Narrative Question Answering with Cutting-Edge Open-Domain QA Techniques: A Comprehensive Study

Xiangyang Mou* **Chenghao Yang*** **Mo Yu***
Bingsheng Yao **Xiaoxiao Guo** **Saloni Potdar** **Hui Su**

Rensselaer Polytechnic Institute & IBM

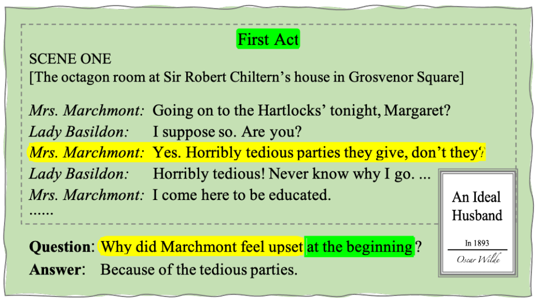
moux4@rpi.edu gflfof@gmail.com

Abstract

Recent advancements in open-domain question answering (ODQA), i.e., finding answers from large open-domain corpus like Wikipedia, have led to human-level performance on many datasets. However, progress in QA over book stories (Book QA) lags behind despite its similar task formulation to ODQA. This work provides a comprehensive and quantitative analysis about the difficulty of Book QA: (1) We benchmark the research on the NarrativeQA dataset with extensive experiments with cutting-edge ODQA techniques. This quantifies the challenges Book QA poses, as well as advances the published state-of-the-art with a ∼7% absolute improvement on Rouge-L. (2) We further analyze the detailed challenges in Book QA through human studies.[1] Our findings indicate that the event-centric questions dominate this task, which exemplifies the inability of existing QA models to handle event-oriented scenarios.

1 Introduction

Recent Question-Answering (QA) models have achieved or even surpassed human performance on many challenging tasks, including single-passage QA[2] and open-domain QA (ODQA)[3]. Nevertheless, understanding rich context beyond text pattern matching remains unsolved, especially answering questions on narrative elements via reading books. One example is NarrativeQA (Kočiskỳ et al., 2018) (Fig. 1). Since its first release in 2017, there has been no significant improvement over the primitive baselines. In this paper, we study this challenging Book QA task and shed light on the inherent difficulties.



First Act

SCENE ONE
[The octagon room at Sir Robert Chiltern's house in Grosvenor Square]

Mrs. Marchmont: Going on to the Hartlocks' tonight, Margaret?
Lady Basildon: I suppose so. Are you?
Mrs. Marchmont: Yes. Horribly tedious parties they give, don't they?
Lady Basildon: Horribly tedious! Never know why I go. ...
Mrs. Marchmont: I come here to be educated.
......

Question: Why did Marchmont feel upset at the beginning?
Answer: Because of the tedious parties.

Figure 1: An example of Book QA. The content is from the book *An Ideal Husband* (Wilde and Fornelli, 1916). The bottom contains a typical QA pair, and the highlighted text is the evidence for deriving the answer.

Despite its similarity to standard ODQA tasks[4], i.e., both requiring finding evidence paragraphs for inferring answers, the Book QA has certain unique *challenges* (Kočiskỳ et al., 2018): *(1)* the narrative writing style of book stories differs from the formal texts in Wikipedia and news, which demands a deeper understanding capability. The flexible writing styles from different genres and authors make the challenge severe; *(2)* the passages that depict related book plots and characters share more semantic similarities than the Wikipedia articles, which increases confusion in finding the correct

* Equal contribution. XM built the whole system, implemented the data preprocessing pipeline, Hard EM ranker, and all the reader modules, and conducted all the QA experiments. CY implemented the unsupervised ICT ranker and the first working version of FiD, and was responsible for the final ranker module. MY is the corresponding author, who proposed and led this project, built the ranker code base (until the DS ranker), designed the question schema and conducted its related experiments and analysis in Part II.

[1] https://github.com/gorov/BookQA.

[2] The SQuAD leaderboard (Rajpurkar et al., 2018): rajpurkar.github.io/SQuAD-explorer.

[3] Wang et al. (2020); Iyer et al. (2020)'s results on Quasar-T (Dhingra et al., 2017) and SearchQA (Dunn et al., 2017).

[4] Historically, open-domain QA meant "QA on any domain/topic". More recently, the term has been restricted to "retrieval on a large pile of corpus" (Chen et al., 2017), so "open-retrieval QA" seems a better term here. However, to follow the recent terminology in the QA community, we still use "open-domain QA" throughout this paper.

evidence to answer a question; *(3)* the free-form nature of the answers necessitates the summarization ability from the narrative plots; *(4)* the free-form answers make it hard to obtain fine-grained supervision at passage or span levels; and finally *(5)* different paragraphs usually have logical relations among them.[5]

To quantify the aforementioned challenges, we conduct a two-fold analysis to examine the gaps between Book QA and the standard ODQA tasks. First, we benchmark the Book QA performance on the NarrativeQA dataset, with methods created or adapted based on the ideas of state-of-the-art ODQA methods (Wang et al., 2018a; Lin et al., 2018; Lee et al., 2019; Min et al., 2019; Guu et al., 2020; Karpukhin et al., 2020). We build a state-of-the-art Book QA system with a *retrieve-and-read* framework, which consists of a **ranker** for retrieving evidence and a **reader** (i.e., QA model) to predict answers given evidence. For the ranker model, we investigate different weakly supervised or unsupervised methods for model training with the lack of passage-level supervision. For the reader model, we fill up the missing study and comparison among pre-trained generative models for Book QA, such as GPT-2 (Radford et al., 2019) and BART (Lewis et al., 2019). Then we investigate approaches to adapt to the book writing styles and to make use of more evidence paragraphs. As a result, our study gives a ∼7% absolute Rouge-L improvement over the published state-of-the-art.

Second, we conduct human studies to quantify the challenges in Book QA. To this end, we design a new question categorization schema based on the types of reading comprehension or reasoning skills required to provide the correct answers. Precisely, we first define the basic **semantic units**, such as entities, event structures in the questions and answers. The question category thus *determines the types of units and the relations between the units*. We annotate $1,000$ questions accordingly and discover the significantly distinctive statistics of the NarrativeQA dataset from the other QA datasets, mainly regarding the focus of event arguments and relations between events. We further give performance decomposition of our system over the question categories, to show the detailed types of challenges in a quantitative way.

In summary, our comprehensive study not only improves the state-of-the-art with careful utilization of recent ODQA advancements, but also reveals the unique challenges in Book QA with quantitative measurements.

2 Related Work

Open-Domain QA ODQA aims at answering questions from large open-domain corpora (e.g., Wikipedia). The recent work naturally adopts a ranker-reader framework (Chen et al., 2017). Recent success in this field mainly comes from improvement in the following directions: (1) distantly supervised training of neural ranker models (Wang et al., 2018a; Lin et al., 2018; Min et al., 2019; Cheng et al., 2020) to select relevant evidence passages for a question; (2) fine-tuning and improving the pre-trained LMs, like ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), as the rankers and readers; (3) unsupervised adaptation of pre-trained LMs to the target QA tasks (Lee et al., 2019; Sun et al., 2019; Xiong et al., 2019a).

Book QA Previous works (Kočiskỳ et al., 2018; Tay et al., 2019; Frermann, 2019) also adopt a ranker-reader pipeline. However, they have not fully investigated the state-of-the-art ODQA techniques. First, the NarrativeQA is a generative QA task by nature, yet the application of the latest pre-trained LMs for generation purposes, such as BART, is not well-studied. Second, lack of fine-grained supervision on evidence prevents earlier methods from training a neural ranking model, thus they only use simple BM25 (Robertson et al., 1995) based retrievers. An exception is (Mou et al., 2020) that constructs pseudo distance supervision signals for ranker training. Another relevant work (Frermann, 2019) uses book summaries as an additional resource to train rankers. However, this is different from the aim of Book QA task in answering questions solely from books, since in a general scenario the book summary cannot answer all questions about the book. Our work is the first to investigate and compare improved training algorithms for rankers and readers in Book QA.

3 Task Setup

3.1 Task Definition and Dataset

Following Kočiskỳ et al. (2018), we define the **Book QA** task as finding the answer $\mathbf{A}$ to a question $\mathbf{Q}$ from a book, where each book contains a number of consecutive and logically-related para-

[5] We consider Challenge (5) more like an opportunity than challenges, and leave its investigation to future work.

graphs $\mathcal{C}$. The size $|\mathcal{C}|$ from different books varies from a few hundred to thousands.

All our experiments are conducted on the NarrativeQA dataset (Kočiskỳ et al., 2018). It has a collection of 783 books and 789 movie scripts (we use books to refer to both of them), each containing an average of 62K words. Besides, each book has 30 question-answer pairs generated by human annotators in free-form natural language. Hence the exact answers are not guaranteed to appear in the books. NarrativeQA provides two different settings, the **summary** setting and the **full-story** setting. The former requires answering questions from book summaries from Wikipedia, and the latter requires answering questions from the original books, assuming that the summaries do not exist. Our Book QA task corresponds to the full-story setting, and we use both names interchangeably.

Following Kočiskỳ et al. (2018), we tokenize the books with SpaCy[6], and split each book into non-overlapping trunks of 200 tokens.

3.2 Baseline

Following the formulation of the open-domain setting, we employ the dominating ranker-reader pipeline that first utilizes a *ranker model* to select the most relevant passages $\mathcal{C}_{\mathbf{Q}}$ to $\mathbf{Q}$ as evidence,

$$\mathcal{C}_{\mathbf{Q}} = \text{top-k}(\{P(\mathbf{C}_i|\mathbf{Q})|\forall\, \mathbf{C}_i \in \mathcal{C}\}); \qquad (1)$$

and then a *reader model* to predict answer $\tilde{\mathbf{A}}$ given $\mathbf{Q}$ and $\mathcal{C}_{\mathbf{Q}}$.

Our baseline QA systems consist of training different base reader models (detailed in Sec. 4.1) over the BM25 ranker. We also compare with competitive public Book QA systems as baselines from (Kočiskỳ et al., 2018; Frermann, 2019; Tay et al., 2019; Frermann, 2019; Mou et al., 2020) under the Narrative full-story setting, and a concurrent work (Zemlyanskiy et al., 2021). As discussed in Section 2, Mou et al. (2020) train a ranker with distant supervision (DS), i.e., the first analyzed ranker method (Fig. 3); Fremann (2019) use exterior supervision from the book summaries, which is considered unavailable by design of the Book QA task. Because the summaries are written by humans, the system can be viewed as benefiting from human comprehension of books. Fig. 2 lists the details of our compared systems.

System	trained ranker	pre-trained LM	extra data
IR+AttSum† (Kočiskỳ et al., 2018)			
IR+BiDAF‡ (Kočiskỳ et al., 2018)			
IAL-CPG† (Tay et al., 2019)			
R^{3‡} (Wang et al., 2018a)	✓		
BERT-heur‡ (Frermann, 2019)	✓	✓	✓
DS Ranker+GPT2† (Mou et al., 2020)	✓	✓	
DS Ranker+BERT‡ (Mou et al., 2020)	✓	✓	
Our best QA system†	✓	✓	

Figure 2: Characteristics of the compared systems. †/‡ refers to generative/extractive QA systems. In addition to the standard techniques, Wang et al. (2018a) use reinforcement learning to train the ranker; Tay et al. (2019) use curriculum to train the reader.

3.3 Metrics

Following previous works (Kočiskỳ et al., 2018; Tay et al., 2019; Frermann, 2019), we use Rouge-L (Lin, 2004) as the main metric for both evidence retrieval and question answering.[7] For completeness, Appendix A provides results with other metrics used in the previous works, including Bleu-1/4 (Papineni et al., 2002), Meteor (Banerjee and Lavie, 2005), and the Exact Match (EM) and F1 scores that are commonly used in extractive QA.

4 Analysis Part I: Experimental Study

This section describes our efforts of applying or adapting the latest open-domain QA ideas to improve Book QA ranker/reader models. Fig. 3 summarizes our inspected approaches. The experimental results quantify the challenges in Book QA beyond open-domain QA.

4.1 QA Reader

Base Reader Models We study the usage of different pre-trained LMs on Book QA, including BART (Lewis et al., 2019), GPT-2 (Radford et al., 2019), T5 (Raffel et al., 2019) and BERT (Devlin et al., 2019). The first three are *generative* readers and can be directly trained with the free-form answers as supervision. Specifically, during training we treat $\mathbf{Q} \oplus [\text{SEP}] \oplus \mathcal{C}_{\mathbf{Q}}$ as input to generate answer $\mathbf{A}$, where [SEP] is the special separation token and $\oplus$ is the concatenation operator.

For the *extractive* reader (BERT), we predict the most likely span in $\mathcal{C}_{\mathbf{Q}}$ given the concatenation of the question and the evidence $\mathbf{Q} \oplus [\text{SEP}] \oplus \mathcal{C}_{\mathbf{Q}}$. Due to the generative nature of Book QA, the true

[6] https://spacy.io/

[7] For fair comparison, we lowercase the answers and remove the punctuation, and use the open-source nlg-eval library (Sharma et al., 2017).

Target Module	Approach	Original Idea in ODQA	Our Improved Version for Book QA
Reader	Book Prereading	Sun et al. (2019) adapt GPT on general QA datasets; Xiong et al. (2019a) adapt BERT on Wikipedia as entity prediction.	We propose to adapt BART to the narrative style with the text-infilling objective.
	Fusion-in-Decoder	Proposed by (Izacard and Grave, 2020) as a new type of ODQA reader.	We improve the decoder with attention over all the encoder states to capture cross-passage interaction.
Ranker	Heuristic distant supervision	N/A	N/A*
	Unsupervised ICT	Proposed by (Lee et al., 2019) as siamese network for both BERT pre-training and dense retrieval.	We improve the method with our book-specific training data selection.
	Hard EM	Proposed by (Min et al., 2019) for reader training.	We adapt the method for ranker training.

Figure 3: Summary of our inspected approaches in Analysis Part I. *We directly apply the heuristics from (Mou et al., 2020) for Book QA.

answer may not have an exact match in the context. Therefore, we follow Mou et al. (2020) to find the span $\mathbf{S}$ that has the maximum Rouge-L score with the ground truth $\mathbf{A}$ as the weak label, subject to that $\mathbf{A}$ and $\mathbf{S}$ have the same length (i.e. $|\mathbf{S}| = |\mathbf{A}|$).

Method 1: Book Prereading Inspired by the literature on the unsupervised adaptation of pretrained LMs (Sun et al., 2019; Xiong et al., 2019a), we let the reader "**preread**" the training books through an additional pre-training step prior to fine-tuning with QA task. This technique helps to better adapt to the narrative writing styles.

Specifically, we extract random passages from all training books to build a passage pool. For each training iteration, we mask random spans from each passage, following the setting in (Lewis et al., 2019). The start positions of spans are sampled from a uniform distribution without overlapping. The length of each span is drawn from a Poisson distribution with $\lambda = 3$. Each span is then replaced by a single `[mask]` token regardless of the span length. We mask 15% of the total tokens in each passage. During the prereading stage, we use the masked passage as the encoder input and the raw passage as the decoder output to restore the raw passage in the auto-regressive way.

Method 2: Fusion-in-Decoder Recently Izacard and Grave (2020) scale BART reader up to large number of input paragraphs. The method, **Fusion-in-Decoder (FiD)**, first concatenates each paragraph to the question to obtain a question-aware encoded vector, then merges these vectors from all paragraphs and feeds them to a decoder for answer prediction. FiD reduces the memory and time costs for encoding the concatenation of all paragraphs, and improves on multiple ODQA datasets. FiD is an interesting alternative for Book QA, since it can be viewed as an integration of the ranker and reader, with the ranker absorbed in the separated paragraph encoding step.

FiD trades cross-paragraph interactions for encoding more paragraphs. The single encoded vector per passage works well for extractive ODQA because the vector only needs to encode information of candidate answers. However, in Book QA, the answers may not be inferred from a single paragraph and integration of multiple paragraphs is necessary. Therefore, in our approach, we concatenate the encoded vectors of all the paragraphs, and rely on the decoder's attention over these vectors to capture the cross-paragraph interactions.

4.2 Passage Ranker

Base Ranker Model Our ranker is a BERT-based binary classifier fine-tuned for evidence retrieval. It estimates the likelihood of each passage to be supporting evidence given a question $\mathbf{Q}$.

Training the ranker models is difficult without high-quality supervision. To deal with this problem, we investigate three approaches for creating pseudo labels, including *distant supervision, unsupervised ranker training and Hard EM training*.

Method 1: Distant Supervision (DS) This is the baseline approach from (Mou et al., 2020). It constructs DS signals for rankers in two steps: First, for each question $\mathbf{Q}$, two BM25 rankers are used to retrieve passages, one with $\mathbf{Q}$ as query and the other with both $\mathbf{Q}$ and the true answer $\mathbf{A}$. Denoting the corresponding retrieval results as $\mathcal{C}_{\mathbf{Q}}$[8] and $\mathcal{C}_{\mathbf{Q}+\mathbf{A}}$, the method samples the positive samples $\mathcal{C}^{+}_{\mathbf{Q}}$ from $\mathcal{C}_{\mathbf{Q}} \cap \mathcal{C}_{\mathbf{Q}+\mathbf{A}}$ and the negative samples $\mathcal{C}^{-}_{\mathbf{Q}}$ from the rest, with the ratio $\sigma \equiv |\mathcal{C}^{+}_{\mathbf{Q}}|/|\mathcal{C}^{-}_{\mathbf{Q}}|$ for each question $\mathbf{Q}$ as a hyperparameter.

Second, to enlarge the margin between the positive and negative samples, the method applies a

[8] For simplicity, we use the notation $\mathcal{C}_{\mathbf{Q}}$ here.

Rouge-L filter upon the previous sampling results to get the refined samples, $\mathcal{C}_{\mathbf{Q}}^{++}$ and $\mathcal{C}_{\mathbf{Q}}^{--}$:

$$\mathcal{C}_{\mathbf{Q}}^{++} = \left\{ \max_{\mathbf{S}\subset\mathbf{C}_i, |\mathbf{S}|=|\mathbf{A}|} Sim(\mathbf{S},\mathbf{A}) > \alpha, \mathbf{C}_i \in \mathcal{C}_{\mathbf{Q}}^{+} \right\}$$

$$\mathcal{C}_{\mathbf{Q}}^{--} = \left\{ \max_{\mathbf{S}\subset\mathbf{C}_i, |\mathbf{S}|=|\mathbf{A}|} Sim(\mathbf{S},\mathbf{A}) < \beta, \mathbf{C}_i \in \mathcal{C}_{\mathbf{Q}}^{-} \right\}.$$

$\mathbf{S}$ is a span in $\mathbf{C}_i$, $Sim(\cdot,\cdot)$ is Rouge-L between two sequences. α and β are hyperparameters.

Method 2: Unsupervised ICT Training Inspired by the effectiveness of Inverse Cloze Task (ICT) (Lee et al., 2019) as an unsupervised ranker training objective, we use it to pre-train our ranker. The rationale is that we construct "pseudo-question" q and "pseudo-evidence" b from the same original passage p and aim at maximizing the probability $P_{\text{ICT}}(b|q)$ of retrieving b given q, which is estimated using negative sampling as:

$$P_{\text{ICT}}(b|q) = \frac{\exp\left(S_{retr}(b,q)\right)}{\sum_{b'\in B}\exp\left(S_{retr}\left(b',q\right)\right)}. \quad (2)$$

$S_{retr}(\cdot,q)$ is the relevance score between a paragraph and the "pseudo-question" q. $b'{\neq}b$ is sampled from original passages other than p.

The selection of "pseudo-questions" is critical to ICT training. To select representative questions, we investigate several filtering methods, and finally develop a book-specific filter [9]. Our method selects the top-scored sentence in a passage as a "pseudo-question" in terms of its total of token-wise mutual information against the corresponding book. The details can be found in Appendix B.

Method 3: Hard EM Hard EM is an iterative learning scheme. It was first introduced to ODQA by Min et al. (2019), to find correct answer spans that maximize the reader performance. Here we adapt the algorithm to ranker training. Specifically, the hard EM can be achieved in two steps. At step t, the E-step first trains the reader with the current top-k selections ${\mathcal{C}_Q}^t$ as input to update its parameters Φ^{t+1}; then derives the new positive passages ${\mathcal{C}_Q^+}^{t+1}$ that maximizes the reader Φ^{t+1}'s probability of predicting $\mathbf{A}$ (Eq. 3). The M-step updates the ranker parameter Θ (Eq. 4):

$${\mathcal{C}_Q^+}^{t+1} = k\text{-}\max_{\mathbf{C}_i\in\mathcal{C}} P(\mathbf{A}|\mathbf{C}_i,\Phi^{t+1}) \quad (3)$$

$$\Theta^{t+1} = \arg\max_{\Theta} P({\mathcal{C}_Q^+}^{t+1}|\Theta^t). \quad (4)$$

In practice, Min et al. (2019) find that initialized with standard maximum likelihood training, the Hard EM usually converges in 1-2 EM iterations.

System	Rouge-L dev	Rouge-L test
Public Extractive Baselines		
BiDAF (Kočiskỳ et al., 2018)	6.33	6.22
R^3 (Wang et al., 2018a)	11.40	11.90
DS-ranker + BERT (Mou et al., 2020)	14.76	15.49
BERT-heur (Fremann, 2019)	–	15.15
ReadTwice (Zemlyanskiy et al., 2021)	22.7	23.3
Public Generative Baselines		
Seq2Seq (Kočiskỳ et al., 2018)	13.29	13.15
AttSum* (Kočiskỳ et al., 2018)	14.86	14.02
IAL-CPG (Tay et al., 2019)	17.33	17.67
DS-Ranker + GPT2 (Mou et al., 2020)	21.89	22.36
Our Book QA Systems		
BART-no-context (baseline)	16.86	16.83
BM25 + BART reader (baseline)	23.16	24.47
Our best ranker + BART reader	25.83	26.95†
Our best ranker + our best reader	**27.91**	**29.21**†
repl ranker with oracle IR	*37.75*	*39.32*

Table 1: Overall QA performance (%) in NarrativeQA Book QA setting. *Oracle IR* combines question and true answers for BM25 retrieval. We use an asterisk (*) to indicate the best results reported in (Kočiskỳ et al., 2018) with multiple hyper-parameters on dev set. The dagger (†) indicates significance with p-value < 0.01.

5 Evaluation Part I: QA System Ablation

We evaluate the overall Book QA system, and the individual modules on NarrativeQA.

Implementation Details: For rankers, we initialize with *bert-base-uncased*. For readers, we use *bert-base-uncased*, *gpt2-medium*, *bart-large* and *T5-base*. The readers use top-3 retrieved passages as inputs, except for the FiD reader which uses top-10, making the readers have comparable time and space complexities.

5.1 Overall Performance of Book QA

We first show the positions of our whole systems on the NarrativeQA Book QA task. Table 1 lists our results along with the state-of-the-art results reported in prior works (see Section 3.2 and Fig. 2 for reference). Empirically, our best ranker is from the combination of heuristic distant supervision and the unsupervised ICT training; our best reader is from the combination of the FiD model plus book prereading (with the top-10 ranked paragraphs as inputs). It is observed that specifically designed pre-training techniques play the most important role. Details of the best ranker and reader can be found in the ablation study.

[9] A unique filter is built for each book.

Overall, we significantly raise the bar on NarrativeQA by 4.7% over our best baseline and 6.8% over the best published one.[10] But there is still massive room for future improvement, compared to the upperbound with oracle ranker. Our baseline is better than all published results with simple BM25 retrieval, showing the importance of reader investigation. Our best ranker (see Section 5.2 for details) contributes to 2.5% of our improvement over the baseline. Our best reader (see Section 5.3 for details) brings an additional $>2\%$ improvement compared to the BART reader.

We conduct a significance test for the results of our best system. There is no agreement on the best practice of the tests for natural language generation (Clark et al., 2011; Dodge et al., 2019). We choose the non-parametric bootstrap test, because it is a more general approach and does not assume specific distributions over the samples. For bootstrapping, we sample 10K subsets, the size of each is 1K. The small p-value (< 0.01) shows the effectiveness of our best model.

As a final note, even the results with oracle IR are far from perfect. It indicates the limitation of text-matching-based IR; and further confirms the challenge of evidence retrieval in Book QA.

5.2 Ranker Ablation

To dive deeper into the effects of our ranker training techniques in Sec. 4.2, we study the intermediate retrieval results and measure their coverage of the answers. The coverage is estimated on the top-5 selections of a ranker from the baseline BM25's top-32 outputs, by both the maximum Rouge-L score of all the overlapped subsequences of the same length as the answer in the retrieved passages; and a binary indicator of the appearance of the answer in the passages (EM). Table 2 gives the ranker-only ablation. On one hand, our best ranker improves both metrics. It also significantly boosts the BART reader compared to the DS-ranker (Mou et al., 2020), as shown in Appendix A. On the other hand, on top of the DS ranker, none of the other techniques can further improve the two ranker metrics significantly. The ICT unsupervised training brings significant improvement over BM25. When adding to the DS-ranker, it brings slight improvement and leads to our best results. Hard EM (Min et al., 2019) does not lead to improvements. Our conjecture is that generative readers does not solely generate purely matching-oriented signals, thus introduces noise in matching-oriented ranker training.

IR Method	EM	Rouge-L
Baseline Rankers		
BM25	18.99	47.48
BERT DS-ranker (Mou et al., 2020)	24.26	52.68
- Rouge-L filtering	22.63	51.02
Repl BERT w/ BiDAF	21.88	50.64
Repl BERT w/ MatchLSTM	21.97	50.39
Our Rankers		
BERT ICT-ranker	21.29	50.35
BERT DS-ranker		
+ Hard EM	22.45	50.50
+ ICT pre-training*	**24.83**	**53.19**
Oracle Conditions		
Upperbound (BM25 top-32)	30.81	61.40
Oracle (BM25 w/ Q+A)	35.75	63.92

Table 2: Ranker performance (top-5) on dev set. Asterisk (*) indicates our best ranker used in Table 1.

The limited improvement and the low absolute performance demonstrate the difficulty of retrieval in Book QA. The gap between our best performance and the upper-bound implies that there is a large potential to design a more advanced ranker.

Additionally, we show that how much useful information our best ranker can provide to our readers in the whole QA system. In our implementation, the BART and FiD readers use top-3 and top-10 paragraphs from the ranker respectively. The top-3 paragraphs from our best ranker give the answer coverage of 22.12% EM and 49.83% Rouge-L; and the top-10 paragraphs give 27.15% EM and 56.77% Rouge-L. In comparison, the BM25 baseline has 15.75%/43.44% for top-3 and 24.08%/53.55% for top-10. Therefore, our best ranker efficiently eases the limited-passage bottleneck brought by the ranker and benefits BART reader much more, which is consistent with our observations in Table 3, Section 5.3.

5.3 Reader Ablation

Table 3 shows how the different reader techniques in Section 4.1 contribute to the QA performance.

First, switching the BART reader to FiD gives a large improvement when using the BM25 ranker (2.8%), approaching the result of "our ranker + BART". This agrees with our hypothesis in Section 4.1 Analysis 2, that FiD takes the roles of both

[10] Appendix A reports the full results, where we achieve the best performance across all of the metrics.

System	Rouge-L dev	Rouge-L test
BM25 + BART reader (baseline)	23.16	24.47
+ BART-FiD reader	25.95	–
Our ranker + BART reader	25.83	26.95
+ BART-FiD reader	26.27	–
repl BART w/ GPT-2	22.22	–
repl BART w/ T5	20.57	–
+ book preread	26.82	–
+ BART-FiD Reader*	**27.91**	**29.21**
+ book preread (*decoder-only*)	26.51	–

Table 3: Ablation of our Reader Model. Asterisk (*) indicates our best reader used in Table 1.

ranker and reader. Second, although the above result shows that FiD's ranking ability does not add much to our best ranker, our cross-paragraph attention enhancement still improves FiD due to better retrieval results (0.5% improvement over "our ranker + BART"). Third, among all the generative reader models, BART outperforms GPT-2 and T5 by a notable margin. Finally, the book prereading brings consistent improvements to both combinations; and the combination of our orthogonal reader improvements finally gives the best results. We also confirm that the prereading helps decoders mostly, as only training the decoder gives comparable results.

6 Analysis Part II: Human Study

This section conducts in-depth analyses of the challenges in Book QA. We propose a *new question categorization scheme* based on the types of comprehension or reasoning skills required for answering the questions; then conduct a human study on 1,000 questions. Consequently, the model performance per category provides further insights of the deficiency in current QA models.

6.1 Question Categorization

There have been many different question categorization schemes. Among them the most widely-used is intention-based, where an intention is defined by the *WH*-word and its following word. Some recent reasoning-focused datasets (Yang et al., 2018; Xiong et al., 2019b) categorize intents by the types of multi-hop reasoning or by the types of required external knowledge beyond texts.

However, all these previous schemes do not reasonably fit our analysis over narrative texts from two aspects: (1) they only differentiate high-level reasoning types, which is useful in knowledge base QA (i.e., KB-QA) but fails to pinpoint the text-based evidence in Book QA; (2) they are usually entity-centric and overlook linguistic structures like events, while events play essential roles in narrative stories. With this, we design a new systematic schema to categorize the questions in the NarrativeQA dataset.

Semantic Unit Definition We first identify a minimum set of basic semantic units, each describing one of the most fundamental components of a story. The set should be sufficient such that (1) each answer can be uniquely linked to one semantic unit, and (2) each question should contain at least one semantic unit. Our final set contains three main classes and nine subclasses (Fig. 4).

We merge the two commonly-used types in the previous analysis, named entities and noun phrases, into the *Concept* class. The *Event* class follows the definition in ACE 2005 (Walker et al., 2006). We also use a special sub-type "*Book Attribute*" that represents the meta information or the global settings of the book, such as the era and the theme of the story in a book.

Question Type Definition On top of the semantic units' definition, each question can be categorized as a query that asks about either a semantic unit or a relation between two semantic units. We use the difference and split all the questions into nine types grouped in four collections (Fig. 5).

- **Concept questions** that ask a *Concept attribute* or a relation between two *Concepts*. The most common types in most ODQA tasks (e.g., TriviaQA) and the QA tasks require multi-hop reasoning (e.g., ComplexQuestions and HotpotQA).
- **Event-argument questions** that ask parts of an event structure. This type is less common in the existing QA datasets, although some of them contain a small portion of questions in this class. The large ratio of these event-centric questions demonstrates the uniqueness of the NarrativeQA dataset.
- **Event-relation questions** that ask relations (e.g., causal or temporal relations) between two events or between an event and an attribute (a state or a description). This type is common in NarrativeQA, since events play essential roles in story narrations. A particular type in this group is the relation that one event serves as the argument of another event (e.g., *how*-questions). It corresponds to the common linguistic phenomenon of (compositional) nested event structures.

SU Type	Sub Type	Description	Example
Concept	Entity	Standard named entities like person, location and organization names. Book-specific character names and their co-references are also included.	**Q**: What is the name of Mortimer Treginnis' sister? **A**: Brenda
	Common Noun-Phrases	Common nouns or noun phrases that are universally used across books and other literature	**Q**: What was Rodgers exposed to while investigating? **A**: Radioactive gas
	Book-Specific	Common nouns or noun phrases that have special meanings or importance in the book of interests	**Q**: Where do Anne and Philippa stay after their first year in college? **A**: Patty's place
Event	Event Expression	Standard textual expression of event structures about "*who did what to whom, when, where and how*"	**Q**: In what way did Christopher atone for his sin? **A**: He helped Will escape and accepted the punishment
	Event Name	Sometimes an important or famous event will be referred with a name	**Q**: When did Harney and Charity kiss for the first time? **A**: On the trip to Nettleton
Attribute	States	The textual description of the state of an entity or concept as an attribute	**Q**: Why does the princess agree to let Ermyntrude pretend to be her? **A**: Because she is timid
	Numerics	Standard attributes with numeric values	**Q**: How may volumes has Darnley written on the origins of life? **A**: Three
	Descriptions	An attribute of an entity or concept which does not have a short attribute phrase to summarize	**Q**: Why didn't Anne accept Gilberts proposal? **A**: She's dreaming of true love
	Book Attributes	Attributes of books themselves, like theme etc.	**Q**: Name the major theme used in the Adventures of Sherlock Holmes? **A**: Social injustice

Figure 4: The definitions of semantic units (**SU**s). The underlined texts represent the recognized SUs of the types.

Question Type	Description	Example
Relation between Concepts	The question asks a relation a concept has, and expects another concept as the answer	**Q**: What is the name of Mortimer Treginnis' sister? **A**: Brenda
Attribute of Concept	The question asks the value of an attribute a concept has, and expects an attribute value as the answer	**Q**: How old is Conan? **A**: Around forty
Event Argument - Concept	The question asks an argument of an event, and expects the argument to be an concept	**Q**: Where was Armitage discovered alive? **A**: Italy
Event Argument - Attribute	Similar to the above, but asks for an argument that is an attribute	**Q**: Where does Lady Dedlock believe Esther to be when the story starts? **A**: She believes her to be dead
Event Trigger	A rare case whether the question asks the action, i.e., the trigger (like main verb) of an event	**Q**: What does Conan do to the Pictish village? **A**: He sets it on fire
Causal Relation	The question asks the cause of an event, and expects another event or a concept attribute as the answer	**Q**: Why is Barabas angry at the Maltese governor? **A**: He robbed him
Temporal Relation	The question asks an event that has a type of temporal relations with another event or an attribute	**Q**: What was Almayer doing when Mrs. Almayer snuck Nina away? **A**: Drinking with the Dutch
Nested Relation	The question asks an argument of an event, while the argument's value is another event or an attribute	**Q**: What did Dain vow to come back and help Almayer with? **A**: Finding the gold mine
Book Attribute	The question asks an attribute of the book itself	**Q**: Where did the majority of the story occur ? **A**: London

Figure 5: The definitions of question types. Note that sometimes the answer repeats parts of the question like the last two examples in the second block, and we ignore these parts when recognizing the SUs in answers.

- **Global-attribute questions** that ask *Book Attribute*: As designed, it is also unique in Book QA.

6.2 Annotation Details

Five annotators are asked to label the semantic unit types and the question types on a total of 1,000 question-answer pairs. There can be overlapped question categories for the same question. A major kind of overlaps is between the three event component types (trigger, argument - concept/attribute) and the three event relation types (causal, temporal and nested). Therefore in the guideline, when the question can be answered with an event component, we ask the annotators to check if the question requires the understanding of event relations. If so, the question should be labeled with the event relation types as these are the more critical information for finding the answers. Similarly, for the other rare cases of category overlaps, we ask the annotators to label the types that they believe are more important for finding the answers.

Correlation between question and answer types Figure 6 shows the ratios of answer types under

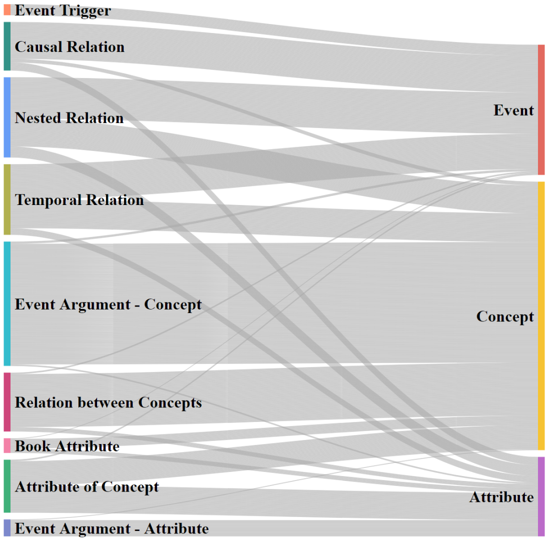


Figure 6: Visualization of the flow from the question types to their expected answer types.

Category	Simple Agreement(%)	κ(%)
Question Type	88.0	89.9
SU Type	92.3	91.2
SU Sub Type	81.3	82.8

Table 4: Annotation agreement. **SU**: Semantic Unit. “SU Type” and “SU Sub Type” are defined in Figure 4.

each question type via a flow diagram. Most question types correspond to a single major answer type, with a few exceptions: (1) Most of the three event-relation questions have events as answers. A small portion of them have concepts or attributes as answers. This is either because the answers are state/description attributes; or because the answers are the arguments of one of the related events queried by the questions. (2) The *Relation b/w Concepts* type has some questions with attribute-typed answers. This is because the questions may ask the names of relations themselves, while some relation names are recognized as description-typed attributes. (3) Most of *Book Attribute* questions have concepts as answers, because they ask for the protagonists or the locations the stories occur at.

Annotation agreement A subset of 150 questions is used for quality checking, with each question labeled by two annotators. Table 4 reports both the simple agreement rates and the Fleiss' Kappa (Fleiss, 1971) κs. Our annotations reach a high agreement with around 90% for question types and SU types and 80% for SU sub-types, reflecting the rationality of our scheme.

6.3 Performance of Question Type Classification on the Annotated Data

We conduct an additional experiment to study how well a machine learning model can learn to classify our question types based on question surface patterns. We use the RoBERTa-base model that demonstrates superior on multiple sentence classification tasks. Since our labeled data is small, we conduct a 10-fold cross validation on our labeled 1,000 instances. For each testing fold, we randomly select another fold as the development set and use the rest folds as training.

The final averaged testing accuracy is 70.2%. Considering the inter agreement rate of 88.0%, this is a reasonable performance, with several reasons for the gap: (1) Our training data is too small and easy to overfit, evidenced by the performance gap between the training accuracy and development accuracy (∼100% versus 73.4%). The accuracy can be potentially increased with more training data. (2) Some of the ambiguous questions require the contexts to determine their types. During labeling, our human annotators are allowed to read the answers for additional information, which leads to a higher upperbound performance. (3) There is a small number of ambiguous cases, on which humans can use world knowledge while models are difficult to employ such knowledge. Therefore, the current accuracy can be potentially increased with a better model architecture.

Error Analysis and Lessons Learned Figure 7 gives major error types, which verifies the our discussed reasons above. The majority of errors are the confusion between *Event Argument - Concept* and *Nested Relation*. The models are not accurate on the two types for several reasons: (1) Sometimes the similar question surface forms can take both concepts and events as an argument. In these cases, the answers are necessary for determining the question type. (2) According to our annotation guideline, we encourage the annotators to label event relations with higher priority, especially when the answer is a concept but serves as an argument of a clause. This increases the labeling error rate between the two types. Another major error type is labeling *Causal Relation* as *Nest Relation*. This is mainly because some questions ask causal relations in an implicit way, on which human annotators have the commonsense to identify the causality but models do not. The third major type is the failures in identifying the *Attribute*

Groundtruth Type	Predicted Type	Freq
Relation between Concepts	$\xrightarrow{\text{Fail}}$ Attribute of Concept	17/110
	$\xrightarrow{\text{Fail}}$ Event Argument - Concept	12/110
Attribute of Concept	$\xrightarrow{\text{Fail}}$ Relation between Concepts	21/120
	$\xrightarrow{\text{Fail}}$ Event Argument - Concept	15/120
Event Argument - Attribute	$\xrightarrow{\text{Fail}}$ Event Argument - Concept	6/34
	$\xrightarrow{\text{Fail}}$ Attribute of Concept	6/34
	$\xrightarrow{\text{Fail}}$ Temporal Relation	4/34
Event Argument - Concept	$\xrightarrow{\text{Fail}}$ Nested Relation	34/283
Event Trigger	$\xrightarrow{\text{Fail}}$ Nested Relation	4/18
	$\xrightarrow{\text{Fail}}$ Event Argument - Concept	4/18
Causal Relation	$\xrightarrow{\text{Fail}}$ Nested Relation	17/126
Nested Relation	$\xrightarrow{\text{Fail}}$ Event Argument - Concept	35/154
Book Attribute	$\xrightarrow{\text{Fail}}$ Attribute of Concept	3/29

Figure 7: Error analysis of question-type classification. We only list the major errors of each type (i.e., incorrect predicted types that lead to >10% of the errors).

Question Type	Ratio(%)	QA Rouge-L Gen	QA Rouge-L Ext	Ranker Rouge-L
Relation b/w Concepts	11.0	40.48	24.46	63.76
Attribute of Concept	12.0	34.09	21.69	56.73
Event - Attribute	3.4	25.88	10.57	49.23
Event - Concept	28.3	27.35	15.73	62.15
Event - Trigger	1.8	29.63	9.28	37.56
Causal Relation	12.6	22.86	10.39	38.47
Temporal Relation	12.6	28.01	15.57	49.20
Nested Relation	15.4	23.02	8.44	48.93
Book Attribute	2.9	23.11	25.71	54.60

Table 5: Performance decomposition to question types of our best generative system (**Gen**, the best BART-based system), extractive system (**Ext**, the best BERT-based system, i.e., our best ranker + BERT reader), and ranker (BERT+ICT from Table 2).

of Concept and the *Relation b/w Concepts* categories. As the attributes can be associated to some predicates, especially when they are descriptions, the models confuse them with relations or events.

The above observations provide insights on future refinement of our annotation guidelines, if people want to further enlarge the labeled data. For example, the *Nested Relation* should be more clearly defined with comprehensive examples provided. In this way, the annotators can better distinguish them from the other types; and can better determine if the nested structure exists and whether to label the *Event Argument* types. Similarly, we could define clearer decision rules among relations, attributes and events, to help annotators distinguish *Relation b/w Concepts*, *Attribute of Concept* and *Event Argument - Concept* types.

Answer Type	Ratio(%)	QA Rouge-L Gen	QA Rouge-L Ext	Ranker Rouge-L
Concept - Entity	35.3	26.76	18.59	66.79
Concept - Common Noun	16.9	31.53	12.90	51.03
Concept - Book Specific	4.3	39.68	26.53	65.54
Event - Expression	25.1	24.62	11.50	39.40
Event - Name	2.8	24.79	5.54	42.88
Attribute - State	4.2	38.75	17.03	53.82
Attribute - Numeric	4.7	33.57	24.44	57.31
Attribute - Description	6.1	26.13	11.15	41.70
Attribute - Book Attribute	0.6	27.91	19.88	52.78

Table 6: Performance decomposition to answer types of our best generative/extractive systems and ranker. **Gen** and **Ext** are the same systems as in Table 5.

7 Evaluation Part II: QA System Performance Decomposition

Table 5 presents both the ratio of each question type and our best generative and extractive performance on it. The ratios reflect NarrativeQA's unique focus on events, as ∼75% of the questions are relevant to the events in book stories. Specifically, ∼34% of the questions ask components of event structures (i.e., arguments or triggers) and 41% ask relations between events (note that these questions may still require the understanding of event structures). By comparison, the two dominating types in the other QA datasets, *Concept Relation* and *Concept Attribute*, only contribute to a ratio of ∼23%. This agrees with human intuitions on the unique challenges in book understanding.

Most difficult question types: The performance breakdown shows that all three event-relation types (*Causal*, *Temporal* and *Nested*) are challenging to our QA systems. The *Causal* relation is the most difficult type with the lowest QA performance. The result confirms that the unique challenge in understanding event relations is still far from being well-handled by current machine comprehension techniques, even with powerful pretrained LMs. Moreover, these types can also be potentially improved by the idea of complementary evidence retrieval (Wang et al., 2018b; Iyer et al., 2020; Mou et al., 2021) in ODQA.

Besides the three event-relation types, the *Event - Attribute* and *Event - Triggers* are also challenging to the extractive system, because the answers are usually long textual mentions of events or states that are not extractable from the passages.

Challenging types for reader: By checking the performance gaps of the generative system and the ranker, we can tell which types are difficult mainly

System	Full Data		Event-Only	
	dev	test	dev	test
BERT+Hard EM	58.1	58.8	–	–
Masque	–	54.7	–	–
BART Reader (ours)	**66.9**	**66.9**	55.1	55.0

Table 7: Rouge-L scores under NarrativeQA summary setting. We list the best public extractive model BERT+Hard EM (Min et al., 2019) and the best generative model Masque (Nishida et al., 2019) for reference.

for the reader.[11] The *Event - Concept* type poses more challenges to the reader, given that the ranker can perform well on them but the overall QA performance is low. These questions are challenging mainly due to the current readers' difficulty in understanding the event structures, since their answers are usually extractable from texts.

Breakdown onto answer types: To better understand the challenges of non-extractable answers, we show the performance on each answer type in Table 6. The answers are mostly extractable when they are entities (including the book-specific terms and numeric values). On these types the extractive systems perform better and the two systems perform closer, compared to the other types. In contrast, the answers are less likely to be extractable from the original passages when they are events, states, and descriptions. An interesting observation is that the *Common Noun Phrases* type is also challenging for the extractive system. It indicates that these answers may not appear in the texts with the exact forms, so commonsense knowledge is required to connect their different mentions.

Quantifying the challenge of event-typed answers to the reader: Table 6 shows that the ranker performs poorly when the answers are events and descriptions. This arouses a question – whether the relatively lower QA performance is *mainly due to the ranker's deficiency*; or *due to the deficiency of both the ranker and the reader*.

To answer this question, we conduct an experiment in the summary setting of NarrativeQA, to eliminate the effects of the ranker. We create a subset of questions with event-typed answers if a question has either of its two answers containing a verb. This procedure results in a subset of 2,796 and 8,248 QA pairs in validation and test sets respectively. We train a BART Reader with all training data in the summary setting, and test on both the full evaluation data and our event-only subsets. Table 7 shows that the performance on the event-only subsets is about 12% lower. The results confirm that questions with event-typed answers are challenging for both the reader and the ranker.

8 Conclusion

We conduct a comprehensive analysis on the Book QA task, taking the representative NarrativeQA dataset as an example. Firstly, we design the Book QA techniques by borrowing the wisdom from the cutting-edge open-domain QA research and demonstrate through extensive experiments that (1) evidence retrieval in Book QA is difficult even with the state-of-the-art pre-trained LMs, due to the factors of rich writing style, recurrent book plots and characters, and the requirement of high-level story understanding; (2) our proposed approaches that adapt pre-trained LMs to books, especially the prereading technique for the reader training, are consistently helpful.

Secondly, we perform a human study and find that (1) a majority of questions in Book QA requires understanding and differentiating events and their relations; (2) the existing pre-trained LMs are deficient in extracting the inter- and intra-structures of the events in the Book QA. Such facts lead us towards the event understanding task for future improvement over the Book QA task.

Acknowledgment

This work is funded by RPI-CISL, a center in IBM's AI Horizons Network, and the Rensselaer-IBM AI Research Collaboration (RPI-AIRC).

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL 2005 Workshop*, pages 65–72.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of ACL 2017*, pages 1870–1879.

Hao Cheng, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2020. Probabilis-

[11] Note that this analysis cannot confirm which types pose challenges to the ranker. This is because for event answers that are relatively longer and generative, there is a natural disadvantage on our pseudo ranker Rouge scores.

tic assumptions matter: Improved models for distantly-supervised document-level question answering. *arXiv preprint arXiv:2005.01898*.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of ACL 2011*, pages 176–181.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

B Dhingra, K Mazaitis, and WW Cohen. 2017. Quasar: Datasets for question answering by search and reading. arxiv 2017. *arXiv preprint arXiv:1707.03904*.

Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A Smith. 2019. Show your work: Improved reporting of experimental results. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2185–2194.

Matthew Dunn, Levent Sagun, Mike Higgins, V Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. Searchqa: A new q&a dataset augmented with context from a search engine. *arXiv preprint arXiv:1704.05179*.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Lea Frermann. 2019. Extractive narrativeqa with heuristic pre-training. In *Proceedings of the 2nd MRQA Workshop*, pages 172–182.

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

Srinivasan Iyer, Sewon Min, Yashar Mehdad, and Wen-tau Yih. 2020. Reconsider: Re-ranking using span-focused cross-attention for open domain question answering. *arXiv preprint arXiv:2010.10757*.

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

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 EMNLP 2020*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *TACL*, 6:317–328.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. *Proceedings of ACL 2019*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. 2018. Denoising distantly supervised open-domain question answering. In *Proceedings of ACL 2018*, pages 1736–1745.

Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2019. A discrete hard em approach for weakly supervised question answering. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2844–2857.

Xiangyang Mou, Mo Yu, Shiyu Chang, Yufei Feng, Li Zhang, and Hui Su. 2021. Complementary evidence identification in open-domain question answering. *arXiv preprint arXiv:2103.11643*.

Xiangyang Mou, Mo Yu, Bingsheng Yao, Chenghao Yang, Xiaoxiao Guo, Saloni Potdar, and Hui Su. 2020. Frustratingly hard evidence retrieval for qa over books. *ACL Nuse Workshop*.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako

Asano, and Junji Tomita. 2019. Multi-style generative reading comprehension. *arXiv preprint arXiv:1901.02262*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, pages 311–318.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL 2018*, pages 2227–2237.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. In *Proceedings of ACL 2018*, pages 784–789.

Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. 1995. Okapi at trec-3. *Nist Special Publication Sp*, 109:109.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. 2019. Improving machine reading comprehension with general reading strategies. In *Proceedings of NAACL 2019*, pages 2633–2643.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of ACL 2019*, pages 4922–4931.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57:45.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018a. R^3: Reinforced ranker-reader for open-domain question answering. In *AAAI 2018*.

Shuohang Wang, Mo Yu, Jing Jiang, Wei Zhang, Xiaoxiao Guo, Shiyu Chang, Zhiguo Wang, Tim Klinger, Gerald Tesauro, and Murray Campbell. 2018b. Evidence aggregation for answer re-ranking in open-domain question answering. In *ICLR 2018*.

Shuohang Wang, Luowei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. 2020. Cluster-former: Clustering-based sparse transformer for long-range dependency encoding. *arXiv preprint arXiv:2009.06097*.

Oscar Wilde and Guido Fornelli. 1916. *An ideal husband*. Putnam.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019a. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

Wenhan Xiong, Jiawei Wu, Hong Wang, Vivek Kulkarni, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019b. Tweetqa: A social media focused question answering dataset. In *Proceedings of ACL 2019*, pages 5020–5031.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP 2018*, pages 2369–2380.

Yury Zemlyanskiy, Joshua Ainslie, Michiel de Jong, Philip Pham, Ilya Eckstein, and Fei Sha. 2021. Readtwice: Reading very large documents with memories. *arXiv preprint arXiv:2105.04241*.

System	Bleu-1	Bleu-4	Meteor	Rouge-L	EM	F1
Public Extractive Baselines						
BiDAF (Kočiskỳ et al., 2018)	5.82/5.68	0.22/0.25	3.84/3.72	6.33/6.22	–	–
R^3 (Wang et al., 2018a)	16.40/15.70	0.50/0.49	3.52/3.47	11.40/11.90	–	–
BERT-heur (Frermann, 2019)	–/12.26	–/2.06	–/5.28	–/15.15	–	–
DS-Ranker + BERT (Mou et al., 2020)	14.60/14.46	1.81/1.38	5.09/5.03	14.76/15.49	**6.79/6.66**	13.75/**14.45**
ReadTwice(E) (Zemlyanskiy et al., 2021)	**21.1/21.1**	**3.6/4.0**	**6.7/7.0**	**22.7/23.3**	–/–	–/–
Our Extractive QA Models						
BM25 + BERT Reader	13.27/13.84	0.94/1.07	4.29/4.59	12.59/13.81	4.67/5.26	11.57/12.55
+ HARD EM	14.39/–	1.72/–	4.61/–	14.10/–	5.92/–	12.92/–
+ ORQA	15.06/14.25	1.58/1.30	**5.28**/5.06	**15.42**/15.22	6.25/6.19	**14.58**/14.30
+ *Oracle IR (BM25 w/ Q+A)*	*23.81/24.01*	*3.54/4.01*	*9.72/9.83*	*28.33/28.72*	*15.27/15.39*	*28.42/28.55*
Public Generative Baselines						
AttSum (top-20) (Kočiskỳ et al., 2018)	19.79/19.06	1.79/2.11	4.60/4.37	14.86/14.02	–	–
IAL-CPG (Tay et al., 2019)	23.31/22.92	2.70/2.47	5.68/5.59	17.33/17.67	–	–
- curriculum	20.75/–	1.52/–	4.65/–	15.42/–		
DS-Ranker + GPT2 (Mou et al., 2020)	24.94/–	4.76/–	7.74/–	21.89/–	6.79/–	19.67/–
Our Generative QA Models						
BM25 + BART Reader	24.52/25.30	4.28/4.65	8.68/9.25	23.16/24.47	6.28/6.73	21.16/22.28
+ DS-Ranker	24.91/25.22	4.28/4.60	8.63/8.82	23.39/24.10	6.67/6.93	21.31/21.93
+ HARD EM	25.83/–	4.48/–	8.75/–	24.31/–	7.29/–	21.91/–
+ Our Ranker	27.06/27.68	5.22/5.45	9.35/9.74	25.83/26.95	8.57/8.95	23.80/25.08
+ Preread	28.54/–	**6.13**/–	9.59/–	26.82/–	10.21/–	25.06/–
+ FiD	28.04/–	5.66/–	9.49/–	26.27/–	9.20/–	24.29/–
+ FiD + Preread	**29.56/29.98**	6.11/**6.31**	**10.03/10.33**	**27.91/29.21**	**10.45/11.16**	**26.09/27.58**
+ *Oracle IR (BM25 w/ Q+A)*	*35.04/36.41*	*8.84/9.08*	*14.78/15.07*	*37.75/39.32*	*15.78/17.27*	*37.71/38.73*
BM25 + GPT-2 Reader	24.54/–	4.74/–	7.32/–	20.25/–	5.12/–	17.72/–
+ Our Ranker	24.85/–	5.01/–	7.84/–	22.22/–	7.29/–	20.03/–
+ *Oracle IR (BM25 w/ Q+A)*	*33.18/32.95*	*8.16/7.70*	*12.35/12.47*	*34.83/34.96*	*17.09/15.98*	*33.65/33.75*
BM25 + T5 Reader	19.28/–	3.67/–	6.62/–	16.89/–	4.17/–	15.47/–
+ Our Ranker	22.35/–	4.31/–	7.59/–	20.57/–	6.13/–	18.48/–
+ *Oracle IR (BM25 w/ Q+A)*	*31.06/31.49*	*8.36/8.32*	*12.61/12.93*	*31.18/32.43*	*12.77/12.84*	*31.23/32.18*

Table 8: Full results on NarrativeQA dev/test set (%) under the Book QA setting. We perform model selection based on the Rouge-L score on development set. DS is short for Distant Supervision in Sec. 4.2.

A Full Results on NarrativeQA

Table 8 gives full results with different metrics.

B Details of ICT Training Data Creation

Our pilot study shows that uniformly sampling the sentences and their source passages as "pseudo-questions" (PQs) and "pseudo-evidences" (PEs) does not work well. Such selected PQs have high probability to be casual, e.g., "*Today is sunny*", thus are not helpful for ranker training.

To select useful PQs, we define the following measure $f(s, b_j)$ to level the affinity between each candidate sentence s and the book b_j:

$$f(s, b_j) = \sum_{w_{ik} \in s} \texttt{pmi}(w_{ik}, b_j) \qquad (5)$$

where $\texttt{pmi}(w_k, b_j)$ is the word-level mutual-information between each word $w_{ik} \in s$ and the book b_j. Intuitively, $\texttt{pmi}(w_k, b_j)$ can be seen as the "predictiveness" of the word w_k with respect to the book b_j, and $f(s, b_j)$ measures the aggregated 'importance" for s. Consequently, the sentence s with the highest $f(s, b_j)$ from each passage p_n will be selected as the PQ; the corresponding p_n with the PQ removed becomes the positive sample; whereas the corresponding negative samples from the same book b_j will be the top-500 passages (exclusive of the source passage p_n) with the highest TF-IDF similarity scores to the PQ.

During sampling, we filter out stopwords and punctuation when computing $f(s, b_j)$. In movie scripts, the instructive sentences like "*SWITCH THE SCENARIO*" that have poor connections to its source passages are also ignored. Finally, we require each PQ contain a minimum number of 3 non-stopwords.