each document. We model the extraction module as a transformer encoder (Vaswani et al., 2017).

Specifically, the encoder we use is an encoder of a pre-trained sequence-to-sequence transformer T5 (Raffel et al., 2020) that consists of an encoder $\mathcal{G}_e$ and a decoder $\mathcal{G}_d$. For each document $D^k = \left\{d_1^k, d_2^k, \ldots, d_n^k\right\}$, we first append its title T^k and the question as follows [1]:

$$I^k = \left[[Q, [SEP], T^k, [SEP], D^k\right] \quad (1)$$

where $[SEP]$ is a space separator for distinguishing different parts of the input.

Then, we independently feed each I^k to the T5 encoder $\mathcal{G}_e$ and acquire the output representations H^k corresponding to I^k as :

$$H^k = \mathcal{G}_e\left(I^k\right) \in \mathbb{R}^{d \times N} \quad (2)$$

where $1 \leq k \leq K$, N is the number of tokens in I^k, and d is the hidden size of the T5 encoder $\mathcal{G}_e$.

2.2 Cross-Document Information Integration

The cross-document information integration module aims at synthesizing information distributed across multiple documents. In M3, we propose two techniques to aggregate scattered evidence: one is to use question-centered fusion mechanism to integrate question-related clues from different document on the encoder side; the other is to employ cross-attention mechanism to performs evidence fusion in the decoder.

Question-centered fusion mechanism For each question token q_i, based on the output representations H^k $(1 \leq k \leq K)$ produced by document content extraction module, we first take all its hidden representations as:

$$R_i = \left[H_i^1, H_i^2, \ldots, H_i^k, \ldots, H_i^K\right] \in \mathbb{R}^{d \times K} \quad (3)$$

[1] As in Raffel et al. (2020), we also add special tokens `question:`, `title:` and `context:` before the question, title and text of each document before concatenating.

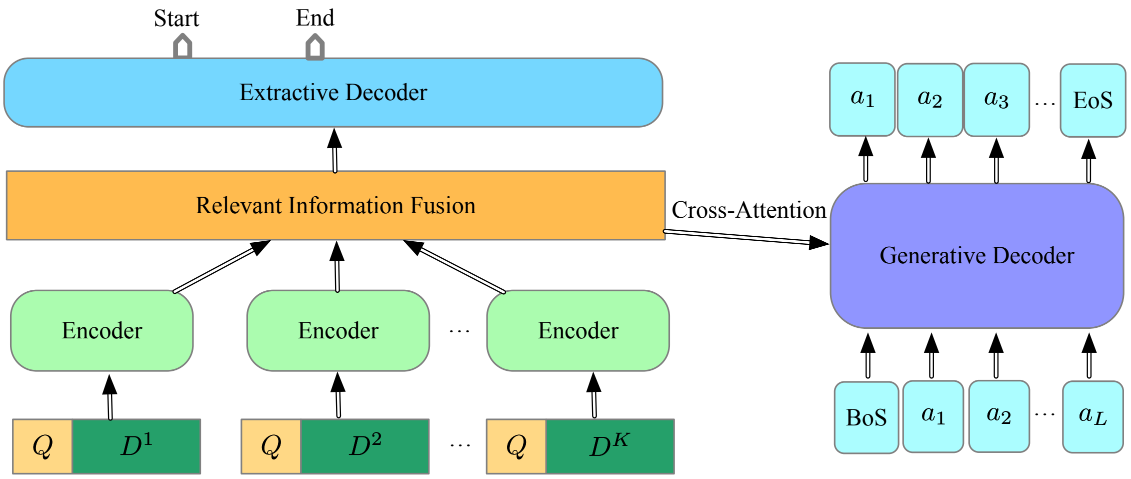


Figure 1: An illustration of our model M3. It consists of three components: a document content extraction module (on the bottom left), a cross-document information integration module (in the middle of the left side) and a multivariate heterogeneous decoding module (on the top left and right). And the same colored blocks of encoder component indicate that they share the same trainable parameters.

where K is the number of documents, H_i^k is the corresponding representation of the i-th question token from input sequence I^k, and $[\cdot\ ,\ \cdot]$ denotes the concatenation operation along the row.

Then, to take a comprehensive consideration of all question-related information, we use a token-wise multi-head self-attention mechanism as Vaswani et al. (2017) and update R_i as:

$$\begin{pmatrix} Q_i \\ K_i \\ V_i \end{pmatrix} = \begin{pmatrix} W_q \\ W_k \\ W_v \end{pmatrix} R_i + \begin{pmatrix} b_q \\ b_k \\ b_v \end{pmatrix} \tag{4}$$

$$\hat{V}_i = \text{Softmax}\left(\frac{Q_i {K_i}^T}{\sqrt{\hat{\lambda}}}\right) V_i \tag{5}$$

$$\hat{R}_i = \text{T5LayerFF}\left(\hat{V}_i\right) \in \mathbb{R}^{d\times K} \tag{6}$$

where $\hat{\lambda}$ is the scaling factor, W_q, W_k, W_v, b_q, b_k and b_v are learnable parameters. T5LayerFF denotes a single-layer feedforward network as defined in (Raffel et al., 2020).

Next, we use a selective gate to fuse the representation R_i and its updated representation $\hat{\mathrm{R}}_\mathrm{i}$:

$$B = \left[R_i; \hat{R}_i; R_i \circ \hat{R}_i; R_i - \hat{R}_i\right] \in \mathbb{R}^{4d\times K} \tag{7}$$

$$E = \text{Relu}\left(W_e B + b_e\right) \in \mathbb{R}^{d\times K} \tag{8}$$

$$G = \sigma\left(W_g B + b_g\right) \in \mathbb{R}^{d\times K} \tag{9}$$

$$\tilde{R}_i = G \circ E + (1 - G) \circ R_i \in \mathbb{R}^{d\times K} \tag{10}$$

where $[\cdot\ ;\ \cdot]$ denotes the concatenation operation along the column, W_e, b_e, W_g and b_g are trainable parameters, and the gated fusion representations, corresponding to the representation R_i, are denoted as $\tilde{R}_i = \left[\tilde{H}_i^1, \tilde{H}_i^2, \ldots, \tilde{H}_i^k, \ldots, \tilde{H}_i^K\right] \in \mathbb{R}^{d\times K}$.

Finally, for each input sequence I^k, we employ another multi-head self-attention operation to spread the gated fusion representations of the whole question $X^k = \left[\tilde{H}_1^k, \tilde{H}_2^k, \ldots, \tilde{H}_m^k\right] \in \mathbb{R}^{d\times m}$, to its corresponding document content representations $C^k = \left[H_{m+1}^k, H_{m+2}^k, \ldots, H_N^k\right]$ as:

$$\begin{pmatrix} \overline{Q} \\ \overline{K} \\ \overline{V} \end{pmatrix} = \begin{pmatrix} \overline{W}_q \\ \overline{W}_k \\ \overline{W}_v \end{pmatrix} \left[X^k, C^k\right] + \begin{pmatrix} \bar{b}_q \\ \bar{b}_k \\ \bar{b}_v \end{pmatrix} \tag{11}$$

$$\overline{Y}^k = \text{Softmax}\left(\frac{\overline{QK}^T}{\sqrt{\overline{\lambda}}}\right) \overline{V} \tag{12}$$

$$\mathrm{Y}^\mathrm{k} = \text{T5LayerFF}\left(\overline{Y}^k\right) + H^k \tag{13}$$

where $Y^k \in \mathbb{R}^{d\times N}$, $\overline{W}_q$,$\overline{W}_k$, $\overline{W}_v$,$\bar{b}_q$ $\bar{b}_k$ $\bar{b}_v$ are parameters to be trained, and $\overline{\lambda}$ a scaling factor.

Overall, we think that the proposed question-centered fusion mechanism describe a way to adjust the hidden state H^k globally. In practice, we tend to update the hidden state H^k multiple times. And each update could be seen as an information integration across different documents in the encoder. Here, we denote the times we used as η and its default value is set to 2.

Cross-attention mechanism While the question-centered fusion mechanism is designed to aggregate question-focused information on the encoder side, the cross-attention mechanism is used to performs evidence fusion in the decoder. Specifically, we first concatenate the output representations corresponding to all of the input documents as:

$$O = \left[H^1, \ldots, H^k, \ldots, H^K\right] \in \mathbb{R}^{d\times(N\star K)} \quad (14)$$

Then, the concatenated representation O is used as an input to the T5 decoder $\mathcal{G}_d$ as in Izacard and Grave (2021b). Since O contains information from multiple documents, the decoder could aggregate evidence contained in these documents by performing cross-attention over O as in (Vaswani et al., 2017) :

$$\tilde{Q} = W_{\tilde{q}} U \quad \tilde{K} = W_{\tilde{k}} O, \quad \tilde{V} = W_{\tilde{v}} O \quad (15)$$

$$\alpha_{i,j} = \tilde{Q}_i^T \tilde{K}_j, \quad \tilde{\alpha}_{i,j} = \frac{\exp\left(\alpha_{i,j}\right)}{\sum_{nk} \exp\left(\alpha_{i,nk}\right)} \quad (16)$$

$$\tilde{O}_i = W_{\tilde{O}} \sum_j \tilde{\alpha}_{i,j} \tilde{V}_{i,j} \quad (17)$$

where U denotes the output of the previous self-attention layer of the decoder, $W_{\tilde{q}}$, $W_{\tilde{k}}$, $W_{\tilde{v}}$ and $W_{\tilde{O}}$ are learnable parameters. Note that, in case of multi-head attention and a stack of transformer-decoders, the above operations are repeatedly performed in parallel with different linear transformations. See Vaswani et al. (2017) for more details.

Last, based on the final attended representations $\tilde{O}$, the probability of generating the answer $\boldsymbol{a}$ is defined as:

$$p(\boldsymbol{a} \mid \tilde{O}; \Theta) = \prod_{t=1}^{L} p\left(a_t \mid \boldsymbol{a}_{<t}, \tilde{O}; \Theta\right) \quad (18)$$

where Θ denotes the parameters and L is the number of answer tokens. In the experiment, we use greedy decoding and keep generating answer tokens until reaching the pre-specified maximum answer length or meeting a special EoS token.

2.3 Multivariate Heterogeneous Decoding

In contrast to previous work on obtaining answers, we train both an extractive reader and a generative reader jointly in an end-to-end differentiable fashion, so that we can take advantage of the fusion of the two heterogeneous signals in the training stage and perform both extractive decoding and generative decoding independently and optionally at inference.

Concretely, for the extractive reader, we first concatenate the representations of all the documents defined by Y^k $(1 \leq k \leq K)$ as:

$$Z = \left[Y^1_{m+1:N}, Y^2_{m+1:N}, \ldots, Y^K_{m+1:N}\right] \quad (19)$$

Then, we decompose the answer span prediction into predicting the start and end positions of the answer span as in Lin et al. (2018):

$$p^s = \operatorname{Softmax}\left(W^s Z\right) \quad (20)$$

$$p^e = \operatorname{Softmax}\left(W^e Z\right) \quad (21)$$

$$p\left(\boldsymbol{a} \mid Z; \Phi\right) = \sum_{j=1}^{|\tau|} p^s\left(a_s^j\right) p^e\left(a_e^j\right) \quad (22)$$

where W^s, W^e and Φ are trainable parameters. Since we don't know the position of the answer exactly in multi-document reading comprehension, we may have several tokens matched to the correct answer in the given document set. Here, we suppose the set $\tau = \left\{\left(a_s^1, a_e^1\right), \left(a_s^2, a_e^2\right), \cdots, \left(a_s^{|a|}, a_e^{|a|}\right)\right\}$ includes the start and end positions of the tokens matched to answer $\boldsymbol{a}$. And $p^s\left(a_s^j\right)$ and $p^e\left(a_e^j\right)$ are the probabilities of a_s^j and a_e^j being start and end words respectively. Finally, the training loss of the extractive reader is defined as:

$$\mathcal{L}_e = -\log p\left(\boldsymbol{a} \mid Z; \Phi\right) \quad (23)$$

For the generative reader, the objective function is simply defined based on $p(\boldsymbol{a} \mid O; \Theta)$ which is described in equation 18:

$$\mathcal{L}_g = -\log p(\boldsymbol{a} \mid \tilde{O}; \Theta) \quad (24)$$

Last but not least, the loss function for the whole model is defined as:

$$\mathcal{L} = \lambda \mathcal{L}_e + \mathcal{L}_g \quad (25)$$

where λ is a hyperparameter that defines the heterogeneous decoding weight and its default value is 0.1.

3 Experiments

3.1 Datasets and Evaluation metrics

We conduct experiments on two mainstream question answering benchmarks – Natural Questions

Datasets	#train	#val	#test	Qlen	Alen
NQ	79,168	8,757	3,610	12.5	5.2
TriviaQA	78,785	8,837	11,313	20.2	5.5

Table 1: Statistics of NQ and TriviaQA dataset. #train, #val, #test: the number of samples in the training, validation, test set. Qlen: the average question length. Alen: the average answer length.

(NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). The questions of NQ were mined from Google search queries and the support documents are made of Wikipedia documents. TriviaQA contains a set of trivia questions that are gathered from trivia and quiz-league websites, and its answers are also scraped from the Web. The statistics of the two datasets are summarized in table 1.

Following prior studies, we use Exact Match (EM) to evaluate our model. The EM metric is originally introduced by Rajpurkar et al. (2016), in which a predicted answer is deemed correct if it matches any answer of the list of gold answers after simple normalization.

3.2 Implementation details

Due to GPU constraints, we only use the base configuration of the pre-trained T5 (Raffel et al., 2020) as our backbone model and we adopt the same hyperparameter settings for all the proposed models and their variants if not specified. And we train our models using AdamW (Loshchilov and Hutter, 2019) optimizer and use a dropout rate of 10%. The training batch size is set to 64. The number of gradient steps is set to 15K. For the learning rate, we adopt a peak learning rate of 10^{-4} which increases linearly during the first 600 gradient steps, and decreases linearly during the rest. During both training and testing, the number of input passages is set to 100 and the maximum length of each passage is limited to 250 word-piece tokens. And at inference, we adopt the greedy decoding strategy to generate answers on the generative reader side. For both the extractive and generative reader, we set the maximum length of answer span to be 30.

As for the data pre-processing, we follow the same setting as Izacard and Grave (2021a). Since our goal is to improve the effectiveness of the machine reader and consequently improve the performance of the whole QA system, we use the support documents retrieved by Izacard and Grave (2021a) throughout our experiments. And we implement our models based on the HuggingFace Transformers library [2]. All our experiments are conducted on 8 Tesla A100 40GB GPUs.

3.3 Baselines

We compare our models with several recent baselines. Since our proposed model M3, once trained, could perform both extractive decoding and generative decoding at the same time, we categorize the baselines into the following two classes:

- Extractive models: These models process each passage individually and use modified objective function (Clark and Gardner, 2018) or other techniques (Min et al., 2019a) that induces models to produce globally agreed output. Such models predict a span from input passages as answer. Here, we mainly consider the following methods: Hard EM(Min et al., 2019a), Path Retriever(Asai et al., 2020), BM25 + BERT (Lee et al., 2019), ORQA (Lee et al., 2019), Graph Retriever (Min et al., 2019b), REALM (Guu et al., 2020), DPR (Karpukhin et al., 2020), GAR (Mao et al., 2021a), GAR+DPR (Mao et al., 2021a).

- Generative models: Some of these models are closed-book, large-scale generative language models, including GPT-3 (Brown et al., 2020) and T5 (Raffel et al., 2020). And others use a sequence-to-sequence model to combine evidence in the decoder only and generate answers in an autoregressive manner, which include SpanSeqGen (Min et al., 2020), RAG (Lewis et al., 2020b), GAR (Mao et al., 2021a), GAR+DPR (Mao et al., 2021a), RIDER (GAR) (Mao et al., 2021b), RIDER (GAR+DPR) (Mao et al., 2021b), FID-base (Izacard and Grave, 2021b), FID-large (Izacard and Grave, 2021b), FID+DK (Izacard and Grave, 2021a), KG-FiD (Yu et al., 2022).

3.4 Comparsions to state-of-the-art baselines

In table 2, we compares the experimental results obtained by our proposed model M3 with existing approaches. Following standard conventions, we report exact match scores on the NQ and TriviaQA benchmarks. Since the proposed model M3 could perform both extractive decoding and generative decoding independently and optionally at inference,

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

Model	Reader Type	#Parameters	NQ	TriviaQA
Hard EM (Min et al., 2019a)	Extractive	110M	28.1	50.9
Path Retriever (Asai et al., 2020)	Extractive	110M	32.6	–
BM25 + BERT (Lee et al., 2019)	Extractive	110M	26.5	47.1
ORQA (Lee et al., 2019)	Extractive	330M	33.3	45.0
Graph Retriever (Min et al., 2019b)	Extractive	110M	34.5	56.0
REALM (Guu et al., 2020)	Extractive	330M	40.4	–
DPR (Karpukhin et al., 2020)	Extractive	110M	41.5	57.9
GAR (Mao et al., 2021a)	Extractive	110M	41.8	**62.7**
GAR+DPR (Mao et al., 2021a)	Extractive	110M	43.8	–
M3 (Ours, Extractive Decoding)	Extractive	110M†	**48.1**	61.2
T5 (Raffel et al., 2020)	Generative	11B	32.8	42.9
GPT-3 few shot (Brown et al., 2020)	Generative	175B	29.9	–
SpanSeqGen (Min et al., 2020)	Generative	400M	42.2	–
RAG (Lewis et al., 2020b)	Generative	400M	44.5	56.1
FID-base (Izacard and Grave, 2021b)	Generative	220M	48.2	65.0
FID-large (Izacard and Grave, 2021b)	Generative	770M	51.4	67.6
GAR (Mao et al., 2021a)	Generative	400M	38.1	62.2
RIDER (GAR) (Mao et al., 2021b)	Generative	400M	–	66.4
GAR+DPR (Mao et al., 2021a)	Generative	400M	45.3	–
RIDER (GAR+DPR) (Mao et al., 2021b)	Generative	400M	48.3	–
FID+DK (Izacard and Grave, 2021a)	Generative	220M	49.6	68.8
KG-FiD (Yu et al., 2022)	Generative	220M	49.6	66.7
M3 (Ours, Generative Decoding)	Generative	220M	**51.7**	**69.9**

Table 2: Comparison to state-of-the-art models on the test sets of NQ and TriviaQA. To provide a fair comparison, we show results from other works with the T5-base configuration when not specified, except for methods that use BART-large(Lewis et al., 2020a) as their backbone models. † indicates that we only need the encoder part of the original T5-base model to perform extractive decoding.

we divide the table into two main sections: extractive models and generative models. It is worth noting that, unlike previous work GAR (Mao et al., 2021a) that requires training an extractive reader and a generative reader separately, our M3 could be trained in a end-to-end fashion while being able to be used as an extractive reader and a generative reader optionally.

From table 2, we can observe that when using the extractive reader to make predictions, M3 consistently outperforms previous extractive models on both NQ and TriviaQA datasets, despite that it only make use of the encoder part of a pre-trained encoder-decoder architecture. This shows that the proposed cross-document information integration on the encoder side is effective and capable of combining evidence from multiple documents. And we also think that the much better extraction performance of M3 might partly benefit from the joint optimization objective since the generative objective function could better handle the distant noisy supervision issue where no gold spans are given to the system, but only the correct answer.

Comparing M3 with previous generative models, we can see that the proposed M3 clearly has a better performance on the NQ and TriviaQA benchmarks. In particular, M3 not only performs better than FID-large, GAR, RIDER (GAR) and RIDER (GAR+DPR) that have more model parameters, but also outperforms the very recent state-of-the-art models (FID+DK and KG-FiD) by 2.1 points on NQ and 1.1 points on TriviaQA. These results demonstrate that aggregating and combining evidence in the encoder and decoder concurrently and the proposed heterogeneous decoding strategy are advantageous over previous methods. In addition, we also find that the generative performance of M3 is better than the extraction performance of M3. We conjecture that it is due to the following two reasons. First, the backbone model of M3 is a generative pre-trained language model—T5-base, which is good at generative taks indeed. Second, when the given documents don't contain an answer span, the generative reader can generate it while the extractive reader cannot.

3.5 Ranking Results

Since our model M3 could perform both extractive decoding and generative decoding at inference time, we also wonder whether aggregating and ranking the outputs of the two decoder could provide additional performance gain.

To achieve so, we adopt an answer ranker module that rescores the topK (K=30) extracted answers produced by the extractive reader and the generated answer obtained via generation-based greedy decoding. Specifically, we simply score each candidate answer $\boldsymbol{c_k}$ using the likelihood of generating the candidate conditioned on the given question and document set:

$$\mathcal{S}\left(\boldsymbol{c_k}\right) = \log p(\boldsymbol{c_k} \mid Q; D) \tag{26}$$

After the ranking phase, we re-evaluate M3's performance. On the NQ dataset, we obtain a final score of 52.1 EM, which gains a +4.0 EM improvement over M3 (Extractive Decoding) and a +0.4 EM improvement over M3 (Generative Decoding).

4 Ablations

In this section, we investigate design choices regarding the key elements of our method: the multi-view evidence fusion and the multivariate heterogeneous decoding strategy. Specifically, we mainly consider the following variants of M3:

Variant 1: This variant perform evidence fusion in the decoder only and is equipped with a generate reader only, which is a Fusion-in-Decoder model (Izacard and Grave, 2021a,b) indeed.

Variant 2: This variant perform evidence fusion in the encoder only and is equipped with an extractive reader only.

Variant 3: This variant perform evidence fusion both in the encoder and decoder and is equipped with a generative reader only.

Variant 4: This variant is the same as M3 in model architecture, but it is only trained with filtered samples, in each of which at least one of the reference answers must appear in the given documents [3].

Model	EM (Generative)	EM (Extractive)
M3	**51.7**	48.1
Variant 1	50.1	–
Variant 2	–	**48.4**
Variant 3	50.6	–
Variant 4	51.2	48.2

Table 3: Exact match scores of M3 and its variants on NQ dataset.

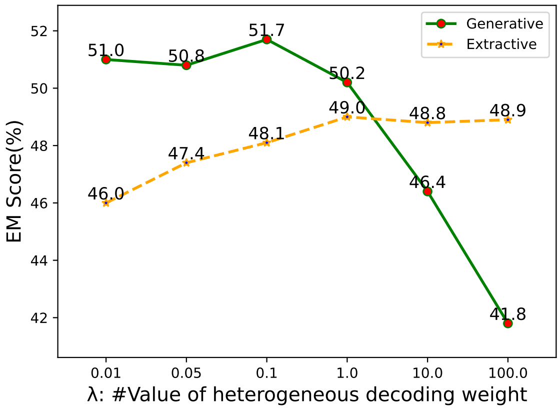


Figure 2: Performance of M3 on NQ as a function of the heterogeneous decoding weight.

Table 3 shows the performance of M3 and its variants on NQ dataset. From it, We note the following observations: (1) M3 outperforms Variant 1 by 1.6% EM (Generative), which signifies the proposed multi-view information integration and the heterogeneous decoding strategy can obviously improve performance. (2) The extractive EM score of Variant 2 is slightly higher than M3's extraction performance, which seem that the heterogeneous decoding strategy may be counterproductive. However, further study on the impact of the heterogeneous decoding weight confirm that it is not true (See section 5.1 for details). (3) Variant 3 is 1.1% EM behind M3's generative performance, which shows that the heterogeneous supervision signal from the extractive reader helps to improve model's performance. (4) Comparing M3 with Variant 4, we could find that the filtered samples is helpful for obtaining better generative performance.

5 Analysis

5.1 Impact of different heterogeneous decoding weights

To get insights into how the extractive reader of M3 affects the generative reader of M3 and vice versa,

[3] For M3, we train it using all the available training samples. When none of the reference answers can be found in the given documents, we simply set the heterogeneous decoding weight defined in equation 25 to be zero and train it as usual.

Objective	EM (Generative)	EM(Extractive)
SUM	51.7	48.1
MAX	**51.8**	48.1
MULTI-OBJ	51.4	**48.2**

Table 4: Performance of M3 under different extractive objectives

we report the performance of M3 with respect to the value of the heterogeneous decoding weight defined in equation 25. The experimental results are shown in figure 2. From it, we could see that increasing the value of the heterogeneous decoding weight leads to relatively stable improvement of M3's extraction performance on NQ. On the other hand, the generative performance of M3 seems to peak around 0.1. Moreover, the best extraction performance of M3 is reached when the heterogeneous decoding weight is set to be 1.0. Hence, we believe that this is evidence that coupling the extractive method and the generative method helps to aggregate and combine evidence informations from multiple documents.

5.2 Effect of different extractive objective functions

We also investigate the performance of M3 on NQ with respect to different extractive objective functions: SUM, MAX and MULTI-OBJ. Here, the SUM objective is defined in equation 22 and is the default one, the MAX objective assume that only one span from the given documents indicates the correct answer and it is described in Lin et al. (2018), and the MULTI-OBJ objective (Cheng et al., 2021) combines a multi-passage HardEM (Min et al., 2019a) loss with passage-level marginal log-likelihood losses. Table 4 show the results. From it, we can find that different objective functions have little impact on M3's performance and using the latest MULTI-OBJ objective leads to the best extraction performance while owning a slightly worse generative performance.

5.3 Performance with different times of information integration

We also report the performance obtained by training with different times of information integration in the encoder. The experimental results are shown in figure 3. From it, we can observe that the model with two times of information integration achieves the best generative performance. And the optimal extraction performance is obtained when only one times of information integration is used. This indicates that the required times of information integration on the encoder side changes with different types of decoding methods. It may be due to that the proposed question-centered fusion mechanism and cross-attention mechanism play different roles in aggregating and combining evidence from multiple documents.

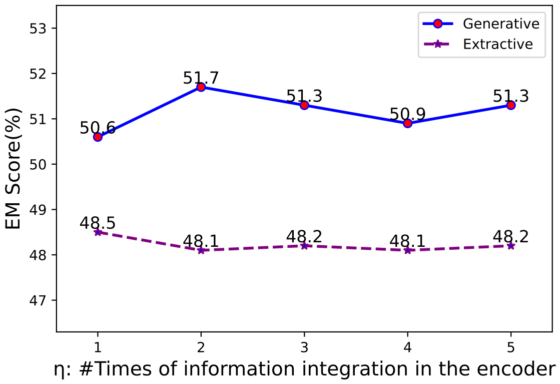
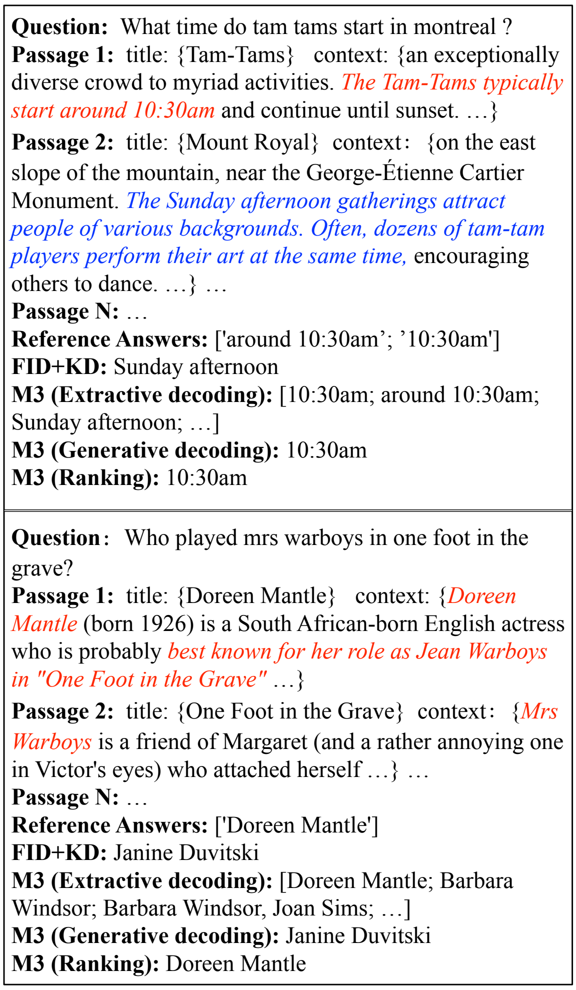


Figure 3: Performance on NQ as more times of information integration on the encoder side are used.

5.4 How M3 behaves differently from FiD+DK?

To get an intuitive understanding of M3' behavior, we conduct further analyses into prediction divergence made by model FiD+DK (Izacard and Grave, 2021a) and our model M3. Figure 4 shows the prediction results on two examples from NQ dataset.

From the first example depicted in figure 4, we can see that FiD+DK mistakenly takes "Sunday afternoon" as the answer, while M3 consistently make the right prediction no matter in what decoding strategies (extractive, generative, or ranking). This implies that the proposed method may help to reduce the errors caused by confusing the related concepts. From the second example, we notice that, both FiD+DK and M3 (Generative decoding), which generate answers word by word, make the wrong predictions. However, when adopting extractive decoding or using another ranking step, M3 could obtain the correct answer. This may suggest that extractive reader is complementary to generative reader. And it also account for that combining the prediction results from extractive and generative reader could yield further perform improvement.



Question: What time do tam tams start in montreal ?
Passage 1: title: {Tam-Tams} context: {an exceptionally diverse crowd to myriad activities. *The Tam-Tams typically start around 10:30am* and continue until sunset. …}
Passage 2: title: {Mount Royal} context: {on the east slope of the mountain, near the George-Étienne Cartier Monument. *The Sunday afternoon gatherings attract people of various backgrounds. Often, dozens of tam-tam players perform their art at the same time,* encouraging others to dance. …} …
Passage N: …
Reference Answers: ['around 10:30am'; '10:30am']
FID+KD: Sunday afternoon
M3 (Extractive decoding): [10:30am; around 10:30am; Sunday afternoon; …]
M3 (Generative decoding): 10:30am
M3 (Ranking): 10:30am

Question: Who played mrs warboys in one foot in the grave?
Passage 1: title: {Doreen Mantle} context: {*Doreen Mantle* (born 1926) is a South African-born English actress who is probably *best known for her role as Jean Warboys in "One Foot in the Grave"* …}
Passage 2: title: {One Foot in the Grave} context: {*Mrs Warboys* is a friend of Margaret (and a rather annoying one in Victor's eyes) who attached herself …} …
Passage N: …
Reference Answers: ['Doreen Mantle']
FID+KD: Janine Duvitski
M3 (Extractive decoding): [Doreen Mantle; Barbara Windsor; Barbara Windsor, Joan Sims; …]
M3 (Generative decoding): Janine Duvitski
M3 (Ranking): Doreen Mantle

Figure 4: Examples of prediction divergence. For M3 (Extractive decoding), we only list the top 3 candidates.

6 Related Work

Open domain QA aims to answer general domain questions using a large collection of documents. It's a longstanding problem in natural language processing (Voorhees and Tice, 2000) and has regained popularity since the work published by Chen et al. (2017). In recent years, various models are put forward (Seo et al., 2019; Raffel et al., 2020; Brown et al., 2020; Lee et al., 2021) and Chen and Yih (2020) give a nice tutorial on this topic. Among these approaches, the retriever-reader method (Wang et al., 2019; Karpukhin et al., 2020) is one of the most promising one, in which the core elements include a document retriever and a multi-document reader.

Extractive Machine Reader is wildly used in multi-paragraph or multi-document reading comprehension task. Wang et al. (2018) propose to aggregate answers from different paragraphs using confidence and coverage scores. Clark and Gardner (2018) propose to use a global shared-normalization over all possible span corresponding to the answer, which is later applied to BERT-based models (Yang et al., 2019; Wang et al., 2019). Min et al. (2019a) utilize an hard expectation maximization technique to tackle the distant noisy supervision issue from multi-document reading comprehension. Besides, similar ways are adopt in (Lin et al., 2018; Cheng et al., 2021).

Generative Machine Reader is mostly used in previous reading comprehension tasks that require to generate answers, such as MS MARCO (Nguyen et al., 2016), ELI5 (Fan et al., 2019). Raffel et al. (2020) show that generative models are competitive in extractive reading comprehension tasks, like SQuAD (Rajpurkar et al., 2016). Afterwards, Izacard and Grave (2021b) demonstrate that using a sequence-to-sequence model to perform evidence fusion in the decoder leads to remarkable performance improvement in open domain QA. This method is later wildly adopt in multi-document reading comprehension (Cheng et al., 2021; Izacard and Grave, 2021a; Yu et al., 2022).

Similar to our work, Fajcik et al. (2021) propose to combine the predictions of both extractive and generative reader, but our model architecutre is different from them and they use more model parameters. Another work close to ours is the approach proposed by (Su et al., 2022), where they first use a SpanBERT (Joshi et al., 2020) to collect answer-related salient information and then combine another generative model BART-large (Lewis et al., 2020a) to make final predictions. Differs from them, we herein perform evidence fusion in both the encoder and decoder and take advantage of both the generative training signal and extractive training signal to improve model performance.

7 Conclusion

In this work, we propose M3, an effective multi-document reading comprehension models, which perform evidence information integration from the perspective of both the encoder and the decoder. Without additional training, M3 could perform extractive decoding and generative decoding individually and optionally. Experiment results on two mainstream open domain QA datasets show that the proposed model M3 achieves better generative performance than state-of-the-art generative methods and obtain competitive or better performance than previous typical extractive models.

Limitations

One limitation of our work is that the proposed model M3 is based on a pre-trained sequence-to-sequence model. It would be interesting to pre-train a cross-document language model that couples masked language modeling pre-training with denoising sequence-to-sequence pre-training and evaluate its performance on multi-document tasks.

Acknowledgements

We thank all the reviewers for their valuable comments and helpful suggestions. We also thank Baidu's Search Science Team for computing support. The work is supported by National Natural Science Foundation of China (Grant No.62036001, No.62032001) and PKU-Baidu Fund (No. 2020BD021). The corresponding author of this paper is Houfeng Wang.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1870–1879. Association for Computational Linguistics.

Danqi Chen and Wen-tau Yih. 2020. Open-domain question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts, ACL 2020, Online, July 5, 2020*, pages 34–37. 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.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 845–855. 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.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. 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.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. Retrieve, read, rerank: Towards end-to-end multi-document reading comprehension. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2285–2295. Association for Computational Linguistics.

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

Gautier Izacard and Edouard Grave. 2021b. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, pages 874–880. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020.

Spanbert: Improving pre-training by representing and predicting spans. *Trans. Assoc. Comput. Linguistics*, 8:64–77.

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 S. H. 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 2020, Online, November 16-20, 2020*, pages 6769–6781. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. 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. *Trans. Assoc. Comput. Linguistics*, 7:452–466.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6634–6647. 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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6086–6096. Association for Computational Linguistics.

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

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

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1736–1745. Association for Computational Linguistics.

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.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021a. Generation-augmented retrieval for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4089–4100. Association for Computational Linguistics.

Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2021b. Reader-guided passage reranking for open-domain question answering. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 344–350. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019a. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 2851–2864. Association for Computational Linguistics.

Sewon Min, Danqi Chen, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019b. Knowledge guided text retrieval and reading for open domain question answering. *CoRR*, abs/1911.03868.

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 2020, Online, November 16-20, 2020*, pages 5783–5797. Association for Computational Linguistics.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 co-located with the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, December 9, 2016*, volume 1773 of *CEUR Workshop Proceedings*. CEUR-WS.org.

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Devendra Singh Sachan, Mostofa Patwary, Mohammad Shoeybi, Neel Kant, Wei Ping, William L. Hamilton, and Bryan Catanzaro. 2021. End-to-end training of neural retrievers for open-domain question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6648–6662. Association for Computational Linguistics.

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.

Dan Su, Xiaoguang Li, Jindi Zhang, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. 2022. Read before generate! faithful long form question answering with machine reading. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 744–756, Dublin, Ireland. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

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

Zhiguo Wang, Patrick Ng, Xiaofei Ma, Ramesh Nallapati, and Bing Xiang. 2019. Multi-passage BERT: A globally normalized BERT model for open-domain 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 5878–5882, Hong Kong, China. Association for Computational Linguistics.

Chien-Sheng Wu, Andrea Madotto, Ehsan Hosseini-Asl, Caiming Xiong, Richard Socher, and Pascale Fung. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 808–819. Association for Computational Linguistics.

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

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), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 4961–4974. Association for Computational Linguistics.