Harvesting More Answer Spans from Paragraph beyond Annotation

Qiaoben Bao[♠], Jiangjie Chen[♠], Linfang Liu[♠], Jingping Liu[♢†], Jiaqing Liang[♠], Yanghua Xiao[♠♡‡]
[♠]Shanghai Key Laboratory of Data Science, School of Computer Science, Fudan University, Shanghai, China
[♡]Fudan-Aishu Cognitive Intelligence Joint Research Center, Shanghai, China
[♢]School of Information Science and Engineering, East China University of Science and Technology, Shanghai, China
{qbbao19,jjchen19,liulf19,shawyh}@fudan.edu.cn,jingpingliu@ecust.edu.cn,l.j.q.light@gmail.com

ABSTRACT

Automatic *A*nswer span *E*xtraction (AE) focuses on identifying key information from paragraphs that can be asked. It has been used to facilitate downstream question generation tasks or data augmentation for question answering. Current work of AE heavily relies on the annotated answer spans from *M*achine *R*eading *C*omprehension (MRC) datasets. However, these methods suffer from the *partial annotation problem* due to the annotation protocols of MRC tasks. To tackle this problem, we propose SCOPE, a **S**tructured **Co**ntext graph network with **P**ositiv**e**-unlabeled learning. SCOPE first represents the paragraph by constructing a graph with both syntactic and semantic edges, then adopts a unified pointer network for answer span identification. SCOPE narrows the discrenpency between AE and MRC by formulating AE as a *P*ositive-*u*nlabeled (PU) learning problem, thus recovering more answer spans from paragraphs. To evaluate newly extracted spans without annotation, we also present an automatic metric from the perspective of question answering and text summarization, which correlates well with human judgments. Comprehensive experiments on both AE and downstream tasks demonstrate the effectiveness of our proposed framework. Our code is available at https://github.com/iambabao/SCOPE.

CCS CONCEPTS

• **Computing methodologies** → **Information extraction**; *Semantic networks*; *Learning paradigms.*

KEYWORDS

Information extraction; Positive-unlabeled learning

ACM Reference Format:
Qiaoben Bao, Jiangjie Chen, Linfang Liu, Jingping Liu, Jiaqing Liang, and Yanghua Xiao. 2022. Harvesting More Answer Spans from Paragraph beyond Annotation. In *Proceedings of the Fifteenth ACM Int'l Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3488560.3498399

[†]Work is done while at Fudan University.
[‡]Corresponding author.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498399

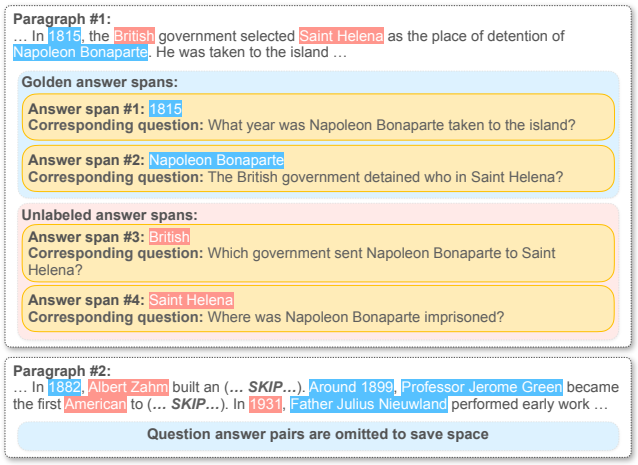

Figure 1: Examples of annotations from SQuAD that illustrate the partial annotation problem. Both paragraphs contain labeled answer spans and unlabeled answer spans.

1 INTRODUCTION

Generating question-answer pairs again unstructured paragraphs has gained significant attention. It is an essential step in data augmentation for Machine Reading Comprehension (MRC) and Question Answering (QA) tasks [2, 38]. Typically, such a system consists of a pipeline of Answer span Extraction (AE) and Question Generation (QG). It first identifies answer spans that can be asked and then generates questions with different focuses. As the first step, AE determines the quality of key information used to generate questions and is still a challenging but less-explored sub-task.

Previous work on AE focuses on designing specialized methods to extract answer spans by modeling this task as a span classification task [6, 20]. The supervision they rely on mainly comes from the existing MRC datasets such as SQuAD [29] where answer spans are explicitly given. However, the supervision in MRC tasks is usually incomplete for AE due to the different annotation protocols. The annotators in such datasets are only required to extract limited answer spans (usually 5) for each paragraph. This annotation procedure ignores other detailed key information that would also be helpful for readers to understand the context. We point out missing annotations will lead to the *partial annotation problem* [37, 40] that provides wrong supervision for previous AE methods and makes them fail to extract more answer spans from paragraphs.

To illustrate the partial annotation problem, we present two examples from SQuAD in Figure 1. In the first paragraph, "*1815*" gives the temporal message for a "**When**" question while "*Napoleon Bonaparte*" tells us "**Who**" was involved in. However, the given paragraph also mentions the location, i.e. "*Saint Helena*", which is worthy of asking for a "**Where**" question but is not labeled. Moreover, these unlabeled answer spans share high similarities with labeled samples, e.g., "*1822*" and "*Around 1899*" are annotated while "*1931*" is missed in the second paragraph. To the best of our knowledge, no previous work explicitly considers this partial annotation problem in AE tasks.

In this paper, we aim to tackle the partial annotation problem to extract *more* answer spans from existing MRC datasets for better question generation and question answering. Given that the fine-grained annotation of answer spans requires significant expert labor, it leads to the first challenge in the *training stage*: a) How can we exploit the existing MRC datasets to extract more high-quality answer spans with partially annotated data? After obtaining a robust model that can extract more spans from paragraphs, the results will include newly extracted spans without annotation and further lead to the second challenge in the *evaluation stage*: b) How can we automatically evaluate newly extracted spans without the help of ground truth?

To solve the challenges above, we revisit AE in both training and evaluation stages. As a key perspective, we formulate AE as a *Positive-unlabeled (PU) learning* problem [9] to solve the first challenge. This roots in our observation in Section 3 that MRC datasets consist of annotated answer spans which are *positive* samples and others which are *unlabeled* ones. We try to re-estimate the risk of unlabeled data with a class prior under PU learning framework. To provide more informative contextual representations for PU classifier, we augment the understanding of textual paragraph by modeling its inherent structure. Specially, we construct a graph with both *syntactic edges* and *semantic edges* for information propagation. Finally, a variant pointer network, namely *unified pointer*, is proposed to classify each candidate span.

For the second challenge, we propose an automatic metric, i.e. *question-worthy* score, to evaluate whether a span is worthy of asking. We empirically disentangle this problem into the measurement of *questionability* and *worthiness*. The first property aims to evaluate whether a span can be asked by humans and is calculated by the question answering system assisted with a question generation model. Meanwhile, a meaningful question tends to probe key information located in salient sentences from paragraph. We, therefore, design the second property by utilizing text summarization techniques that measure the salience of sentences.

To summarize, the contributions of this paper include:

- We investigate the partial annotation problem in AE task and propose SCOPE, a graph-based method under PU learning framework to solve it.
- We decompose the question-worthiness of answers into questionability and worthiness, and propose an automatic evaluation metric to assess them.
- The extracted answer spans, when used as augmented data, successfully boost downstream MRC task, which also demonstrates the effectiveness of SCOPE.

2 RELATED WORK

Answer Span Extraction. The basic idea behind AE is highly related to information extraction tasks (i.e., Semantic Role Labeling (SRL) [17], Open Information Extraction (OpenIE) [39], etc.). Different from SRL and OpenIE that focus on identifying arguments for a central verb, AE utilizes a verb-free extraction paradigm, extracting entities, numbers, or other key information from paragraphs that can be asked. Previous work on AE mainly takes the sequence labeling [5, 6] or span selection [32] setting to identify answer spans. By regarding each semantic chunk in paragraphs as candidate spans, others also try to find answer spans through syntactic rules [4, 26] or sampling from a joint distribution [20]. However, when most research endeavors focus on designing sophisticated architecture, we point out the supervision they rely on is far from sufficient and will hinder the generalization of models. In this paper, we take a further step to analyze the partial annotation problem in AE task.

Positive-unlabeled Learning. PU learning tries to learn an unbiased binary classifier from only positive and unlabeled data. This setting is very common in the real scenario where labeling all positive data requires significant expert labor [25] or negative samples are hard to obtain [35]. Existing PU learning methods can be roughly divided into two categories: a) One tries to select high-quality negative samples from unlabeled data as possible negative data [19]. b) The other proposes to use modified risk estimators and re-weight the empirical risk for unbiased learning [8, 9, 16]. In this paper, we mainly focus on the second branch to learn an unbiased binary classifier. Different from conventional supervised or semi-supervised learning where labeled samples are provided for each class, PU learning only has labeled samples as positive data and others as unlabeled data.

3 EXPLORE THE PARTIAL ANNOTATION PROBLEM

Since there is no specific dataset for AE, a common way to train the AE model is to use the annotation from MRC datasets. In this section, we first formalize the AE task under MRC setting in Section 3.1. Then we conduct a dataset-dependent analysis in Section 3.2 on two well-known MRC datasets, i.e., SQuAD [29] and DROP [10], to show the limitation of MRC datasets.

3.1 Problem Definition

Given a paragraph $\mathcal{P} = \{t_1, t_2, \cdots, t_N\}$ with N tokens, the goal of AE is to identify a set of answer spans $\mathcal{S} = \{s_i\}$ against the candidate span set $\mathcal{S}^*$. Generally, $|\mathcal{S}^*|$ contains all possible text fragments $t_{i:j}$ from the original paragraph where $i \leq j$.

Under MRC setting, we have paragraph $\mathcal{P}$ paired with a set of QA pairs $\{q_i, a_i\}$, where q_i is the question and a_i is the corresponding answer. Previous work uses answers in $\{q_i, a_i\}$ to construct *positive* data $\mathcal{S}_p$ for AE where each $s_i \in \mathcal{S}_p$ corresponds to an answer a_i. The main backward of this setting is that many answer spans are not annotated. Directly taking the *unlabeled* data $\mathcal{S}_u = \mathcal{S}^* \backslash \mathcal{S}_p$ as *negative* samples and conducting the conventional training will lead to the wrong classification boundary [9]. Therefore, our goal is to find the optimal classifier $f_{\theta^*}(\cdot)$ with parameter θ^* for partially

Table 1: Results of re-annotation on both datasets. γ indicates the proportion of answers that are not annotated by the original dataset.

Dataset	#Sentences	$\lvert\mathcal{S}_p\rvert$	$\lvert\mathcal{M}\rvert$	γ
SQuAD	237	219	207	48.59%
DROP	445	296	492	62.44%

annotated data $\mathcal{S}^*$ as follows:

$$\theta^* = \arg\max_{\theta} \sum_{s \in \mathcal{S}^*} \log P(y^*|s;\theta), \tag{1}$$

where y^* denotes the ground truth label to s.

3.2 Dataset-dependent Analysis

To better explore the partial annotation problem, we narrow down the definition of answer span to factoid information in paragraphs (e.g., entities, dates, values, etc.) that makes up more than 90% of answer spans from SQuAD and DROP [1]. We take the following re-annotation procedure: We first randomly pick 50 paragraphs from each dataset. Two annotators are asked to annotate answer spans for 25 paragraphs respectively. Our annotation guidelines encourage annotators to ask similar types of information missed by the original dataset, like the unlabeled entity and dates in the second example shown in Figure 1. For spans mentioned multiple times in one paragraph, we annotate the first appearance if it can be asked, since it brings more information to the reader. Then two annotators re-check each other's results for agreement [2]. If they have conflicts, we will send these samples to the third annotator for final judgments.

Given a set of spans $\mathcal{S}$, we define the missing rate γ as follows:

$$\gamma = \frac{|\mathcal{M}|}{|\mathcal{S}_p \cup \mathcal{M}|}, \tag{2}$$

where $\mathcal{S}_p$ is the set of positive spans and $\mathcal{M}$ is the set of answer spans not given in $\mathcal{S}$. We do not consider negative spans $\mathcal{S}_n$ in calculation mainly because γ plays a role similar to the Recall. Note that when $\mathcal{S}$ is the original golden span set from datasets, it contains only $\mathcal{S}_p$ and $|\mathcal{M}|$ is equal to the number of newly annotated spans.

As shown in Table 1, we annotated 237 (445) sentences for SQuAD (DROP) and the missing rate reaches 48.59% (62.44%). This indicates that there are a comparable number of positive answer spans not annotated among unlabeled candidate spans, suggesting that the entire dataset contains *positive* and *unlabeled* data. In this scenario, conventional supervised training methods are not satiable since they heavily rely on well-annotated data. With this in mind, we formulate the AE task as a PU learning problem and propose SCOPE to solve the partial annotation problem in AE. Meanwhile, it is hard to judge whether a single answer span is worthy of asking since it may be unlabeled. To better distinguish noisy spans from answer spans without annotation, we further introduce an automatic metric for evaluating the question-worthiness nature of given spans.

[1]DROP contains three types of answers and we only consider the extractive examples where the answer is a span from the original paragraph.
[2]Our agreement of first two annotators is 89.37% for SQuAD and 76.83% for DROP.

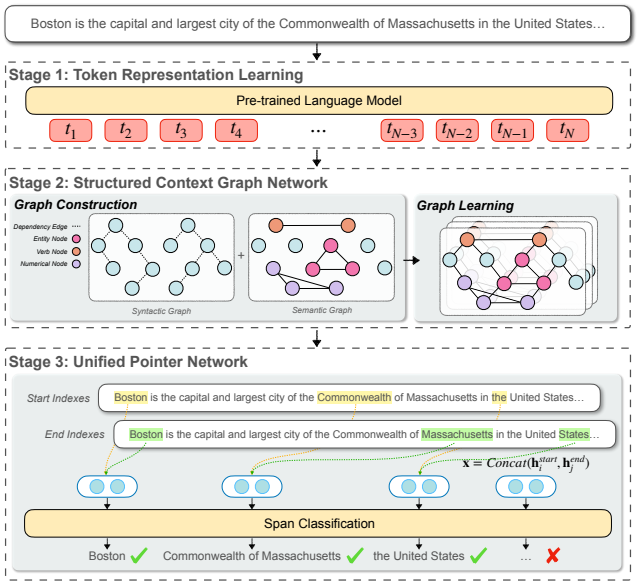

Figure 2: An overview of SCOPE. Taking a paragraph as input, its representation is first obtained by the PLM. Then we use GAT to propagate information over context graph (self-loop edges are omitted for brevity). Finally, a unified pointer network is employed to identify answer spans. The entire model is trained in an end-to-end fashion under PU learning framework.

4 METHOD

In this section, we first present the overview of SCOPE and detail its components in Section 4.1- 4.3. Then the automatic metric is introduced in Section 4.5.

The overview of SCOPE is outlined in Figure 2. For a given paragraph, token representation learning module first obtains its contextual embedding by Pre-trained Language Models (PLMs), e.g., BERT [3] or RoBERTa [22]. Then a structured context graph is constructed with both syntactic and semantic edges. We apply Graph Attention Network (GAT) [33] to iteratively propagate information between different key components. The learned representations are then fed into a variant pointer network, namely unified pointer, to identify target spans by matching start and end indexes. Finally, we optimize the proposed model under the PU learning framework to alleviate the bias introduced by annotation.

4.1 Token Representation Learning

Token representation learning module aims to obtain the initial representations of input tokens. To be in line with backbone PLM, the given paragraph $\mathcal{P}$ is first tokenized into sub-tokens $\{e_1, e_2, \cdots, e_{N'}\}$. The PLM receives the sub-token sequence and outputs the contextual representation matrix $\mathbf{E}' = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{N'}\} \in \mathbb{R}^{N' \times d_1}$, where d_1 is the hidden size. We then reconstruct representation for each token t_i by averaging sub-token representations as $\mathbf{t}_i = \text{Average}_j(\mathbf{e}_j)$ where $e_j \in t_i$, and the final token representations can be denoted as $\mathbf{E} = \{\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N\} \in \mathbb{R}^{N \times d_1}$.

4.2 Structured Context Graph Network

When PLMs mainly capture the contextual information from paragraph, we further enhance the structural information by introducing the structured context graph network.

Graph Construction. We construct the structured context graph following two categories of syntactic and semantic edges. We first build *syntactic edges* with the dependency parsing tree due to the promising results achieved in many information extraction tasks [14, 30]. We encourage the model to reveal the intrinsic structure in each sentence with syntactic edges because a valid span prefers to form a sub-tree. It can also effectively capture the dependency relation between entities and central verbs while filtering out irrelevant information. However, syntactic edges lack the information exchange between sentences and are insensitive to what is worth asking. To tackle this problem, we further enrich the graph with *semantic edges* using heuristic rules that stem from the associations between different key elements. It is motivated by the observation that humans tend to ask different types of questions like "*What happened to whom, when, and where*". Specifically, we extract the verbs (what), locations (where), times (when), and other noun phrases (whom) based on named entity recognition and part-of-speech results. Then, tokens that have the same type are fully connected to form a sub-graph. After constructing both syntactic and semantic edges, we merge them for the final graph learning. We also introduce a self-loop edge to every node because it includes the node information itself in the message propagation process.

Graph Learning. Let $\mathbf{H}^l = \{\mathbf{h}_1^l, \mathbf{h}_2^l, \cdots, \mathbf{h}_N^l\}$ denote the node representation in l^{th} layer. We first initialize $\mathbf{H}^0$ with token representation $\mathbf{E}$, then apply GAT on structured context graph. Specifically, for each linked node pair i and j, we compute the attention weight as follows:

$$\begin{aligned} \alpha_{i,j}^l &= \text{softmax}_j(\mathbf{e}_{i,j}^l) = \frac{\exp(\mathbf{e}_{i,j}^l)}{\sum_{k\in\mathcal{N}_i}\exp(\mathbf{e}_{i,k}^l)}, \\ \mathbf{e}_{i,j}^l &= f^l(\mathbf{W}^l\mathbf{h}_i^l, \mathbf{W}^l\mathbf{h}_j^l), \end{aligned} \tag{3}$$

where $\mathbf{e}_{i,j}^l$ denotes the attention coefficient between two linked nodes which is further normalized into $\alpha_{i,j}^l$, $\mathcal{N}_i$ are the neighborhoods of node i, $\mathbf{W}^l \in \mathbb{R}^{d_2\times d_1}$ is learned parameter, $f^l : \mathbb{R}^{d_2}\times\mathbb{R}^{d_2} \to \mathbb{R}$ is a linear transformation.

After obtaining the attention weights, we update the node representations for next layer as follows:

$$\mathbf{h}_i^{l+1} = \sum_{j\in\mathcal{N}_i} \alpha_{i,j}\mathbf{h}_j^l. \tag{4}$$

We stack L layers of GAT to capture structural relations from multi-hop neighbors. The final hidden states $\mathbf{H}^L = \{\mathbf{h}_1^L, \mathbf{h}_2^L, \cdots, \mathbf{h}_N^L\}$ are then feed into a unified pointer network to extract answer spans. For a clearer description, we simplify $\mathbf{H}^L$ as $\mathbf{H}$ in later sections.

4.3 Unified Pointer Network

There are two main components determining a span in a paragraph, i.e., the start and end indexes. A straightway for span identification is to apply a pointer network with two classifiers separately [31]. This strategy suffers from the weakness of softmax function that only one span can be predicted with the highest probability.

In this section, we adopt a simple yet effective variant pointer network, namely unified pointer, to select spans from paragraph. Different from identifying the start and end tokens in pointer network, unified pointer network represents each candidate span uniformly with its start and end token representations. It provides a more fine-grained span representation and can extract multiple spans without a manually assigned threshold.

To get the start and end representations, we first apply a transformation on $\mathbf{H}$:

$$\mathbf{H}^{start} = \mathbf{W}^{start}\mathbf{H}, \quad \mathbf{H}^{end} = \mathbf{W}^{end}\mathbf{H}, \tag{5}$$

where $\mathbf{W}^{start}, \mathbf{W}^{end} \in \mathbb{R}^{d_3\times d_1}$ are learned parameters.

For any span s starting from i^{th} token and ending at j^{th} token, a binary classifier is applied to predict the final results:

$$\hat{y} = \text{softmax}(\mathbf{W}^{span}\mathbf{x}), \quad \mathbf{x} = [\mathbf{H}_i^{start}; \mathbf{H}_j^{end}], \tag{6}$$

where $\mathbf{W}^{span} \in \mathbb{R}^{2\times(2*d_3)}$ is learned parameter shared by different spans, and $[\cdot;\cdot]$ is the concatenation operation of two vectors.

4.4 Positive-unlabeled Learning Method

One challenge for training the AE model with MRC datasets is that there are lots of answer spans without annotation in $\mathcal{S}^*$. Directly regarding all unlabeled candidate spans $\mathcal{S}_u$ as negative data $\mathcal{S}_n$ will lead to a biased estimation of training loss. In this section, we try to solve the partial annotation problem with non-negative Positive-unlabeled learning (nnPU) [16]. We first review the overall idea of nnPU, and then apply it to estimate the unbiased training loss.

Non-negative Risk Estimator. Let $\mathbf{x} \in \mathcal{X}$ be the representation of span from paragraph obtained by Eq. 6 and $y \in \mathcal{Y}$ be the corresponding label, where $\mathcal{X} \subset \mathbb{R}^{2*d_3}$ and $\mathcal{Y} \subset \{0, 1\}$. We denote the entire classification framework as $f : \mathbb{R}^{2*d_3} \to \mathbb{R}^2$ and loss function as $\ell : \mathbb{R}\times\mathcal{Y} \to \mathbb{R}$. Then the risk of any classifier f can be represented as $R_\ell = \mathbb{E}_{\mathbf{x},y}\ell(f(\mathbf{x}), y)$.

By splitting the risk into two parts, one for positive samples and the other for negative samples, we get the reformulated R_ℓ as follows:

$$R_\ell = \pi\mathbb{E}_{\mathbf{x},y=1}\ell(f(\mathbf{x}), 1) + (1-\pi)\mathbb{E}_{\mathbf{x},y=0}\ell(f(\mathbf{x}), 0), \tag{7}$$

where $\pi = P(y = 1)$ is the prior distribution of positive samples in the dataset, $\mathbb{E}_{\mathbf{x},y=1}$ and $\mathbb{E}_{\mathbf{x},y=0}$ are the expectation of positive and negative samples.

Recall that we only have a labeled set of positive samples $\mathcal{S}_p$ and an unlabeled set $\mathcal{S}_u$. In this scenario, the main problem becomes how to estimate $\mathbb{E}_{\mathbf{x},y=0}\ell(f(\mathbf{x}), 0)$ without true negative samples. Since it always holds that $P(y = 0)P(\mathbf{x}|y = 0) = P(\mathbf{x}) - P(y = 1)P(\mathbf{x}|y = 1)$, the expectation of negative samples can be reformulated as follows:

$$(1-\pi)\mathbb{E}_{\mathbf{x},y=0}\ell(f(\mathbf{x}), 0) = \mathbb{E}_x\ell(f(\mathbf{x}), 0) - \pi\mathbb{E}_{x,y=1}\ell(f(\mathbf{x}, 0)), \tag{8}$$

where $\mathbb{E}_x$ denotes the expectation of unlabeled samples.

Based on Eq. 7 and Eq. 8, Kiryo et al. [16] alleviate the overfitting problem with a lower-bound as follows:

$$\begin{aligned}\tilde{R}_\ell &= \pi R_p^+ + \max(0, R_u^- - \pi R_p^-),\\ R_p^+ &= \mathbb{E}_{x,y=1}\ell(f(\mathbf{x}), 1),\\ R_u^- &= \mathbb{E}_x\ell(f(\mathbf{x}), 0),\\ R_p^- &= \mathbb{E}_{x,y=1}\ell(f(\mathbf{x}), 0).\end{aligned} \tag{9}$$

This provides an alternating way to estimate risk $\tilde{R}_\ell$ with only positive and unlabeled data.

PU Objective Function. For each candidate span, we have the final prediction $\hat{y}$ over two classes with Eq. 6. We use the binary cross-entropy as our loss function $\ell = \text{CrossEntropy}(y, \hat{y})$ where y is the label from partially annotated dataset. The final empirical loss is calculated with positive and unlabeled samples as follows:

$$\begin{aligned}\tilde{R}_\ell &= \pi\hat{R}_p^+ + \max(0, \hat{R}_u^- - \pi\hat{R}_p^-),\\ \hat{R}_p^+ &= \frac{1}{|\mathcal{X}_p|}\sum_{\mathbf{x}\in\mathcal{X}_p}\ell(f(\mathbf{x}, 1)),\\ \hat{R}_u^- &= \frac{1}{|\mathcal{X}_u|}\sum_{\mathbf{x}\in\mathcal{X}_u}\ell(f(\mathbf{x}, 0)),\\ \hat{R}_p^- &= \frac{1}{|\mathcal{X}_p|}\sum_{\mathbf{x}\in\mathcal{X}_p}\ell(f(\mathbf{x}, 0)),\end{aligned} \tag{10}$$

where $\mathcal{X}_p$ and $\mathcal{X}_u$ are representations of positive samples and unlabeled samples respectively.

The model is trained in an end-to-end fashion by minimizing $\tilde{R}_\ell$. To calculate the risk term, we only have to estimate the prior distribution π of all positive samples. This can be easily calculated as follows: a) We first obtain the distribution of annotated positive samples in the original dataset as π'. b) Then, we calculate the missing rate γ defined in Section 3.2 with a few annotations. c) The π can be estimated with $\frac{\pi'}{1-\gamma}$.

4.5 Evaluation Framework

Since conventional metrics only measure the alignment of extracted spans with annotated spans, spans without annotation can not be evaluated automatically. This problem is more pronounced on partially annotated datasets. We, therefore, propose an automatic metric to measure the question-worthiness nature of spans from the intuition of humans.

Studies in psychology show that humans tend to ask questions for factoid information [23] and will "select answers that are informative about inferred interests" [12]. These two points of view motivate us to measure an answer span from two perspectives: 1) *questionability* 2) *worthiness.*

Principle 1. *If a span is **askable**, there exists at least one question that can be answered by this span with a high probability.*

Questionability reflects whether humans can ask questions for the given factoid information. Intuitively, this is the basic characteristic of an answer span that decides its usage in downstream tasks. To measure the questionability, we reduce it to a Question Generation and Question Answering (QGQA) framework, which serves as dual tasks in many areas [11, 38].

The QG model first takes the given span s and its paragraph as input, and generates corresponding question for it. Then we pair the question with paragraph and send it into a QA model. The QA model scores each token with a start score v_{start} and an end score v_{end}. These two scores indicate the probability of token being start or end of the answer. The *questionability* score of the given span is then calculated as follows:

$$score_q(s) = \frac{1}{2} * \left(v_{start}(i) + v_{end}(j)\right), \tag{11}$$

where i and j are the start and end positions of given span in paragraph. In practice, we generate multiple questions with beam search to alleviate the noise introduced by each module, and take the highest score as the final score.

Principle 2. *If a span is **worthy** of asking, it contains more information for people to ask a question.*

We design the worthiness of a span to evaluate how important it is in the paragraph. However, it is hard to automatically decide the worthiness without human involvement. As a way of expressing information, we suppose a meaningful question tends to ask the key information located in salient sentences. This point of view shares the same idea of sentence salience in extractive summarization [36]. Therefore, we use the summarization score of the context to measure the coarse-grained worthiness of answer spans. Specifically, we run an extractive summarization model to assign all sentences a salience score $v_{salience} \in [0, 1]$. For any given span from paragraph, we define the *worthiness* score as follows:

$$score_w(s) = v_{salience}(C_s), \tag{12}$$

where C_s is the sentence in which given span s located.

Question-worthy Score. Based on $score_q$ and $score_w$, our *question-worthy score* ϵ of given span s is a simple fusion of these two scores as follows:

$$\epsilon(s) = \frac{1}{2} * \left(score_q(s) + score_w(s)\right). \tag{13}$$

5 EXPERIMENTS

5.1 Datasets and Compared Methods

Datasets. We conduct experiments on two well-known MRC datasets, namely SQuAD [29] and DROP [10]. Since both datasets have a blind test set, we split the public available portion into "train", "development" and "test". For SQuAD, we use the widely adopted three-way split released by Du et al. [7]. For DROP, we randomly select a test set from the training set, and the test set remains the same size as the development set. More details can be found in Appendix A.

Compared Methods. We mainly compare SCOPE against several published state-of-the-art baselines, which can be divided into three types, i.e., classification-based, tagging-based, and pointer-based methods. To train the classification-based methods, we first use Stanza [27] to extract all named entities from paragraph as candidate spans (denoted as "ENT"), then build a classifier to filter out noisy spans.

- ENT: which contains only named entities in paragraphs extracted by the off-the-shelf NLP toolkit.

Table 2: Overall performance on conventional metrics. "Avg. spans" indicates the average number of spans per paragraph extracted by different methods.

Model	SQuAD				DROP			
	Precision	Recall	F1	Avg. spans	Precision	Recall	F1	Avg. spans
ENT	*13.63*	*40.41*	*20.39*	*12.82*	*6.31*	*52.58*	*11.27*	*43.90*
ENT Classifier (BERT$_{base}$)	**48.55**	20.37	28.70	1.81	36.90	19.10	25.17	2.73
⌞ (SpanBERT$_{base}$)	47.90	21.09	29.29	1.90	**38.62**	20.52	26.80	2.80
⌞ (RoBERTa$_{base}$)	47.57	20.61	28.76	1.87	35.54	20.90	26.32	3.10
Sequence Tagger (BERT$_{base}$)	44.39	25.96	32.76	2.53	30.07	20.60	24.45	3.61
⌞ (SpanBERT$_{base}$)	46.96	25.98	33.45	2.39	35.24	20.52	25.94	3.07
⌞ (RoBERTa$_{base}$)	48.43	25.19	33.14	2.25	34.91	19.33	24.88	2.92
Boundary-aware NER (Zheng et al., 2019)	32.80	18.67	23.79	2.46	34.40	7.23	11.95	1.11
BiFlaG (Luo and Zhao, 2020)	36.50	25.97	30.35	3.08	38.13	20.03	26.27	2.77
MRC NER (BERT$_{base}$) (Li et al., 2020)	37.71	25.84	30.67	2.96	29.53	21.20	24.68	3.78
⌞ (SpanBERT$_{base}$)	37.04	27.94	31.85	3.26	31.79	20.52	24.94	3.40
⌞ (RoBERTa$_{base}$)	39.59	27.39	32.38	2.99	32.38	22.43	26.50	3.65
SCOPE (BERT$_{base}$)	36.96	39.99	38.41	4.68	30.74	32.32	31.51	5.54
⌞ (SpanBERT$_{base}$)	41.15	39.70	**40.41**	4.17	33.95	35.84	34.87	5.56
⌞ (RoBERTa$_{base}$)	36.10	**45.19**	40.14	**5.41**	33.51	**37.08**	**35.21**	**5.83**

- ENT Classifier: which is a classification-based method that uses a classifier built upon the results of "ENT".
- Sequence Tagger: which is a tagging-based method that views span extraction as a sequence labeling task.
- Boundary-aware NER [41]: which is a pointer-based method that contains both boundary detection module and label prediction module.
- BiFlaG [24]: which is a pointer-based method that constructs an entity graph and an adjacent graph for nested spans extraction.
- MRC NER [18]: which is a pointer-based method that extracts spans by answering corresponding questions.

5.2 Performance on Conventional Metrics

Table 2 reports the overall performance on conventional metrics. It's worth mentioning that even though the original datasets are partially annotated, the performance on conventional metrics still reflects the quantitative alignment between extracted results and the original annotation, especially in Recall. Generally, SCOPE significantly outperforms other baselines by a large margin, achieving 40.41 F1 with SpanBERT$_{base}$ on SQuAD and 35.21 F1 with RoBERTa$_{base}$ on DROP. This performance gain is mainly due to the improvement in Recall, which confirms the ability of SCOPE to extract more high-quality answer spans.

From the detailed comparisons between SCOPE and other methods, we can find that "Sequence Tagger" provides a strong baseline on SQuAD but has a serious drop on small dataset DROP. As a compromise, SCOPE has a consistent performance on both datasets. As an off-the-shelf method, "ENT" provides a relatively high Recall. However, it fails to identify noise spans because not all entities are informative enough for asking a question. Without considering the partial annotation problem introduced by MRC datasets, "ENT Classifier" suffers from false negative samples and fail to identify more answer spans. The low Recall and the small number of spans extracted by baseline models will further limit their application in downstream tasks.

Table 3: Overall performance on proposed metric. For all PLM-based methods, we choose the RoBERTa$_{base}$ implementation for a fair comparison. We report both the average score and corresponding standard deviation (in braces).

Model	Avg. $score_q$	Avg. $score_w$	Avg. ϵ
Golden	*80.07 (0.21)*	*38.09 (0.13)*	*59.08 (0.13)*
ENT Classifier	**78.52** (0.21)	32.93 (0.13)	55.72 (0.12)
Sequence Tagger	74.34 (0.21)	35.66 (0.12)	55.00 (0.12)
Boundary-aware NER	70.02 (0.25)	35.30 (0.13)	52.66 (0.14)
BiFlaG	75.95 (0.22)	34.83 (0.13)	55.39 (0.13)
MRC NER	75.07 (0.21)	**36.09** (0.12)	55.58 (0.12)
SCOPE	76.94 (0.21)	35.60 (0.12)	**56.27** (0.12)

5.3 Performance on Question-worthy Score

In order to measure the generalization ability of models in extracting more answer spans, we study the quality of newly extracted spans with proposed question-worthy score. The results are summarized in Table 3. As expected, the question-worthy score on golden spans outperforms all model outputs, indicating that annotated answer spans are still high-quality in the perspective of question-ability and worthiness. This is mainly because humans can fully consider the context of an answer span when they are reading the paragraph. Combined with the results in Table 2, we can find that SCOPE achieves the best ϵ when extracting more answer spans compared to other baselines. This is in line with our original intention of introducing PU learning to extract more answer spans from paragraph without degrading the quality of the results. Note that "ENT Classifier" achieves the highest $score_q$ among all methods. We suppose this is mainly due to the advantage of off-the-shelf NLP

Table 4: Overall performance on data argumentation. For all PLM-based methods, we choose the RoBERTa$_{base}$ implementation for a fair comparison

Backbone Model	Exact Match	F1
BERT$_{large}$ (Devlin et al., 2019)	78.7	81.9
BERT$_{large}$ (Our implementation)	77.9	81.3
⌞ ENT Classifier	77.8 (−0.1)	81.1 (−0.2)
⌞ Sequence Tagger	79.0 (+1.1)	82.2 (+0.9)
⌞ Boundary-aware NER	77.3 (−0.6)	80.8 (−0.5)
⌞ BiFlaG	79.1 (+1.2)	82.1 (+0.8)
⌞ MRC NER	79.3 (+1.4)	82.6 (+1.3)
⌞ SCOPE	**79.9 (+2.0)**	**83.2 (+1.9)**

Table 5: Ablation study of different modules of SCOPE.

Model	Precision	Recall	F1
SCOPE (RoBERTa$_{base}$)	36.10	**45.19**	**40.14**
w/o syntactic edges	39.94	39.37	39.65
w/o semantic edges	41.16	37.89	39.46
w/o PU learning	**48.55**	28.72	36.09

tools that can extract candidate spans with clear boundaries. However, many entities are misclassified and it leads to a low $score_w$ for "ENT Classifier". We also find that "Boundary-aware NER" fails to correctly identify answer boundaries as expected, resulting in poor performance on $score_q$.

5.4 Performance on Data Augmentation for MRC

Another way to evaluate the extracted results is to see whether they can boost the performance of downstream tasks. Therefore, we analyze how SCOPE could effectively augment the data for MRC tasks in this section. We conduct experiments on SQuAD 2.0 [28]. For all methods, we first obtain the augmented QA pairs by running inference on training data. Then we train a BERT-based MRC model and report the performance on development set.

The overall results are demonstrated in Table 4. SCOPE outperforms all baselines, which brings about 2.0 performance gain on EM and 1.9 performance gain on F1. It demonstrates the question-answer pairs extracted from paragraphs are of high quality. In contrast, "ENT Classifier" and "Boundary-aware NER" hurt the results due to noise spans.

5.5 Analysis

To better understand the strengths and limitations of SCOPE and question-worthy score, we further analyze the results of both. All analyses are conducted on SQuAD dataset.

Influence of Different Modules. To analyze the influence brought by different modules, we conduct ablation studies on both structured context graph and PU learning framework. The results are reported in Table 5. We can find that our final model outperforms both SCOPE (w/o syntactic edges) and SCOPE (w/o semantic edges).

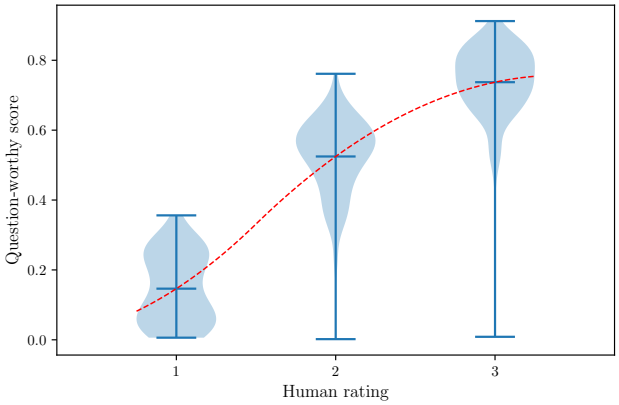

Figure 3: Question-worthy score against human rating. The red line goes through the mean scores of each category.

This indicates the ability of structured context graph to integrate features from both syntactic and semantic edges. When we remove the PU learning framework, the biased learning objective misleads the model in the training stage, resulting in a low Recall.

Influence of Prior Distribution. As a key component of PU learning framework, the estimation of π may influence the final results. To show the robustness of SCOPE, we conduct experiments under different settings of π. We estimate π with different scales vary from 1.50 to 2.50 and present the results in Table 6. Recall that π controls the number of ground distribution of positive samples. Therefore, when we increase the scale factor, the number of extracted spans also increases. Based on the annotation in Section 3.2, our default estimation of scale factor equals 2.00 shows the best performance. Furthermore, SCOPE has a consistent performance gain with different backbone PLMs and prior π, suggesting that we do not need significant expert labor to do a fine-grained re-annotation over a large amount of data.

Effect of PU Learning. We further analyze the results of SCOPE with and without PU learning framework to show how PU learning influences the extracted results. We manually annotate 50 paragraphs from model outputs as Section 3.2 did [3]. As illustrated in Table 7, SCOPE yields almost twice the number of spans with PU learning. The results also share a high correlation with human intuition which leads to a significant decrease in missing rate. Note that the model can reduce the missing rate by extracting as many spans as possible without considering the question-worthiness nature of spans. We therefore report the Accuracy of the extracted answer spans. The performance boost on Accuracy also indicates that the decrease in missing rate is achieved by extracting more high-quality answer spans.

Human Correlation. To find out how the proposed metric correlates with humans, we manually divide 555 extracted spans from 100 paragraphs into three types: 1) Span with no sense or can not be asked. 2) Answer span that can be asked but is meaningless or with bad boundaries. 3) Answer span that is worthy of asking.

[3] Our agreement of first two annotators is 86.67%.

Table 6: Ablation study of π that controls the risk estimator on SQuAD. With the increase of scale factor, model extracts more answer spans from paragraphs. † denotes the default estimation in our previous experiments.

	BERT$_{base}$				SpanBERT$_{base}$				RoBERTa$_{base}$			
	Precision	Recall	F1	Avg. spans	Precision	Recall	F1	Avg. spans	Precision	Recall	F1	Avg. spans
$\pi' \times 1.50$	**39.93**	35.02	37.31	3.79	**44.67**	35.33	39.46	3.42	**42.18**	36.83	39.32	3.78
$\pi' \times 1.75$	39.21	36.25	37.67	4.00	41.77	38.32	39.97	3.97	40.28	38.62	39.43	4.15
$\pi' \times 2.00$†	36.96	39.99	**38.41**	4.68	41.15	39.70	**40.41**	4.17	36.10	45.19	**40.14**	5.41
$\pi' \times 2.25$	29.32	47.65	36.30	7.03	37.99	43.04	40.36	4.90	33.31	**47.63**	39.20	6.18
$\pi' \times 2.50$	29.41	**49.17**	36.81	7.23	34.39	**46.64**	39.59	5.86	32.18	47.06	38.22	6.32

Table 7: Human evaluation on model outputs.

Model	#Spans	γ	Accuracy
Golden	*212*	*47.91*	*N/A*
SCOPE (RoBERTa$_{base}$)	**345**	**41.77**	**88.12**
w/o PU	186	63.64	83.33

We report the question-worthy score against human rating in Figure 3. The Pearson correlation coefficient [1] between human ratings and the proposed metric is 0.8308 with $p < 0.01$, which shows high a correlation. From the detailed analysis, we can find it is easy to distinguish bad answer spans from good answer spans (type 1 vs type 2 & 3) for both human and proposed metric. But it is even hard for a human to clearly define what is more worthy of asking (type 2 vs type 3). Besides, there are some extreme cases that result in very low scores. This is mainly because the proposed metric is based on learning methods that are sensitive to span boundaries. We will further show the bad cases in the case study.

Case Study. We present some examples in Figure 4, which show the effectiveness and weakness of our framework. From paragraph #1, we can find our model greatly recalls answer spans from paragraph beyond annotation. Similar to the golden annotation, the newly extracted answer spans focus on information about magazines that cover the main content of the entire paragraph. However, since it is a preliminary study on question-worthy nature of answer spans, our proposed metric is still sensitive to answer boundaries as shown in paragraph #2. It gives "more than 2,300" a low $score_q$ because QA model tries to answer the question with answer "2,300".

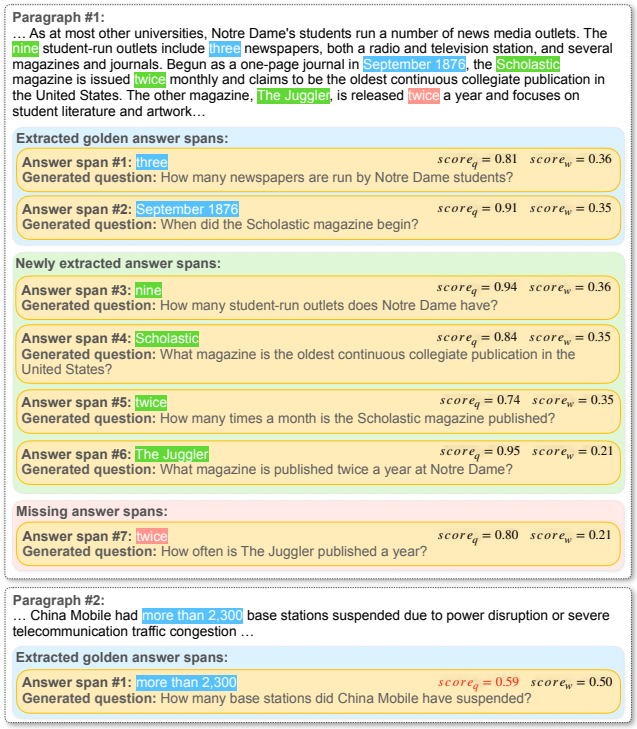

Figure 4: Case study of SCOPE and proposed metric. Blue spans are golden spans that are correctly extracted. Green spans are results that are newly extracted beyond annotation. Red spans are results that fail to identify.

6 CONCLUSION

In this paper, we revisit the problem of answer span extraction in both modeling and evaluation stages. Based on our analysis, we reveal the annotation problem in answer span extraction task may lead to a biased model. We propose SCOPE, a graph-based method under positive-unlabeled learning framework, to bridge this gap. SCOPE achieves significant performance gain which can recall more answer spans from paragraphs. To better explore the nature of question-worthy answer spans without human labor, we propose an automatic metric, i.e. question-worthy score, from the perspective of question answering and text summarization. In future work, we will explore the possibility of combining the extraction model with the automatic metric in a unified framework. Besides, current solution on question-worthy score is still a preliminary study. We will try to propose a more fine-grained metric to better evaluate the nature of high-quality answer spans.

ACKNOWLEDGMENTS

We thank anonymous reviewers from current and past versions of the manuscript for their comments and suggestions. This work was supported by National Key Research and Development Project (No.2020AAA0109302), Shanghai Science and Technology Innovation Action Plan (No.19511120400) and Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103).

REFERENCES

[1] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer, 1–4.

[2] Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the Growth: Difficulty-Controllable Question Generation through Step-by-Step Rewriting. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 5968–5978.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.

[4] Kaustubh Dhole and Christopher D Manning. 2020. Syn-QG: Syntactic and Shallow Semantic Rules for Question Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 752–765.

[5] Xinya Du and Claire Cardie. 2017. Identifying where to focus in reading comprehension for neural question generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2067–2073.

[6] Xinya Du and Claire Cardie. 2018. Harvesting Paragraph-level Question-Answer Pairs from Wikipedia. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1907–1917.

[7] Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to Ask: Neural Question Generation for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1342–1352.

[8] Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. 2015. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*. PMLR, 1386–1394.

[9] Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of learning from positive and unlabeled data. *Advances in neural information processing systems* 27, 703–711.

[10] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 2368–2378.

[11] Wee Chung Gan and Hwee Tou Ng. 2019. Improving the robustness of question answering systems to question paraphrasing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 6065–6075.

[12] Robert D. Hawkins, Andreas Stuhlmüller, Judith Degen, and Noah D. Goodman. 2015. Why do you ask? Good questions provoke informative answers. *Cognitive Science* (2015).

[13] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems* 28, 1693–1701.

[14] Zhanming Jie, Aldrian Muis, and Wei Lu. 2017. Efficient dependency-guided named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 31.

[15] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations'*.

[16] Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 1674–1684.

[17] Ayal Klein, Jonathan Mamou, Valentina Pyatkin, Daniela Stepanov, Hangfeng He, Dan Roth, Luke Zettlemoyer, and Ido Dagan. 2020. QANom: Question-Answer driven SRL for Nominalizations. In *Proceedings of the 28th International Conference on Computational Linguistics*. 3069–3083.

[18] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A Unified MRC Framework for Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 5849–5859.

[19] Xiaoli Li and Bing Liu. 2003. Learning to classify texts using positive and unlabeled data. In *Proceedings of the 18th international joint conference on Artificial intelligence*. 587–592.

[20] Bang Liu, Haojie Wei, Di Niu, Haolan Chen, and Yancheng He. 2020. Asking questions the human way: Scalable question-answer generation from text corpus. (2020), 2032–2043.

[21] Yang Liu and Mirella Lapata. 2019. Text Summarization with Pretrained Encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 3730–3740.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).

[23] George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin* 116, 1 (1994), 75.

[24] Ying Luo and Hai Zhao. 2020. Bipartite Flat-Graph Network for Nested Named Entity Recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6408–6418.

[25] Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2019. Distantly Supervised Named Entity Recognition using Positive-Unlabeled Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2409–2419.

[26] Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training Question Answering Models from Synthetic Data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 5811–5826.

[27] Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 101–108.

[28] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 784–789.

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2383–2392.

[30] Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence Relation Extraction with Document-level Graph Convolutional Neural Network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4309–4316.

[31] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional Attention Flow for Machine Comprehension. In *International Conference on Learning Representations'*.

[32] Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Adam Trischler, and Yoshua Bengio. 2018. Neural Models for Key Phrase Extraction and Question Generation. In *Proceedings of the Workshop on Machine Reading for Question Answering*. 78–88.

[33] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[34] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 38–45.

[35] Dustin Wright and Isabelle Augenstein. 2020. Claim Check-Worthiness Detection as Positive Unlabelled Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*. 476–488.

[36] Liqiang Xiao, Lu Wang, Hao He, and Yaohui Jin. 2020. Modeling content importance for summarization with pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 3606–3611.

[37] Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*. 2159–2169.

[38] Jianxing Yu, Xiaojun Quan, Qinliang Su, and Jian Yin. 2020. Generating multi-hop reasoning questions to improve machine reading comprehension. (2020), 281–291.

[39] Junlang Zhan and Hai Zhao. 2020. Span model for open information extraction on accurate corpus. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 9523–9530.

[40] Yue Zhang, Zhenghua Li, Jun Lang, Qingrong Xia, and Min Zhang. 2017. Dependency parsing with partial annotations: an empirical comparison. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 49–58.

[41] Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A Boundary-aware Neural Model for Nested Named Entity Recognition. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 357–366.

A STATISTICS OF DATASETS

In this work, we mainly conduct experiments on two MRC datasets, namely SQuAD [29] and DROP [10], the statistics of datasets are summarized in Table 8.

- SQuAD: which is built upon a large set of Wikipedia articles and contains more than 100K question-answer pairs. The answer to each question is a span in the paragraph. During annotation, crowdworkers were tasked with asking no more than 5 questions. It results in the absence of many answer spans for AE task.
- DROP: which focuses on a more challenging MRC setting with quantitative reasoning over paragraphs. The answers in DROP contain both numbers, dates, and text spans. We therefore only use the extractive examples of DROP in our experiments.

Table 8: Statistics of datasets. The scale of SQuAD is much larger than DROP, while DROP has longer paragraphs. Both datasets have an average number of answer spans per paragraph about 5, except the development set of SQuAD. This is mainly because there is an additional answers collection stage in SQuAD development set.

	SQuAD			DROP		
	Train	Dev	Test	Train	Dev	Test
#Passages	16466	2067	2430	4457	507	507
Avg. passage len	139.07	143.24	120.60	256.57	230.99	250.91
Avg. spans	4.50	8.22	4.33	5.26	5.60	5.30
Avg. span len	3.59	3.75	3.11	2.33	2.26	2.26

B IMPLEMENTATION DETAILS

SCOPE. In our experiments, we inherit the Huggingface's [34] implementation as well as most of the parameters. We evaluate three types of backbone PLMs, namely BERT$_{base}$, SpanSERT$_{base}$, and RoBERTa*base*. For graph module, we use Stanza [27] to construct the graph described in Section 4.2 and stack 3 layers of GAT whose hidden size is the same as backbone PLMs. We set batch size to 12, learning rate to 2e-5 for BERT$_{base}$ and SpanSERT$_{base}$ and 1e-5 for RoBERTa*base*. Adam optimizer [15] with warming-up is used. As mentioned in Section 4.4, we estimate the prior distribution $\pi = 2.00 * \pi'$ for SQuAD and $\pi = 2.75 * \pi'$ for DROP based on our re-annotation.

Question-worthy score. For questionability score, we use the QG model implemented with T5$_{base}$ [4] and the QA model implemented with RoBERTa$_{large}$ [5]. Specifically, we generate 5 questions with beam search for each span. For worthiness score, we use BERTSum [21] to calculate the salience score for each sentence. The checkpoint we used is released by the original paper trained on CNN/DailyMail [13] without further fine-tuning.

[4] https://huggingface.co/valhalla/t5-base-qg-hl
[5] https://huggingface.co/deepset/roberta-large-squad2