# CHAPTER 4
# OPEN-DOMAIN RETRIEVAL FOR BOOK QA IN THE ABSENCE OF ANNOTATIONS

## 4.1 Overview

In the next two chapters, we discuss the essential role played by the relations among the events in the book QA task. The book QA task is a special case of open-domain QA, answering questions about books and movie scripts. The study is a response to some challenges discussed in Section 1.3.2, including the lack of annotations, dependencies between evidence and question, and the flexibility of natural languages. This chapter focuses on building an efficient ranker for evidence retrieval without supporting evidence annotations.

Recent advancements in open-domain QA (ODQA), for example, finding answers from large open-domain document collections such as Wikipedia, have led to human-level performance in many datasets. However, progress in QA regarding book stories (book QA) lags behind, despite its similar task formulation to that of ODQA. Book QA presents certain unique challenges [31]: (1) the narrative writing style of book stories differs from the formal texts of Wikipedia and the news and demands a deeper understanding capability; the flexible writing styles used in various genres by different authors create a daunting challenge; (2) the passages that depict related book plots and characters share more semantic similarities than do the Wikipedia articles, which increases confusion in finding the correct evidence to answer a question; (3) the free-form nature of the answers necessitates the summarization of the narrative plots; and (4) the free-form answers make it difficult to obtain fine-grained supervision at the passage or span level.[8]

This chapter first provides a comprehensive and quantitative analysis of the difficulties of book QA in which we analyze the detailed challenges in book QA through a set of human studies. Our findings indicate that event-centric questions dominate this task, which

Portions of this chapter have previously appeared as: X. Mou, C. Yang, M. Yu, B. Yao, X. Guo, S. Potdar and H. Su, "Narrative question answering with cutting-edge open-domain QA techniques: a comprehensive study," *Trans. Assoc. Comput. Linguistics (TACL)*, vol. 9, pp. 1032-1046, Sep. 2021.

Portions of this chapter have previously appeared as: X. Mou, M. Yu, B. Yao, C. Yang, X. Guo, S. Potdar, H. Su, "Frustratingly hard evidence retrieval for QA over books," In *Proc. 1st Workshop Narrative Understanding Storylines Events (NUSE)*, 2020, pp. 108-113.

[8] In addition to the four challenges, book QA has a specialty that the paragraphs form a logically related sequence. We consider this specialty more like an opportunity than challenges, and leave its investigation to future work.

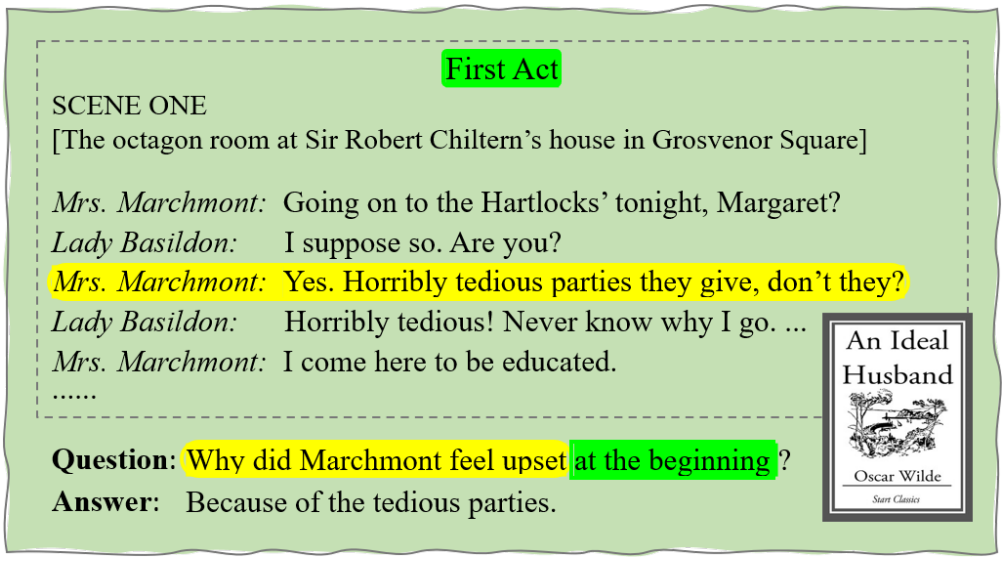


**Figure 4.1: An example of book QA. The content is from the book *An Ideal Husband* [184]. The bottom contains a typical QA pair, and the highlighted text is the evidence for deriving the answer.**

exemplifies the inability of existing QA models to handle event-oriented scenarios. Then we train different rankers with two families of techniques and test on the benchmark NarrativeQA dataset. We extensively explore the usage of cutting-edge ODQA techniques in evidence retrieval, and also probe the effects of commonsense knowledge from various event-centered datasets with our proposed multitask training strategy. Our experiments demonstrate the feasibility of our proposed methods in dealing with the lack of evidence annotation.

## 4.2 Related Work

**Book QA:** Previous works [31], [185], [186] have also adopted a ranker–reader pipeline, although they have not fully investigated state-of-the-art ODQA techniques. First, although NarrativeQA is a generative QA task, the application of the latest pretrained LMs, such as BART for generation purposes, has not been well studied. Second, the lack of fine-grained supervision of evidence prevented earlier methods from training neural ranking models; thus, they only used simple BM25-based retrievers. An exception is Mou et al. [168], who constructed pseudo-distance supervision signals for ranker training. Another relevant work [186] used book summaries as an additional resource to train rankers. However, this differs from the aim of the book QA task, which is to answer questions solely from books, since in a general scenario, a book summary cannot answer all questions about a book. Our

work is the first to investigate and compare improved training algorithms for rankers and readers in book QA.

**Open-Domain QA:** ODQA aims to answer questions from large open-domain corpora (e.g., Wikipedia). Recent work has adopted a ranker–reader framework [121], and success in this field derives primarily from improvements in the following directions: (1) distantly supervised training of neural ranker models [6], [163], [187], [188] to select relevant evidence passages; (2) fine-tuning and improving pretrained LMs such as ELMo [189] and BERT [176] as rankers and readers; and (3) the unsupervised adaptation of pretrained LMs to the target QA tasks [190]–[192]. We borrow ODQA techniques and apply them to the book QA task.

**Lottery Ticket Hypothesis:** Integrating external knowledge into neural networks usually involves multitask learning on several heterogeneous datasets. The process of traditional multitask learning is pragmatically tedious and can be repetitive when new data is available. We explore a more efficient way for multitask learning and avoid the repetitive training process by applying the Lottery Ticket Hypothesis (LTH). LTH receives an increasing attention and we leverage the advances of LTH to introduce external knowledge into book QA. Frankle and Carbin [193] propose the idea of lottery ticket hypothesis (LTH) for the first time, claiming the existence of a sparse subnetwork in a dense neural network that can achieve on-par or better performance compared to the original full-size neural network. They also provide the iterative magnitude pruning (IMP) method to find the subnetwork. Gale et al. [194], Liu et al. [195], Frankle et al. [196], Zhou et al. [197] improve the IMP method by using different learning strategies and scoring metrics to find a more stable and robust subnetwork. Morcos et al. [198], Mehta [199], Chen et al. [200], [201], Ansell et al. [202] further study the transferability of the lottery tickets across different datasets and tasks, among which Morcos et al. [198], Mehta [199], Chen et al. [200] focus on computer vision tasks while Chen et al. [201], Ansell et al. [202] in the NLP domain. Different from [201], [202], our study proves the transferability in more complicated QA tasks instead of basic NLP tasks.

## 4.3 Human Study: Event-Centric Questions and Evidence

Before investigating the off-the-shelf LMs and technologies on the book QA task, we first conducted in-depth analyses of the challenges in book QA by studying the data of the

benchmark NarrativeQA dataset. We proposed a new question categorization scheme based on the types of comprehension or reasoning skills required for answering the questions; then we conducted a human study using 1,000 questions. Consequently, the model performance per category provided further insights into the deficiencies in current QA models.

**Table 4.1: Definitions of semantic units (SUs). The underlined texts represent the recognized SUs of the types.**

| 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 |

### 4.3.1 Question Categorization

There have been many different question categorization schemes. Among them, the most widely used is intention-based, in which an intention is defined by the WH word and the word that follows. Some recently developed reasoning-focused datasets [10], [203] categorize intention by types of multihop reasoning or required external knowledge beyond texts.

However, none of the previous schemes fit reasonably with our analysis of narrative texts in two aspects: (1) they only differentiate high-level reasoning types, which is useful in knowledge-based QA (KBQA) but fails to pinpoint 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 in mind, we designed a new systematic schema to categorize the questions in the NarrativeQA dataset.

**Table 4.2: Definitions of question types. Note that sometimes the answer repeats parts of the question, as in the last two examples in the second block; we ignore these parts when recognizing the SUs in answers.**

| 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 |

**Semantic Unit Definition:** We first identified a minimum set of basic semantic units, each describing one of the most fundamental components of a story. The set needed to be sufficient such that (1) each answer can be uniquely linked to one semantic unit, and (2) each question must contain at least one semantic unit. Our final set contained three main classes and nine subclasses (Fig. 4.1).

We merged the two most commonly used types from the previous analysis—entities and noun phrases—into the concept class. The event class follows the definition in Ace (2005) [204]. We also used a special subtype—book attribute—that represents the meta information or the global settings of the book such as the era and the theme of the story.

**Question Type Definition:** In addition to the semantic units' definitions, each question can be categorized as a query that asks about either a semantic unit or a relation between two semantic units. We used the difference and split all the questions into nine types grouped into four collections (Fig. 4.2).

• **Concept questions** ask for a concept attribute or a relation between two concepts. The most common types in most ODQA (e.g., TriviaQA) and QA tasks require multihop reasoning (e.g., ComplexQuestions and HotpotQA).

• **Event-argument questions** ask about parts of an event structure. This type is less common in the existing QA datasets, although some datasets like SQuAD contain a small portion of questions in this class. The large ratio[9] of these event-centric questions demonstrates the uniqueness of the NarrativeQA dataset.

• **Event-relation questions** ask about relations such as causal or temporal relations between two events or between an event and an attribute (a state or a description). This is the most common class in NarrativeQA, but it is not significant in other benchmarks since events play essential roles in story narrations. A particular type in this group is the relation that one event has to the argument of another event (e.g., "how" questions); this corresponds to the common linguistic phenomenon of nested event structures.

• **Global-attribute questions** ask about book attributes, and they are unique to book QA.

### 4.3.2 Annotation Details

Five annotators were asked to label the semantic unit types and the question types on a total of 1,000 question-answer pairs. Overlapping question categories for the same question were allowed. A major kind of overlap was between the three event-component types (trigger, argument, and concept/attribute) and the three event relation types (causal, temporal, and nested). Therefore, when the question could be answered with an event component, we asked the annotators to check whether the question required the understanding of event relations. If so, the question was labeled an event-relation type as this represents the more critical information needed to find the answers. Similarly, for the other rare cases of category overlap, we asked the annotators to label the types that they believed were more important for finding the answers.

**Correlation between question and answer types:** Figure 4.2 shows the ratios of answer types under each question type via a flow diagram. Most question types corresponded to a single major answer type, with a few exceptions: (1) Most of the three event-relation questions received events as answers. A small portion of them received concepts or attributes as answers. This was either because the answers were state/description attributes or because the answers were arguments for one of the related events queried by the questions. (2) The relation-between-concepts type showed some questions with attribute-type answers. This

[9]Together with event-relation questions

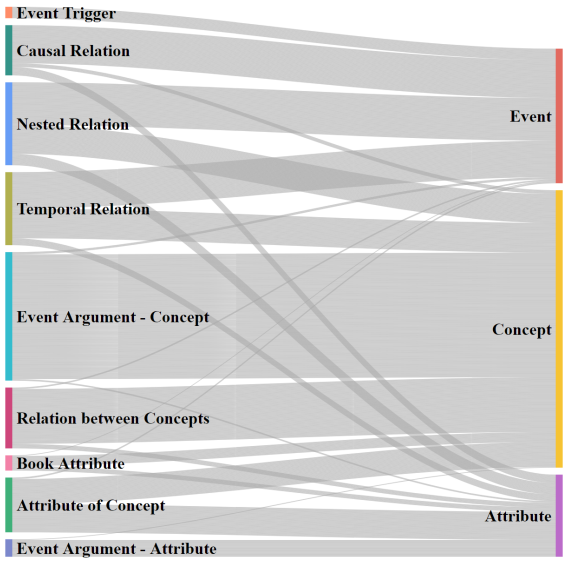


**Figure 4.2: Visualization of the flow from the question types to the expected answer types.**

**Table 4.3: Annotation agreement. SU: semantic unit. "SU type" and "SU Subtype" are defined in Table 4.1.**

| Category | Simple Agreement(%) | $\kappa$(%) |
|---|---|---|
| Question Type | 88.0 | 89.9 |
| SU Type | 92.3 | 91.2 |
| SU Subtype | 81.3 | 82.8 |

is because the questions may have asked for the names of the relations themselves, while some relation names were recognized as description-type attributes. (3) Most of the book attribute questions received concepts as answers because they asked for the protagonists or the locations where stories occurred.

**Annotation agreement:** A subset of 150 questions was used for quality checking, with each question labeled by two annotators. Table 4.3 reports both the simple agreement rates and the Fleiss' kappa ($\kappa$s) [205]. Our annotations reached high agreement with around 90% for question and SU types and 80% for SU subtypes, reflecting the rationality of our scheme.

### 4.3.3 Performance of Question-Type Classification on the Annotated Data

We conducted an additional experiment to study how well a machine learning model could learn to classify our question types based on the questions' surface patterns. We

used the RoBERTa model, which has demonstrated superior accuracy in multiple sentence classification tasks. Since the amount of our labeled data was small, we conducted a 10-fold cross validation on our labeled 1,000 instances. For each testing fold, we randomly selected another fold as the development set and used the rest of the folds as training.

The final averaged testing accuracy was 70.2 percent. Considering the interagreement rate of 88.0%, this is a reasonable performance, and there are several reasons for the gap: (1) Our training data is small and therefore easy to overfit, evidenced by the performance gap between training accuracy and development accuracy ($\sim$100% versus 73.4%). The accuracy can be potentially increased with more training data. (2) Some of the ambiguous questions required context to determine their types. During labeling, our human annotators were allowed to read the answers for additional information, which led to a higher upper-bound performance. (3) There were a small number of ambiguous cases in which humans could use world knowledge, while models have difficultly employing such knowledge. Therefore, the current accuracy can be potentially increased through better model architecture.

**Error Analysis and Lessons Learned:** Figure 4.4 presents the major error types, verifying the reasoning in our above discussion. The majority of errors were based on the confusion between event argument - concept and nested relation. The models are not accurate for these two types for several reasons: (1) Sometimes similar surface forms of questions can take both concepts and events as arguments. In these cases, the answers are necessary for determining the question type. (2) In our annotation guidelines, we encouraged the annotators to label event relations with higher priority, especially when the answer was a concept but served as the argument of a clause. This increased the labeling error rate between the two types. Another major error type involves labeling causal relations as nested relations. This occurs mainly because some questions ask causal relations in an implicit way, for which human annotators have the commonsense to identify the causality but models do not. The third major type concerns failures in identifying the attribute of a concept and the relation between concept 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 insight into future refinements of our annotation guidelines to further increase the amount of labeled data. For example, the nested-relation category should be more clearly defined with comprehensive examples provided. In this way,

**Table 4.4: Error analysis of question-type classification. We only list the major errors of each type (i.e., incorrect predicted types that led to >10% of the errors).**

| Ground-truth 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 |

annotators can better distinguish them from other types and can better determine if a nested structure exists and how to label the event argument types. Similarly, we can define clearer decision rules for relations, attributes, and events to help annotators distinguish between concept, attribute of concept, and event argument - concept types.

## 4.4 Methods

The investigation of the NarrativeQA dataset in the previous section provides a qualitative insight into the book QA task. In this section, we first officially define the book QA task and then introduce two groups of methods. The first group leverages the existing QA techniques and is adapted to the book QA task. The second group considers and makes use of the findings in the previous human study for further improvements.

### 4.4.1 Task Definition

Following Kočiskỳ et al. [31], we defined the **book QA** task as finding the answer **A** to a question **Q** from a book, where each book contains a number of consecutive and logically related paragraphs $\mathcal{C}$. The size $|\mathcal{C}|$ from different books varies from a few hundred to thousands.

### 4.4.2 Open-Domain QA Rankers

This section describes our efforts in applying or adapting the latest open-domain QA ideas to improve book QA ranker/reader models. Fig. 4.5 summarizes the approaches we examined. The experimental results quantify the challenges in book QA beyond open-domain QA.

**Table 4.5: Summary of the approaches we inspected for the ranker. *We directly applied the heuristics from [168] for book QA.**

| Approach | Original Idea in ODQA | Our Improved Version for book QA |
|---|---|---|
| Heuristic distant supervision | N/A | N/A* |
| Unsupervised ICT | Proposed by [190] as Siamese network for both BERT pretraining and dense retrieval. | We improve the method with our book-specific training data selection. |
| Hard EM | Proposed by [187] for reader training. | We adapt the method for ranker training. |

#### *4.4.2.1 Method 1: Distant Supervision*

This is the baseline approach from Mou et al. [168]. It constructs distance supervision (DS) signals for rankers in two steps. First, for each question **Q**, two BM25 rankers are used to retrieve passages, one with **Q** as the query and the other with both **Q** and the true answer **A**. Denoting the corresponding retrieval results as $\mathcal{C}_{\mathbf{Q}}$[10] 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 **Q** as a hyperparameter.

Next, to enlarge the margin between the positive and negative samples, the method applies a **Rouge-L filter** on the previous sampling results to get the refined samples $\mathcal{C}^{++}_{\mathbf{Q}}$ and $\mathcal{C}^{--}_{\mathbf{Q}}$.

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

$$\mathcal{C}_{\mathbf{Q}}^{++} = \left\{ \max_{\mathbf{S}\subset\mathbf{C}_i, |\mathbf{S}|=|\mathbf{A}|} Sim(\mathbf{S}, \mathbf{A}) > \alpha, \mathbf{C}_i \in \mathcal{C}_{\mathbf{Q}}^{+} \right\} \tag{4.1}$$

$$\mathcal{C}_{\mathbf{Q}}^{--} = \left\{ \max_{\mathbf{S}\subset\mathbf{C}_i, |\mathbf{S}|=|\mathbf{A}|} Sim(\mathbf{S}, \mathbf{A}) < \beta, \mathbf{C}_i \in \mathcal{C}_{\mathbf{Q}}^{-} \right\}. \tag{4.2}$$

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

#### *4.4.2.2 Method 2: Unsupervised Inverse Cloze Task Training*

[11] Inspired by the effectiveness of inverse cloze task (ICT) [190] as an unsupervised ranker training objective, we use it to pretrain our ranker by constructing a "pseudo-question" $q$ and "pseudo-evidence" $b$ from the same original passage $p$ aimed at maximizing the probability $P_{\text{ICT}}(b|q)$ of retrieving $b$ given $q$, which is estimated using negative sampling as follows:

$$P_{\text{ICT}}(b|q) = \frac{\exp\left(S_{retr}(b,q)\right)}{\sum_{b'\in B}\exp\left(S_{retr}\left(b',q\right)\right)}. \tag{4.3}$$

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

The selection of "pseudo-questions" is critical to ICT training. To select representative questions, we investigated several filtering methods, finally developing a book-specific filter [12]. Our method selects the top-scored sentence in a passage as a "pseudo-question" in terms of its total token-wise mutual information within the corresponding book. The details can be found in Appendix A.2.

#### *4.4.2.3 Method 3: Hard EM*

Hard EM is an iterative learning scheme. It was first introduced to ODQA by [187], to find correct answer spans that maximize reader performance. [13] Here, we adapted the algorithm to ranker training. Specifically, 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

[11] The work is originated from the equal contribution in *Narrative Question Answering with Cutting-Edge Open-Domain QATechniques: A Comprehensive Study.*

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

[13] As shown in [206], the Hard EM objective is related to policy gradient. Therefore, we categorize [187] and [6] to the same class.

its parameters $\Phi^{t+1}$; it then derives the new positive passages $\mathcal{C}_Q^{+t+1}$ that maximize the probability of reader $\Phi^{t+1}$ predicting $\mathbf{A}$ (Eq. 4.4). The M-step updates the ranker parameter $\Theta$ (Eq. 4.5):

$$\mathcal{C}_Q^{+t+1} = k\text{-}\max_{\mathbf{C}_i \in \mathcal{C}} P(\mathbf{A}|\mathbf{C}_i, \Phi^{t+1}) \tag{4.4}$$

$$\Theta^{t+1} = \arg\max_{\Theta} P(\mathcal{C}_Q^{+t+1}|\Theta^t). \tag{4.5}$$

### 4.4.3 Event-Aware Rankers

The human study in Section 4.3 reveals the event-centric nature of the questions and the importance of the event-related knowledge to the QA tasks. However, explicitly modeling the cross-event relationship for a book remains impractical, considering that the events are not necessarily represented at sentence level but can be at a higher level, even comprising several paragraphs. An alternative way of modeling event structure is to incorporate the commonsense about the events by re-pretraining an LM with different event-oriented auxiliary tasks other than the QA tasks mentioned in Section 4.4.2. In this section, we outline several event-oriented tasks we use to train event-enhanced rankers and propose an efficient way of introducing external structural knowledge into neural networks.

#### *4.4.3.1 Contextual and Transferable Mask Learning*

Contextual and transferable mask learning (CTML) is a strategy for efficiently migrating structure knowledge from one domain to another via a contextual mask, backed by the LTH [193]. In the LTH, we consider a dense network, $f(\mathbf{x};\theta)$, with the initial parameters $\theta \sim \theta_0$. After $k$ training iterations, we have the new parameters, $\theta_k$. Following the IMP method [193], we select a subset of the parameters, whose absolute differences in magnitude are greater than a threshold, $\sigma$. In our experiment, we simply rank the parameters by their absolute differences in magnitude from top to bottom, and we select the top $\epsilon\%$. We denote the selected subset at iteration $k$ as $m \odot \theta_k$, where $m_k \in \{0,1\}^{|\theta|}$ is a mask. For the unselected parameters, we reset them to the initial values and freeze them in the rest of the training. Therefore, with mask $m_k$, we have our sparse network

$$f(\mathbf{x}, \mathbf{k} \odot \theta_{\mathbf{k}} + (\mathbf{1} - \mathbf{m_k}) \odot \theta_{\mathbf{0}}) \tag{4.6}$$

Following Frankle and Carbin [193], Frankle et al. [196], Yu et al. [207], we next prune the network iteratively by selecting $\epsilon\%$ of the resting parameters,

$$\frac{||m^{(n+1)}||_0}{||m^{(n)}||_0} = \epsilon\% \tag{4.7}$$

and the size of the subnetwork becomes increasingly smaller.

Each mask is task specific and contextualized, which suggests that the selected parameter and structure are associated with certain specific abilities. The transferability of the masks is defined by the phenomenon so that when multiple masks are combined (i.e., $m_{new} = \sum_{i=1}^{n} m_i$), so are their corresponding abilities. This allows these contextual masks to be trained only once on an auxiliary task and the associated knowledge to be incorporated into a target domain by simply applying the mask to the backbone model; this significantly reduces the effort and difficulties of multitask learning.

To sum up, our proposed CTML follows three steps: (1) training contextualized masks on each auxiliary task; (2) combining masks to form an integrated mask; and (3) fine-tuning the backbone model on the target task with the integrated mask. CTML not only simplifies knowledge transfer but also speeds up the fine-tuning procedure, because only a subset of parameters must be tuned during the training.

#### *4.4.3.2 Auxiliary Tasks and Datasets*

**Table 4.6: A summary of the event-centered dataset that are used to improve our rankers.**

| Dataset | Original Task | Knowledge |
|---|---|---|
| BookSum [208] | Summerization | Temporal, Causal, Discourse |
| ATOMIC (Subevent subset) [209] | Commonsense | Subevent, Temporal, Causal |
| Stories [210] | Subevent | Subevent, Temporal |

Having proposed CTML, we introduce three eventive auxiliary tasks in this part that are used in our experiments to enhance models' awareness of event-related properties.

**BookSum** BookSum is a dataset for long-form summarization tasks. It contains collections of narrative documents, such as novels and movie scripts, that resemble the documents in the NarrativeQA dataset in terms of document length and writing style. The dataset also

provides highly abstractive paragraph-, chapter-, and book-level human written summaries. Summarization on varying levels of granularity requires the capability to infer non-trivial causal and temporal dependencies and identify discourse structures that are considered potentially beneficial for increasing the event awareness of a ranker. Our experiments use only paragraph-level data, whose sources and target lengths better match the actual use cases in our retrieval tasks. Since the paragraph-level summarization pairs are aligned by algorithm rather than manually, we add an extra Rouge-L filter. In doing so, we obtain 1,389 training pairs and 348 testing pairs with negative sampling.

**ATOMIC$^{20}_{20}$** ATOMIC$^{20}_{20}$ is a general-purpose, commonsense knowledge graph containing 1.33M commonsense knowledge tuples across 23 commonsense relations that covers social, physical, and eventive aspects of everyday inferential knowledge. Our experiments employ the subevent subset of the entire knowledge graph, which contains 12,845 tuples; we believe this to be the most beneficial for our retrieval task. In the subset, the events are represented by short sentences (e.g., "X gets X's car repaired"), and we formulate a binary classification task to identify whether one event is a subevent of another. With our negative sampling technique, we obtain a total of 53,796 training pairs and 11,568 dev pairs.

**Stories** Stories collects sets of natural language descriptions of script-specific event sequences from volunteers over the internet. The sequence of subevents is temporally ordered and taken together to fulfill a task. For example, to "answer the doorbell," one must complete the following steps: "move to door," "unlock door," and "open door." The dataset includes 185 parent events and 3,180 annotations of subevent sequences. Following a similar preprocessing procedures to those of BookSum and ATOMIC$^{20}_{20}$, we generate 24,085 positive and 189,605 negative pairs and split them into a training set of 143,216 and a test set of 35,805.

## 4.5 Evaluation

In this section, we describe our experimental settings, followed by our analysis of the evidence retrieval performance and the ablation study.

### 4.5.1 Experiment Setup

#### *4.5.1.1 Dataset*

All our experiments were conducted on the NarrativeQA dataset [31], which contains a collection of 783 books and 789 movie scripts (we use the term "books" to refer to both of these), each containing an average of 62K words. Additionally, each book contains 30 question-answer pairs generated by human annotators in free-form natural language. Hence, exact answers are not guaranteed to appear in the books. NarrativeQA provides two different settings, the **summary** and **full-story** settings. The former requires answering questions based on book summaries from Wikipedia, and the latter requires answering questions based on the original books, assuming that summaries do not exist. Our book QA task corresponds to the full-story setting, and we use both names interchangeably.

Following Kočiskỳ et al. [31], we tokenized the books with SpaCy[14] and split each book into nonoverlapping trunks of 200 tokens.

#### *4.5.1.2 Baseline Ranker*

Following the formulation of the open-domain setting, we employed 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,

$$\mathcal{C}_{\mathbf{Q}} = \text{top-k}(\{P(\mathbf{C}_i|\mathbf{Q})|\forall\ \mathbf{C}_i \in \mathcal{C}\}); \tag{4.8}$$

and then a *reader model* predicts answer $\tilde{\mathbf{A}}$ given $\mathbf{Q}$ and $\mathcal{C}_{\mathbf{Q}}$. In this chapter, we only focus on the ranker module.

Our baseline system is a BM25 ranker. It estimated the likelihood of each passage being supporting evidence given a question $\mathbf{Q}$. In addition, we also compared with competitive public book QA systems as baselines following [31], [168], [185], [186] under the narrative full-story setting. As discussed in Section 4.2, Mou et al. [168] trained a ranker with distant supervision (DS), the first analyzed ranker method (Fig. 4.5). Yin et al. [211] provided a newly trained LM DocNLI that is equipped with the natural language inference capabilities at various scales, beyond the sentence-level inference.

[14] `https://spacy.io/`

**Table 4.7: Characteristics of the compared systems, following Mou et al. [168]. †/‡ refers to generative/extractive QA systems. In addition to the standard techniques, Wang et al. [6] used reinforcement learning to train the ranker, and Tay et al. [185] used curriculum to train the reader to overcome the divergence of retrieval qualities between training and testing.**

| System | trained ranker | pretrained LM | extra data |
|---|---|---|---|
| IR+AttSum† [31] | | | |
| IR+BiDAF‡ [31] | | | |
| IAL-CPG† [185] | | | |
| R³‡ [6] | ✓ | | |
| BERT-heur‡ [186] | ✓ | ✓ | ✓ |
| DS Ranker+GPT2† [168] | ✓ | ✓ | |
| DS Ranker+BERT‡ [168] | ✓ | ✓ | |
| Our best QA system† | ✓ | ✓ | |

#### *4.5.1.3 Metrics*

Following previous works [31], [185], [186], we used the study of Rouge-L [212] as the main metric to evaluate the evidence retrieval quality. Considering the large length difference between the retrieved passage and the true answer, we computed their rolling Rouge-L as follows: we went through the context, iteratively computed the Rouge-L score between the true answer and each candidate span of the same length as the true answer having overlapping considered, and finally took the maximum value as our rolling Rouge-L score.

#### *4.5.1.4 Hyperparameters*

During implementation, we initialized rankers with `bert-base-uncased`.[15] The distant supervision algorithm finds the best performance at $\alpha = 0.7$, $\beta = 0.3$, and $\sigma = 8$ (Section 4.4.2).

### 4.5.2 Ranker Performance

To further explore the effects of our ranker training techniques, as discussed in Section 4.4.2, we studied the retrieval results and measured their coverage of the answers. Coverage is estimated based on the top-five selections of a ranker from the baseline BM25's top-32 outputs, by both the maximum Rouge-L score of all of the overlapped occurrences

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

**Table 4.8: Ranker performance (top-five) on dev set. DS stands for distant supervision. In the middle block, we list the top three/ten results, which are used in our readers (* for BART and ** for FiD; see our reader ablation in Section 5.6).**

| IR Method | EM | Rouge-L |
|---|---|---|
| BM25 (baseline) | 18.99 | 47.48 |
| BERT ranker (with DS) | 24.26 | 52.68 |
| - Rouge-L filtering | 22.63 | 51.02 |
| + Hard EM | 22.45 | 50.50 |
| + ICT | 24.83 | 53.19 |
| Repl BERT w/ BiDAF | 21.88 | 50.64 |
| Repl BERT w/ MatchLSTM | 21.97 | 50.39 |
| DocNLI DS-ranker [211] (baseline) | 25.04 | 53.43 |
| + BookSum | 25.25 | 53.63 |
| + Atomic Subevent | 25.22 | 53.64 |
| + Story Subevent | 25.36 | **53.85** |
| + multitask | 25.39 | 53.68 |
| + multi-mask | **25.42** | 53.59 |
| BM25 (top-3)* | 15.75 | 43.44 |
| BM25 (top-10)** | 24.08 | 53.55 |
| Best Ranker (top-3)* | 22.12 | 49.83 |
| Best Ranker (top-10)** | 27.15 | 56.77 |
| Upperbound (BM25 top-32) | 30.81 | 61.40 |
| Oracle (BM25 w/ Q+A) | 35.75 | 63.92 |

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 4.8 shows ranker-only ablation. On the one hand, our best ranker improved both metrics over the BM25 baseline, but on the other, the gain comes primarily from the effective encoding and representation capability of the BERT model and our proposed Rouge-L filtered distant supervision. The performance drop caused by replacing BERT with the BiDAF or MatchLSTM models further demonstrates the superiority of the pretrained LM. Furthermore, none of the other techniques could further improve the ranker significantly. The ICT unsupervised training brought significant improvement over BM2, while Hard EM [187] did not lead to any improvements. We posit that a generative reader generates other than purely match-oriented signals thus introducing noise into matching-oriented ranker training.

Table 4.8 also clearly demonstrates that incorporating event-related commonsense from

**Table 4.9: Overall QA performance (%) in NarrativeQA book QA setting with a pretrained BART reader. *Oracle IR* combines question and true answers for BM25 retrieval. The asterisk (*) indicates the best results reported in [31] with multiple hyper-parameters on dev set. The dagger ($^\dagger$) indicates significance with p-value $< 0.01$.**

| System | Rouge-L | |
|---|---|---|
| | dev | test |
| Seq2Seq [31] | 13.29 | 13.15 |
| AttSum* [31] | 14.86 | 14.02 |
| IAL-CPG [185] | 17.33 | 17.67 |
| BERT-heur [186] | – | 15.15 |
| DS-Ranker + GPT2 [168] | 21.89 | 22.36 |
| BART-no-context (baseline) | 16.86 | 16.83 |
| + BM25 (baseline) | 23.16 | 24.47 |
| + ICT | 25.83 | 26.95$^\dagger$ |
| + Story Subevent | 27.26 | 28.21$^\dagger$ |
| *repl ranker with oracle IR* | *37.75* | *39.32* |

external knowledge at various scales is beneficial. The slight improvement from DocNLI LM [211] mostly results from its doubled parameters over the BERT model. However, when we re-pretrain the DocNLI model on event-centric auxiliary tasks, the improvements on our retrieval results can be consistently observed, up to a 0.4% rise. This also gives us the best performance, with 53.85 on Rouge-L. More details about the auxiliary tasks are covered in Section 4.5.4

Overall, the limited improvement and low absolute performance demonstrate the difficulty of retrieval in book QA. The gap between our best performance and the upper bound implies that much potential exists to design a more advanced ranker.

Furthermore, in Table 4.9, we show how much useful information our best ranker can provide to our readers in the entire QA system. In our implementation, the BART reader used top-three paragraphs from the ranker. The top-three paragraphs from our best ranker provided answer coverage of 22.12% EM and 49.83% Rouge-L, and the top-10 paragraphs provided 27.15% EM and 56.77% Rouge-L. In comparison, the BM25 baseline had 15.75%/43.44% for the top three and 24.08%/53.55% for the top 10. Our best ranker thus efficiently eased the limited-passage bottleneck brought about by the ranker, benefiting the BART reader much more, which is consistent with our observations in the study of the

reader in the next chapter.

### 4.5.3 Optimal Size of Contextual Masks

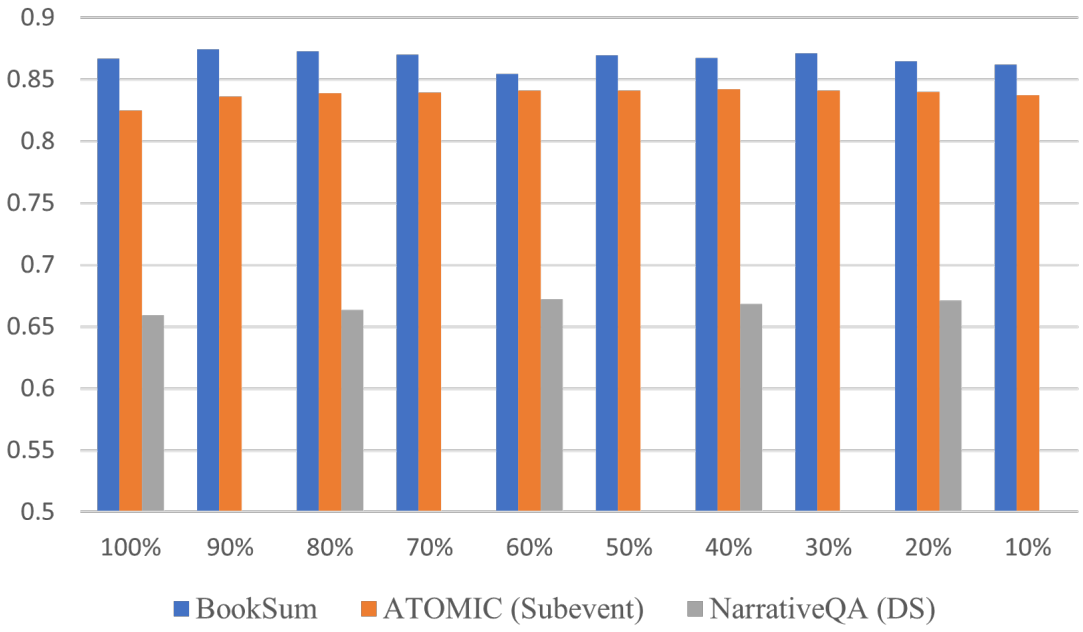


**Figure 4.3: Fine-tuned accuracy on different tasks with different sizes of masks. We test with five sizes for the NarrativeQA task due to the massive amount of time and computational resources it requires. DS stands for distant supervision and is used to emphasize that it is not the ranking or reasoning task on NarrativeQA.**

We experiment with model performance, with different mask sizes on three tasks. Table 4.3 first reveals that on each task, there is a subnetwork that has slightly better result than the original dense network, which confirms the correctness of LTH. Second, it is obvious that the performance does not fluctuate much with a varying mask size across the tasks, which differs slightly from the observations in [193], [213] but is similar to [197], [198]. We posit that the three artifact auxiliary tasks are relatively simple and do not require a sophisticated model.

### 4.5.4 Transformability of Sparse Masks

To further validate the transferability of the contextualized masks, we perform a cross-task study on a smaller scale with the help of three additional datasets: MATRES [214], McTACO [215], and CoNLL [216]. Table 4.10 compares the performance with homogeneous

**Table 4.10: F1 score (%) of the cross-task validation for the transferability of the contextualized masks. The mask size is 16% of the full size. The bold values are the best scores achieved by using their own masks.**

| Dataset | Task | No Mask | MA-Mask | Mc-Mask | Co-Mask | AT-Mask |
|---|---|---|---|---|---|---|
| MATRES | Temporal | 73.88 | **77.28** | 75.26 | 73.07 | 74.98 |
| McTACO | Temporal | 85.68 | 87.02 | **87.14** | 85.91 | 86.29 |
| CoNLL | NER | 83.59 | 84.15 | 83.99 | **86.94** | 86.52 |
| ATOMIC | Subevent | 82.49 | 84.08 | 84.13 | 83.97 | **84.35** |

and heterogeneous masks on different tasks. Two tasks/masks are considered homogeneous if they share or are trained on the same task (the second column of Table 4.10); otherwise, they are considered heterogeneous.

On each dataset, we find that (1) the best score is always achieved with the masks trained on their own tasks, and (2) the homogeneous masks lead to a smaller performance drop, while the heterogeneous ones yield a larger decrease in F1 scores. It confirms the transferability of these masks and validates the correctness of our CTML. Another interesting finding is that the subevent mask is beneficial to all other tasks and that the subevent task benefits from the masks trained on all other tasks. We reason that this occurs because the subevent is a more complex task and requires knowledge from other basic tasks.

## 4.6 Chapter Summary

In this chapter, we first conducted a comprehensive analysis of the book QA task, using the representative NarrativeQA dataset as an example. We performed a human study and found that the majority of the questions in book QA require understanding and differentiating events and their relations. The findings lead us to the event understanding task for future improvements to the book QA task. Second, our extensive experiments demonstrated that the adaptation of the techniques from cutting-edge ODQA research helped achieve better QA results than the strong pretrained LM baseline. Third, we introduced the eventive commonsense knowledge to the neural networks and brought extra gains to the retrieval task. To efficiently incorporate event-centered knowledge into a ranker, we applied the LTH to the QA retrieval problem and built a contextualized model parameter mask for each auxiliary task. The knowledge integration was then simplified into mask overlay. The overall efforts yield new state-of-the-art evidence retrieval results on the NarrativeQA dataset.