How Many Answers Should I Give? An Empirical Study of Multi-Answer Reading Comprehension

Chen Zhang[1], **Jiuheng Lin**[1], **Xiao Liu**[1], **Yuxuan Lai**[3], **Yansong Feng**[1,2]*, **Dongyan Zhao**[1,4,5]

[1] Wangxuan Institute of Computer Technology, Peking University, China
[2] The MOE Key Laboratory of Computational Linguistics, Peking University, China
[3] Department of Computer Science, The Open University of China
[4] State Key Laboratory of Media Convergence Production Technology and Systems
[5] Beijing Institute for General Artificial Intelligence
{zhangch,lxlisa,fengyansong,zhaody}@pku.edu.cn
linjiuheng@stu.pku.edu.cn laiyx@ochn.edu.cn

Abstract

The multi-answer phenomenon, where a question may have multiple answers scattered in the document, can be well handled by humans but is challenging enough for machine reading comprehension (MRC) systems. Despite recent progress in multi-answer MRC, there lacks a systematic analysis of how this phenomenon arises and how to better address it. In this work, we design a taxonomy to categorize commonly-seen multi-answer MRC instances, with which we inspect three multi-answer datasets and analyze where the multi-answer challenge comes from. We further analyze how well different paradigms of current multi-answer MRC models deal with different types of multi-answer instances. We find that some paradigms capture well the key information in the questions while others better model the relationship between questions and contexts. We thus explore strategies to make the best of the strengths of different paradigms. Experiments show that generation models can be a promising platform to incorporate different paradigms. Our annotations and code are released for further research[1].

1 Introduction

In the typical setting of machine reading comprehension, such as SQuAD (Rajpurkar et al., 2016), the system is expected to extract a single answer from the passage for a given question. However, in many scenarios, questions may have multiple answers scattered in the passages, and all the answers should be found to completely answer the questions, such as the examples illustrated in Figure 1. Recently, a series of MRC benchmarks featuring multi-answer instances have been constructed, including DROP (Dua et al., 2019), Quoref (Dasigi et al., 2019) and MultiSpanQA (Li et al., 2022). Most current research efforts focus primarily on improving the overall QA performance on these benchmarks (Hu et al., 2019; Segal et al., 2020; Li et al., 2022). Yet, as far as we know, there still lacks a systematic analysis of how the phenomenon of multi-answer arises and how we can better tackle this challenge.

Figure 1: Two examples from existing multi-answer MRC datasets.

In this paper, we systematically analyze the categorization of multi-answer MRC instances and investigate how to design a strong multi-answer MRC system. We try to answer the following research questions: (1) Where does the multi-answer challenge come from? (2) How do different MRC models specifically deal with the multi-answer challenge? (3) How can we design better models by combining different multi-answer MRC paradigms?

We first analyze existing multi-answer MRC datasets to track the origin of the multi-answer challenge. Previous works have attempted to categorize multi-answer instances primarily based on the distances or relationships between multiple an-

* Corresponding author.
[1] https://github.com/luciusssss/how-many-answers

Findings of the Association for Computational Linguistics: ACL 2023, pages 5811–5827
July 9-14, 2023

swers (Li et al., 2022; Ju et al., 2022). Yet, they did not holistically consider the interaction between questions and contexts. We observe that in some cases the number of answers is indicated in the question itself (*two players* in Example A of Figure 1) while in others we have no idea until we read the documents carefully (Example B of Figure 1).

To better understand this challenge, we develop a taxonomy for the multi-answer phenomenon, based on how the number of answers is determined: the question itself suffices, or both the question and the passage should be taken into consideration. We annotate 6,857 instances from DROP, Quoref, and MultiSpanQA based on our taxonomy and find that the procedure of dataset construction has a large influence on the expressions in the questions. Most questions in crowdsourced datasets contain certain clues indicating the number of answers. By contrast, real-world information-seeking questions are less likely to specify the number of answers, which is usually dependent on the passages.

We further use our annotations to examine the performance of current MRC solutions regarding the multi-answer challenge (Hu et al., 2019; Segal et al., 2020; Li et al., 2022), which can be categorized into 4 paradigms, i.e., TAGGING, NUMPRED, ITERATIVE and GENERATION. We analyze their strengths and weaknesses and find that some efforts, e.g., NUMPRED, are good at capturing the key information in the questions, while others, e.g., ITERATIVE, can better model the relation between questions and contexts. This motivates us to investigate better ways to benefit from different paradigms.

Given the complementary nature of these paradigms, we wonder whether a combination of paradigms improves performance on multi-answer MRC. We explore two strategies, early fusion and late ensemble, to benefit from different paradigms. With a generation model as the backbone, we attempt to integrate the paradigms NUMPRED and ITERATIVE, in a lightweight Chain-of-Thought style (Wei et al., 2022). Experiments show that the integration remarkably improves the performance of generation models, demonstrating that GENERATION is a promising platform for paradigm fusion.

Our contributions are summarized as follows: (1) We design a taxonomy for multi-answer MRC instances according to how the number of answers can be determined. It considers both questions and contexts simultaneously, enlightening where the multi-answer challenge comes from. (2) We annotate 6,857 instances from 3 datasets with our taxonomy, which enables us to examine 4 paradigms for multi-answer MRC in terms of their strengths and weaknesses. (3) We explore various strategies to benefit from different paradigms. Experiments show that generation models are promising to be backbones for paradigm fusion.

2 Task Formulation

In multi-answer MRC, given a question Q and a passage P, a model should extract several spans, $A = \{a_1, a_2, ..., a_n\}(n \geq 1)$, from P to answer Q. Each span, $a_i \in A$, corresponds to a partial answer to Q, and the answer set A as a whole answers Q completely. These spans can be contiguous or discontiguous in the passage.

We distinguish between two terms, *multi-answer* and *multi-span*, which are often confused in previous works. *Multi-answer* indicates that a question should be answered with the complete set of entities or utterances. *Multi-span* is a definition from the perspective of answer annotations. In certain cases, the answer annotation of a question can be either single-span or multi-span, as explained in the next paragraph. Ideally, we expect that the answers to a multi-answer question should be annotated as multi-span in the passage, where each answer is grounded to a single span, although some of them can be contiguous in the passage.

Q0: What's Canada's official language?
P: [...] **English** and **French**, are the official languages of the Government of Canada. [...]

For example, in Q0, there are two answers, *English* and *French*, to the given question. According to the annotation guidelines of SQuAD, one might annotate this instance with a single continuous span *English and French*. Yet, this form of annotation is not preferred in the multi-answer MRC setting. It blurs the boundary of different answers and fails to denote explicitly the number of expected answers. Thus, it is suboptimal for a comprehensive model evaluation. Instead, we suggest denoting each answer with distinct spans, say, annotating this instance with two spans, *English* and *French*. With this criterion, we can encourage models to disentangle different answers. With fine-grained answer annotations, we can also assess how well a model answers a question sufficiently and precisely.

This annotation criterion generally conforms to the annotation guidelines of existing multi-answer datasets, e.g., DROP, Quoref and MultiSpanQA.

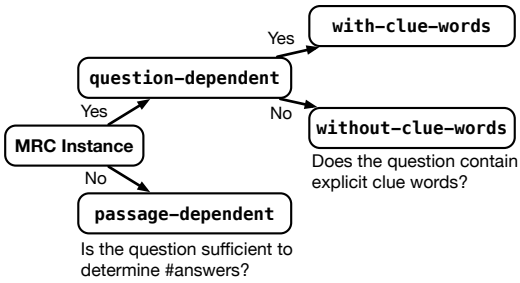

Figure 2: Illustration of our taxonomy for multi-answer MRC instances.

Type	Question	# Ans.
Cardinal	Which **two** players completed 1-yard TD pass?	2
Ordinal	Who scored the **first** touchdown of the game?	1
Comp./Super.	What's the **largest** pizza chain in America?	1
Alternative	Is San Juan Bautista incorporated **or** unincorporated?	1
Other Semantics	What are the first names of the **trio** who try to call 911?	3

Table 1: Examples of various types of clue words. Comp./Super. denotes comparatives and superlatives.

A few instances violating the criterion are considered as bad annotations, as discussed in Section 4.2. See more remarks on the task formulation in Appendix A.

3 Taxonomy of Multi-Answer MRC

To better understand the challenge of multi-answer, we first design a taxonomy to categorize various multi-answer MRC instances. It assesses how the number of answers relates to the question or passage provided. Different from the previous works that classify questions according to the distances or relations between multiple answers (Li et al., 2022; Ju et al., 2022), our taxonomy, taking both questions and passages into consideration, focuses on how the number of answers is determined. This enables us to analyze multi-answer questions and single-answer questions in a unified way. We illustrate our taxonomy in Figure 2 and elaborate on each category as follows.

`Question-Dependent` If one can infer the exact number of answers from the question without referring to the passage, this instance belongs to the `question-dependent` category. According to whether there are clue words that directly indicate the number of answers, this type is further divided into two sub-categories:

(a) In a `with-clue-words` question, one can find a few words that indicate the number of answers. In Q1, the word *two* in the question indicates that two answers are expected.

> **Q1**: What are the two official languages of Puerto Rico?
> **P**: [...] **English** is an official language of the Government of Puerto Rico. [...] As another official language, **Spanish** is widely used in Puerto Rico. [...]

We group the clue words into five types: cardinal, ordinal, comparative/superlative, alternative, and other lexical semantics, as illustrated in Table 1.

(b) In a `without-clue-words` question, although we can not locate obvious clue words, we can infer the number of answers with sentence semantics or commonsense knowledge. In Q2, we can determine that there is only one conversion result for the question based on sentence semantics instead of any single words.

> **Q2**: 1 light year equal to how many km?
> **P**: [...] The light-year is a unit of length used to express astronomical distances. It is about **9.5 trillion kilometres** or 5.9 trillion miles. [...]

In Q3, we can infer that the following question has only one answer, based on the commonsense that there is only one winner of a given Super Bowl.

> **Q3**: Who won Super Bowl XXXIX?
> **P**: [...] The Eagles advanced to Super Bowl XXXIX, where they dueled the 2004 **New England Patriots** season. [...] The Patriots won 24-21. [...]

`Passage-Dependent` In a `passage-dependent` instance, the question itself is not adequate to infer the number of answers. One needs to rely on the provided passage to decide how many answers are needed to answer the question. In Q4, we have no idea of the number of answers solely based on the question. If we refer to the passage, we will find ten answers to the question.

> **Q4**: Which countries does the Danube River flow through?
> **P**: [...] Originating in **Germany**, the Danube flows southeast for 2,850 km, passing through or bordering **Austria**, **Slovakia**, **Hungary**, **Croatia**, **Serbia**, **Romania**, **Bulgaria**, **Moldova** and **Ukraine** before draining into the Black Sea. [...]

4 Analyses of Multi-Answer Datasets

We investigate existing multi-answer datasets based on our designed taxonomy to analyze where the multi-answer challenge comes from.

Dataset	All	Single-Ans.	Multi-Ans.
DROP	3,133	2,609	524
Quoref	2,418	2,198	220
MultiSpanQA	1,306	653	653
Total	6,857	5,460	1,397

Table 2: The number of instances for human annotation in the validation set of each dataset.

4.1 Datasets

We annotate the validation sets of three widely-used multi-answer MRC datasets, i.e., DROP (Dua et al., 2019), Quoref (Dasigi et al., 2019), and MultiSpanQA (Li et al., 2022). The number of annotated questions is listed in Table 2 and more statistics are in Appendix B.

DROP is a crowdsourced MRC dataset for evaluating the discrete reasoning ability. The annotators are encouraged to devise questions that require discrete reasoning such as arithmetic. DROP has four answer types: numbers, dates, single spans, and sets of spans. Since the previous two types of answers are not always exact spans in the passages, we only consider the instances whose answers are single spans or sets of spans.

Quoref focuses on the coreferential phenomena. The questions are designed to require resolving coreference among entities. 10% of its instances require multiple answer spans.

MultiSpanQA is a dataset specialized for multi-span reading comprehension. The questions are extracted from NaturalQuestions (Kwiatkowski et al., 2019), which are real queries from the Google search engine.

4.2 Annotation

Annotation Process Our annotation process is two-staged: we first automatically identify some `question-dependent` instances and then recruit annotators to classify the remaining ones.

In the first stage, we automatically identify the questions containing certain common clue words such as numerals (full list in Appendix B) to reduce the workload of whole-process annotation. Afterward, the annotators manually check whether each instance is `question-dependent`. Out of the 4,594 recalled instances, 3,727 are identified as `question-dependent`.

In the second stage, we recruit annotators to annotate the remaining 3,130 instances. For each instance, given both the question and the answers, the annotators should first check whether the form of answers is correct and mark incorrect cases as `bad-annotation`[2]. We show examples of common `bad-annotation` cases in Table 10. After filtering out the `bad-annotation` ones, the annotators are presented with the question only and should decide whether they could determine the number of answers solely based on the question. If so, this instance is annotated as `question-dependent`; otherwise `passage-dependent`. For a `question-dependent` instance, the annotators are further asked to extract the clue words, if any, from the question, which determines whether the instance is `with-clue-words` or `without-clue-words`.

Quality Control Six annotators participated in the annotation after qualification. Each instance is annotated by two annotators. In case of any conflict, a third annotator resolves it. An instance is classified as `bad-annotation` if any annotator labels it as `bad-annotation`. Cohen's Kappa between two initial annotators is 0.70, indicating substantial agreement. See more details in Appendix B.

4.3 Analyses of Annotation Results

With our annotated data, we study how the multi-answer instances differ across different datasets under our designed taxonomy. We find that the distributions of instance types are closely related to how the datasets are constructed.

Instance Types The distributions of instance types in different datasets are shown in Table 3. `Question-dependent` prevails in DROP and Quoref, making up over 70% of the two datasets. In contrast, most instances in MultiSpanQA are `passage-dependent`. This difference stems from how the questions are collected. DROP and Quoref use crowdsourcing to collect questions with specific challenges. Given a passage, the annotators know the answers in advance and produce questions that can only be answered through certain reasoning skills. These artificial questions are more likely to contain clues to the number of answers, such as the question with ordinal in Table 1. By contrast, the questions in MultiSpanQA are collected from search engine queries. Users generally have no idea of the answers to the queries. The number of answers, as a result, is more often de-

[2]In the first stage, the annotators also need to check whether an instance is `bad-annotation`.

Dataset	passage-dependent	question-dependent			bad-annotation
		All	with-clue-word	no-clue-word	
DROP	826 (26.4%)	**2,242 (71.6%)**	2,204 (70.3%)	38 (1.2%)	65 (2.1%)
Quoref	711 (29.4%)	**1,704 (70.5%)**	1,639 (67.8%)	65 (2.7%)	3 (0.2%)
MultiSpanQA	**991 (75.9%)**	285 (21.8%)	121 (9.3%)	164 (12.6%)	30 (2.3%)
Total	2,528 (36.9%)	4,231 (61.7%)	3,964 (57.8%)	267 (3.9%)	98 (1.4%)

Table 3: Distribution of instance types in three datasets.

Dataset	with-clue-word	Cardinal	Ordinal	Comp./Super.	Alternative	Other Semantics
DROP	2,204	113 (5.1%)	592 (26.9%)	**1,298 (58.9%)**	1,214 (55.1%)	135 (6.1%)
Quoref	1,639	83 (5.1%)	35 (2.1%)	25 (1.5%)	0 (0.0%)	**1,501 (91.6%)**
MultiSpanQA	121	**51 (41.8%)**	26 (21.3%)	23 (19.0%)	2 (1.6%)	19 (15.6%)

Table 4: Distribution of clue word types in three datasets. A question may contain multiple types of clue words.

pendent on the provided passages, such as Q4 in Section 3.

Clue Words Since a large portion (57.8%) of the annotated instances belong to the with-clue-word type, we further investigate the distribution of clue words in different datasets, shown in Table 4. On the one hand, the questions contain a large variety of clue words, demonstrating the complexity of multi-answer MRC. On the other hand, the prevailing type of clue words is different in each dataset, reflecting the preference in dataset construction. Specifically, nearly 60% of the with-clue-word questions in DROP are alternative questions with comparatives/superlatives, because DROP's annotators are encouraged to inject discrete reasoning challenges, e.g., comparison, when writing questions. In Quoref, 91% of the clue words indicate the number of answers through their lexical semantics. This unbalanced distribution results from the emphasis on coreference resolution: most questions begin with *what is the name of the person who ...*, where *name of the person* is identified as clue words. In MultiSpanQA, whose questions are search engine queries, 63% of the with-clue-word questions contain numerals. If users already know the number of desired answers, they tend to restrict it in the question, such as *seven wonders of the world*.

We provide more analyses on of how the instance types are distributed with respect to the specific number of answers in Appendix C.

5 Existing Multi-Answer MRC Models

Based on our categorization of the multi-answer instances, we continue to investigate how existing multi-answer MRC models perform differently on various types of multi-answer instances. We summarize current solutions into four paradigms according to how they obtain multiple answers, as illustrated in Figure 3.

TAGGING Segal et al. (2020) cast the multi-answer MRC task as a sequence tagging problem, similar to named entity recognition (NER), so that the model can extract multiple non-contiguous spans from the context.

NUMPRED (Number Prediction) Hu et al. (2019) first predict the number of answers k as an auxiliary task and then select the top k non-overlapped ones from the output candidate spans.

ITERATIVE Searching for evidence iteratively is widely adopted in many QA tasks (Xu et al., 2019; Zhao et al., 2021; Zhang et al., 2021), but it is not explored in multi-answer MRC. We adapt this idea to extract multiple answers iteratively. In each iteration, we append the previously extracted answers to the question, with the word *except* in between, and then feed the updated question to a single-answer MRC model. The iterative process terminates when the model predicts no more answers.

GENERATION Generation has been adopted as a uniform paradigm for many QA tasks (Khashabi et al., 2020, 2022), but it is less explored on multi-answer MRC. For GENERATION, we concatenate all answers, with semicolons as separators, to form an output sequence, and finetune the model to generate it conditioned on the question and passage.

5.1 Experimental Setup

Implementation Details We use RoBERTa-base (Liu et al., 2019) for the three extractive

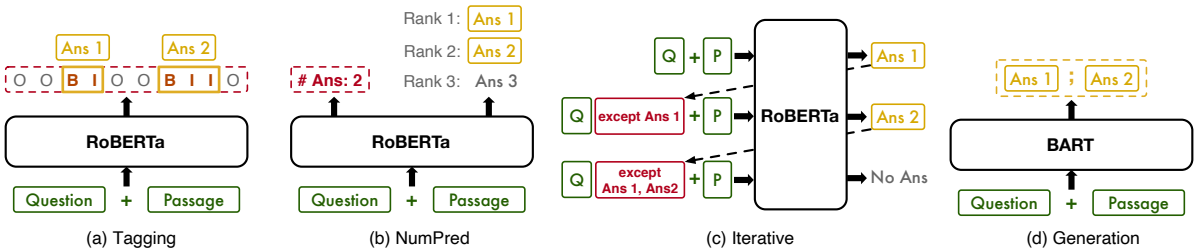

Figure 3: An illustration of four paradigms for multi-answer MRC.

Model	EM			PM		
	P	R	F1	P	R	F1
DROP						
TAGGING	**61.86**	**63.91**	**62.87**	**77.53**	**77.39**	**77.46**
NUMPRED	61.59	56.77	59.09	76.71	74.86	75.77
ITERATIVE	60.66	60.07	60.36	76.19	76.04	76.11
GENERATION	60.07	57.15	58.58	75.39	72.39	73.86
Quoref						
TAGGING	**71.00**	**72.21**	**71.60**	**80.44**	**79.74**	**80.09**
NUMPRED	65.61	63.57	64.57	77.30	78.20	77.75
ITERATIVE	67.28	66.35	66.81	78.57	78.58	78.57
GENERATION	63.57	63.39	63.48	73.38	74.02	73.70
MultiSpanQA						
TAGGING	61.31	**68.84**	64.85	80.45	**83.08**	81.75
NUMPRED	55.03	46.06	50.15	80.16	75.26	77.63
ITERATIVE	**66.32**	67.98	**67.14**	**84.39**	80.96	**82.64**
GENERATION	65.40	62.60	63.97	82.06	78.14	80.06

Table 5: Performance of four paradigms on three datasets.

Model	p-dep.	q-dep.		
		All	w/-clue	w/o-clue
DROP				
TAGGING	**74.57**	**79.11**	**80.88**	68.77
NUMPRED	72.37	77.54	79.32	70.08
ITERATIVE	73.47	77.60	79.21	65.73
GENERATION	72.18	74.77	76.19	**72.62**
Quoref				
TAGGING	70.60	**84.86**	**85.23**	75.76
NUMPRED	69.45	81.88	82.44	70.12
ITERATIVE	**71.42**	82.18	82.37	**77.30**
GENRATION	66.31	77.41	78.38	52.63
MultiSpanQA				
TAGGING	82.28	79.66	86.60	73.36
NUMPRED	77.77	77.11	78.19	**78.77**
ITERATIVE	**82.78**	**82.09**	**87.22**	77.80
GENERATION	80.57	78.05	81.73	75.85

Table 6: The performance (PM F1) of four paradigms on different types of instances. `p-dep.` denotes `passage-dependent`. `q-dep.` denotes `question-dependent`.

paradigms and BART-base (Lewis et al., 2020) for GENERATION. We train models on the training sets of each dataset and evaluate them on the corresponding validation sets with our instance type annotations. See more details in Appendix D.1.

Metrics We adopt the official metrics of MultiSpanQA (Li et al., 2022), including the precision (P), recall (R), and F1 in terms of exact match (EM) and partial match (PM). See Appendix D.2 for details.

5.2 Results and Analyses

We report the overall performance in Table 5, and the performance on different instance types in Table 6. We observe that each of these paradigms has its own strengths and weaknesses.

TAGGING outperforms other paradigms on DROP and Quoref, whose dominating instance type is `question-dependent`. Although TAGGING has no explicit answer number prediction step, it can still exploit this information implicitly because it takes the question into account during the sequential processing of every token. Besides, TAGGING, as a common practice for entity recognition, is good at capturing the boundaries of entities. Thus, it is not surprising that it performs the best on DROP and Quoref, most of whose answers are short entities.

ITERATIVE achieves the best overall performance on MultiSpanQA, whose prevailing instance type is `passage-depenent`. This paradigm does not directly exploit the information of the number of answers given in the question. Rather, it encourages adequate interactions between questions and passages, performing single-answer extraction at each step. As a result, ITERATIVE does well for the questions whose number of answers heavily depends on the given context.

As for NUMPRED, although we expect high performance on `question-dependent` instances, it lags behind TAGGING by approximately 2% in PM F1 on DROP and Quoref. This might result

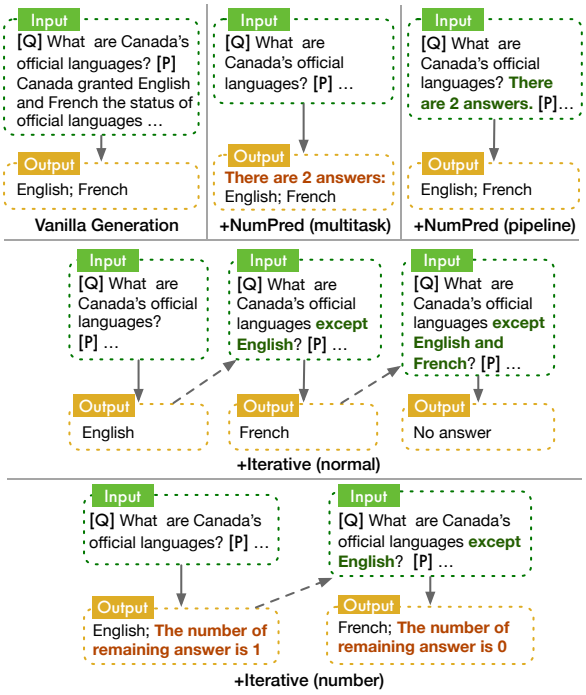

Figure 4: An illustration of different strategies for early fusion of paradigms.

from the gap between training and inference. The model treats the answer number prediction and answer span extraction as two separate tasks during training, with limited interaction. Yet during inference, the predicted number of answers is used as a hard restriction on multi-span selection. Different from the decent performance on DROP and Quoref, NUMPRED performs worst among the four paradigms on MultiSpanQA, because it is difficult for models to accurately predict the number of answers for a long input text that requires thorough understanding.

Among all paradigms, GENERATION generally performs the worst. Under the same parameter scale, extractive models seem to be the better choice for tasks whose outputs are exact entity spans from the input, while generation models do well in slightly longer answers. This also explains the smaller gap between GENERATION and extractive paradigms on MultiSpanQA compared to that on DROP and Quoref: MultiSpanQA has many descriptive long answers instead of short entities only.

6 Fusion of Different Paradigms

From the above analysis, we can see that extractive methods can better locate exact short spans in the passage, and NUMPRED can provide potential guidance on the number of answers. Meanwhile, the generation models can better handle longer answers and are more adaptable to different forms of inputs and outputs. Now an interesting question is how to combine different paradigms to get the best of both worlds.

We explore two strategies for combining different paradigms: **early fusion** and **late ensemble**. The former mixes multiple paradigms in terms of model architectures while the latter ensembles the predictions of different models. We discuss our exploration of late ensemble in Appendix E.1 since model ensemble is a well-explored technique. Here we primarily elaborate on early fusion. We carry out a series of pilot studies to demonstrate the potential of paradigm fusion.

Previous works attempt to fuse two extractive paradigms, TAGGING and NUMPRED (Segal et al., 2020; Li et al., 2022). However, they only lead to marginal improvements, probably because TAGGING can already implicitly determine answer numbers well and the help of NUMPRED is thus limited.

Although the performance of base-size generation models on multi-answer MRC is inferior to that of extractive ones, generation models of larger sizes show great potential with more parameters and larger pre-training corpora (Khashabi et al., 2020, 2022). More importantly, GENERATION can easily adapt to various forms of inputs and outputs. We carry out pilot studies using a generation model as the backbone and benefiting from the ideas of other paradigms. We propose several lightweight methods to combine GENERATION with NUMPRED and ITERATIVE, as illustrated in Figure 4.

GENERATION + NUMPRED Inspired by recent works on Chain-of-Thought (Wei et al., 2022), we guide the model with prompts indicating the number of answers. We introduce a **NUMPRED prompt sentence** (NPS) in the form of *There are* $\{2, 3, ...\}$ *answers/There is only one answer*. We experiment with two variants, multitask and pipeline. In the multitask variant, the model outputs an NPS before enumerating all the answers. In the pipeline variant, we predict the number of answers with a separate classifier and then append the NPS to the question as extra guidance.

GENERATION + ITERATIVE We substitute the original extractor of ITERATIVE with a generator. The iterative process terminates when the model

outputs the string *No answer*. Besides the normal setting, we experiment with another variant that additionally outputs an NPS in the form of *The number of remaining answers is* $\{1, 2, 3, ...\}$.

Results Our main experiments are conducted with BART-base and BART-large due to our limited computational budget. For the pipeline variant of GENERATION + NUMPRED, we use RoBERTa-base as an answer number classifier. The overall experiment results are reported in Table 7 and the results on different question types are reported in Appendix E.2.

When GENERATION is multitasking with NUMPRED, it outperforms the vanilla one consistently. The NPS in the output provides a soft but useful hint for the succeeding answer generation, improving the accuracy of answer number prediction by 1.7% on average for BART-base. The pipeline variant is often inferior to the multitasking one due to error propagation. Especially, its performance drops a lot on MultiSpanQA, whose instances are `passage-dependent`. The accuracy of the answer number classifier on MultiSpanQA lags behind that on the other two datasets by more than 12%. Thus the NPS in the input, with an unreliably predicted answer number, is more likely to mislead the subsequent answer span generation.

The combination of GENERATION and ITERATIVE does not always lead to improvement. This might be because the answer generation process of GENERATION is already in an iterative style: in the output sequence, each answer is generated conditioned on the previously-generated ones. The incorporation of ITERATIVE thus does not lead to further improvement. When we further introduce an NPS with the number of remaining answers, the performance generally outperforms the normal setting. This proves that GENERATION, as a backbone, is easy to integrate with various hints.

Pilot Study on GPT-3.5 To investigate whether these fusion strategies work on larger models, we conduct a pilot study on GPT-3.5. We use the 653 multi-answer instances in the validation set of MultiSpanQA for experiments. The prompts are listed in Appendix E.2. The experiment results are shown in Table 8.

When given only one example for in-context learning, GPT-3.5 can already achieve 79.27% PM F1 on the multi-answer instances, with only a small gap between BART trained on full data. Its EM F1 score is low because GPT-3.5 cannot handle the boundaries of answer spans well. This is not unsurprising since one example is not sufficient for GPT-3.5 to learn the annotation preference of span boundaries in MultiSpanQA. If we ask GPT-3.5 to predict the number of answers before giving all the answers, we observe an improvement of 10.1% EM F1 and 3.1% PM F1. This proves the effectiveness of fusing NUMPED with larger generation models

As evidenced by the above trials, it is promising to fusion different paradigms. We hope that our exploration will inspire future works adopting larger generation models for multi-answer MRC.

Model	Base		Large	
	EM	PM	EM	PM
DROP				
Vanilla GENERATION	58.58	73.86	66.43	80.55
+NUMPRED (multitask)	**60.02**	**74.34**	**69.61**	**82.85**
+NUMPRED (pipeline)	59.19	73.94	66.45	80.63
+ITERATIVE (normal)	58.44	73.58	66.55	80.53
+ITERATIVE (number)	58.98	74.07	68.19	82.17
Quoref				
Vanilla GENERATION	63.48	73.70	76.57	84.47
+NUMPRED (multitask)	66.25	75.43	**77.04**	84.45
+NUMPRED (pipeline)	67.94	77.42	75.42	83.66
+ITERATIVE (normal)	**68.81**	**78.23**	74.72	82.60
+ITERATIVE (number)	63.33	73.34	76.67	**84.57**
MultiSpanQA				
Vanilla GENERATION	63.97	80.06	69.13	84.61
+NUMPRED (multitask)	**64.85**	**80.58**	**69.31**	**84.82**
+NUMPRED (pipeline)	39.71	60.94	45.34	68.09
+ITERATIVE (normal)	63.26	79.97	65.62	82.88
+ITERATIVE (number)	63.84	80.04	66.77	83.41

Table 7: The performance (EM F1 and PM F1) of different strategies for early fusion of paradigms.

Model	Setting	EM F1	PM F1
Vanilla BART-base	Supervised	66.77	81.24
Vanilla BART-large	Supervised	71.93	85.83
Vanilla GPT-3.5	One-Shot	53.34	79.27
GPT-3.5 + NUMPED	One-Shot	63.45	82.38

Table 8: The performance of BART and GPT-3.5 on the multi-answer instances of MultiSpanQA.

7 Related Works

Compared to the vast amount of single-answer MRC datasets, the resources for multi-answer MRC are limited. Aside from the datasets in Section 4.1, MASH-QA (Zhu et al., 2020) focuses on the healthcare domain, with 27% of the questions having multiple long answers, ranging from phrases to sentences. CMQA (Ju et al., 2022) is

another multi-answer dataset in Chinese, featuring answers with conditions or different granularities. For our analysis, we select two commonly-used datasets, DROP and Quoref, as well as a newly-released dataset, MultiSpanQA.

Current models addressing multi-answer MRC generally fall into two paradigms: TAGGING (Segal et al., 2020) and NUMPRED (Hu et al., 2019), as explained in Section 5. ITERATIVE (Xu et al., 2019; Zhao et al., 2021; Zhang et al., 2021; Gao et al., 2021) and GENERATION (Khashabi et al., 2020, 2022) have been adopted for many types of QA tasks including knowledge base QA, multiple-choice QA, and open-domain QA. Nevertheless, their performance on multi-answer MRC is less explored. In our paper, we also study how to adapt these paradigms for multi-answer MRC. Apart from the exploration of model architectures for multi-answer MRC, Lee et al. (2023) attempt to generate multi-answer questions as data augmentation.

Previous works have made preliminary attempts in fusing two extractive paradigms. Segal et al. (2020) adopt a single-span extraction model for single-answer questions and TAGGING for multi-answer questions; Li et al. (2022) add a NUMPRED head to the TAGGING framework. The predicted number of answers is used to adjust the tagging results. Both strategies lead to marginal improvement over the baselines. We instead resort to GENERATION for paradigm fusion, considering its potential with larger sizes and its flexibility in inputs and outputs.

8 Conclusion

In this paper, we conduct a systematic analysis for multi-answer MRC. We design a new taxonomy for multi-answer instances based on how the number of answers is determined. We annotate three datasets with the taxonomy and find that multi-answer is not merely a linguistic phenomenon; rather, many factors contribute to it, especially the process of data collection. With the annotation, we further investigate the performance of four paradigms for multi-answer MRC and find their strengths and weaknesses. This motivates us to explore various strategies of paradigm fusion to boost performance. We believe that our taxonomy can help determine what types of questions are desirable in the annotation process and aid in designing more practical annotation guidelines. We hope that our annotations can be used for more fine-grained diagnoses of MRC systems and encourage more robust MRC models.

Limitations

First, our taxonomy of multi-answer MRC instances only considers whether we know the *exact* number of answers from the questions. In some cases, one might have an *imprecise estimate* of answer numbers from the question. For example, for the question *Who are Barcelona's active players?*, one might estimate that there are dozens of active players for this football club. Yet, these estimations are sometimes subjective and difficult to quantify. Therefore, this instance is classified as `passage-dependent` according to our current taxonomy. We will consider refining our taxonomy to deal with these cases in the future.

Second, we did not conduct many experiments with pre-trained models larger than the large-size ones due to limited computational budgets. Generation models of larger sizes show great potential with more parameters and larger pre-training corpora. We encourage more efforts to deal with multi-answer MRC with much larger models, such as GPT-3.5.

Acknowledgments

This work is supported by NSFC (62161160339). We would like to thank the anonymous reviewers for their valuable suggestions, and our great annotators for their careful work, especially Zhenwei An, Nan Hu, and Hejing Cao. Also, we would like to thank Quzhe Huang for his help in this work. For any correspondence, please contact Yansong Feng.

References

Pradeep Dasigi, Nelson F. Liu, Ana Marasović, Noah A. Smith, and Matt Gardner. 2019. Quoref: A reading comprehension dataset with questions requiring coreferential reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5925–5932, Hong Kong, China. Association for Computational Linguistics.

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), pages 2368–2378, Minneapolis, Minnesota. Association for Computational Linguistics.

Yifan Gao, Henghui Zhu, Patrick Ng, Cicero Nogueira dos Santos, Zhiguo Wang, Feng Nan, Dejiao Zhang, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021. Answering ambiguous questions through generative evidence fusion and round-trip prediction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3263–3276, Online. Association for Computational Linguistics.

Minghao Hu, Yuxing Peng, Zhen Huang, and Dongsheng Li. 2019. A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1596–1606, Hong Kong, China. Association for Computational Linguistics.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Yiming Ju, Weikang Wang, Yuanzhe Zhang, Suncong Zheng, Kang Liu, and Jun Zhao. 2022. CMQA: A dataset of conditional question answering with multiple-span answers. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1697–1707, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. 2022. Unifiedqa-v2: Stronger generalization via broader cross-format training. *arXiv preprint arXiv:2202.12359*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Seongyun Lee, Hyunjae Kim, and Jaewoo Kang. 2023. Liquid: A framework for list question answering dataset generation. *arXiv preprint arXiv:2302.01691*.

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

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. MultiSpanQA: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

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*.

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

Elad Segal, Avia Efrat, Mor Shoham, Amir Globerson, and Jonathan Berant. 2020. A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3074–3080, Online. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

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

Kun Xu, Yuxuan Lai, Yansong Feng, and Zhiguo Wang. 2019. Enhancing key-value memory neural networks for knowledge based question answering. In *Proceedings of the 2019 Conference of the North American*

Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 2937–2947, Minneapolis, Minnesota. Association for Computational Linguistics.

Chen Zhang, Yuxuan Lai, Yansong Feng, and Dongyan Zhao. 2021. Extract, integrate, compete: Towards verification style reading comprehension. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2976–2986, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chen Zhao, Chenyan Xiong, Jordan Boyd-Graber, and Hal Daumé III. 2021. Multi-step reasoning over unstructured text with beam dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4635–4641, Online. Association for Computational Linguistics.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849, Online. Association for Computational Linguistics.

A Additional Remarks on Task Formulation

As discussed in Section 2, *multi-answer* and *multi-span* are two orthogonal concepts. We have already shown an example (Q0 in Section 2) where a *multi-answer* question can be annotated as *single-span* by certain annotation guidelines. Here is another example to demonstrate the difference between *multi-answer* and *multi-span*.

> **Q**: Which offer of Triangle-Transit is most used by students?
> **P**: [...] Triangle-Transit offers **scheduled**, fixed-route regional and commuter **bus service**. The first is most used by students.

This is an example where a *single-answer* question can be annotated as *multi-span*. A single answer, *scheduled bus service*, will be annotated as multiple-span, i.e., *scheduled* and *bus service* in the passage.

Considering the differences between *multi-answer* and *multi-span*, we suggest carefully distinguishing between these two terms in the future.

B Annotation Details

Dataset Statistics We report more statistics of the annotated datasets in Table 9. MultiSpanQA has the largest average number of answers since it is a dataset designed especially for multi-answer questions. The answers in MultiSpanQA are generally longer than those in DROP and Quoref because many of the answers in MultiSpanQA are long descriptive phrases or clauses instead of short entities. For all three datasets, the distances between answers are large. This indicates that the answers to a large proportion of the questions are discontiguous in the passages, demonstrating the difficulty of multi-answer MRC.

Dataset	DROP	Quoref	MultiSpanQA
Length of Question	9.4	15.5	9.0
Length of Context	214.7	326.0	219.9
Length of Answer	1.9	1.6	3.1
#Answers	1.2	1.1	1.9
#Answers (Multi)	2.5	2.4	2.9
Distance Between Ans.	30.5	17.3	10.3

Table 9: Dataset Statistics, including the (a) average length (in words) of questions, contexts, and answers, (b) the average number of answers for all the instances and the multi-answer ones, (c) the average distances (in words) between answers.

Pre-defined Clue Words Here, we list the pre-defined clue words in the first stage of annotation:

- Numerals, including cardinals and ordinals
- Comparatives and superlatives
- The word *or*, as an indicator of alternative questions.
- Other words, including *only*, *last*, *single*, *name of the person*, and, *top*.

Selection of Annotators A total of 10 graduates proficient in English participated in our annotation task. We first provided training materials to the annotators and asked them to annotate 100 sample instances. Based on their annotation accuracy on the sample instances, six of them are qualified to continue annotating the remaining instances. The annotators are paid $10 per hour, which is adequate given the participants' demographic. The annotators are informed of how the data would be used.

Examples of Bad Annotations In Table 10, we present several examples we marked as `bad-annotation`. Common reasons for bad annotations including incorrect segmentation of answers, irrelevant answers, and duplicate answers.

C Additional Analyses on Annotation Results

We report more statistics of the annotation results in Table 11 and Table 12, and conduct additional analyses from the perspective of the number of answers.

For multi-answer instances, `passage-dependent` questions account for the largest proportion, followed by `with-clue-word`. As for the single-answer instances in DROP and Quoref, they tend to be `question-dependent`, while in MultiSpanQA most of them are `passage-dependent`. In terms of the clue words in the `with-clue-word` questions, cardinal numbers are more common in multi-answer questions while other types of clue words are more likely to appear in single-answer questions.

D Experimental Setup

D.1 Implementation Details

We use base-size models for our main experiments for sake of energy savings. Since T5-base has twice as many parameters as RoBERTa-base and BART-base, we did not use it to ensure fair comparisons. We carefully tune each model on the

Type	Example	Explanation
Incorrect segmentation of answers	**Source:** DROP **Question:** Which event occurred first, Duke Magnus Birgersson started a war or Erik Klipping gathered a large army? **Annotated Answers:** Duke Magnus Birgersson; started a war	The correct answer *Duke Magnus Birgersson started a war* is wrongly split into two spans, *Duke Magnus Birgersson* and *started a war*.
Irrelevant Answers	**Source:** DROP **Question:** Who scored first in the second half of the game, Cowboys or 49ers? **Annotated Answers:** end of the half; San Francisco scored; making the score 28-14	All three annotated answers are not related to the questions. A correct answer should be either *Cowboys* or *49ers*.
Duplicate answers	**Source:** MultiSpanQA **Question:** who benefited by title ix of the education amendments **Annotated Answers:** women; women playing college sports	One annotead answer, *women* is duplicated with the other, *women playing college sports*.

Table 10: Examples and explanations of `bad-annotation` cases.

#Ans	p-dep.	q-dep. w/-clue	q-dep. w/o-clue
DROP			
1	480	**2,085**	37
2	**209**	105	1
3	**74**	11	0
>3	**63**	3	0
Quoref			
1	582	**1,548**	65
2	**82**	62	0
3	**28**	23	0
>3	**19**	6	0
MultiSpanQA			
1	**448**	56	140
2	**300**	32	22
3	**131**	14	2
>3	**112**	19	0

Table 11: Distribution of question types according to the number of answers. `p-dep.` denotes `passage-dependent`. `q-dep.` denotes `question-dependent`.

#Ans	Alternative	Cardinal	Comp./Super.	Ordinal	Others
DROP					
1	1,213	3	1,293	588	132
2	1	97	4	3	3
3	0	11	0	0	0
>3	0	2	1	1	0
Quoref					
1	0	1	25	35	1,492
2	0	55	0	0	7
3	0	21	0	0	2
>3	0	6	0	0	0
MultiSpanQA					
1	2	1	14	25	14
2	0	21	5	1	5
3	0	12	2	0	0
>3	0	17	2	0	0

Table 12: Distribution of clue word types in three datasets according to the number of answers.

training set and report its best performance on the validation set. We use an NVIDIA A40 GPU for experiments. A training step takes approximately 0.5s for RoBERTa-base and 0.2s for BART-base. We describe the implementation details of different models here.

TAGGING We use the implementation by Segal et al. (2020)[3]. We use the IO tagging variant, which achieves the best overall performance according to the original paper. We adopt the best-performing hyperparameters provided by the original paper.

NUMPRED Because the implementation by the original paper (Hu et al., 2019)[4] does not support RoBERTa, we re-implement the model with Huggingface Transformers (Wolf et al., 2020)[5]. We use the representation of the first token in the input sequence for answer number classification. The maximum number of answers of the classifier is 8. The batch size is 12. The number of training epochs is 10. The learning rate is 3e-5. The maximum sequence length is 512.

ITERATIVE Our implementation is based on the scripts of MRC implemented by Huggingface. Dur-

[3] https://github.com/eladsegal/tag-based-multi-span-extraction

[4] https://github.com/huminghao16/MTMSN

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

ing training, the order of answers for each iteration is determined by their order of position in the passage. The batch size is 8. The number of training epochs is 8. The learning rate is 3e-5. The maximum sequence length is 384. During inference, the beam size is set to 3 and the length penalty is set to 0.7. The maximum length of answers is 10.

GENERATION Our implementation is based on the scripts of sequence generation implemented by Huggingface. The batch size is 12. The learning rate is 3e-5. The number of training epochs is 10. The maximum input length is 384. The maximum output length is 60.

D.2 Evaluation Metrics

Here, we describe the evaluation metrics used in our experiments, which are the official ones used by MultiSpanQA (Li et al., 2022). The metrics consist of two part: exact match and partial match.

Exact Match An exact match occurs when a prediction fully matches one of the ground-truth answers. We use micro-averaged precision, recall, and F1 score for evaluation.

Partial Match For each pair of prediction p_i and ground truth answer t_j , the partial retrieved score s_{ij}^{ret} and partial relevant score s_{ij}^{rel} are calculated as the length of the longest common substring (LCS) between p_i and t_j, divided by the length of p_i and t_j respectively, as:

$$s_{ij}^{ret} = \frac{\text{len}(\text{LCS}(p_i, t_j))}{\text{len}(p_i)}$$

$$s_{ij}^{rel} = \frac{\text{len}(\text{LCS}(p_i, t_j))}{\text{len}(t_j)}$$

Suppose there are n predictions and m ground truth answers for a question. We compute the partial retrieved score between a prediction and all answers and keep the highest one as the retrieved score of that prediction. Similarly, for each ground truth answer, the relevant score is the highest one between it and all predictions. The precision, recall, and F1 are finally defined as follows:

$$\text{Precision} = \frac{\sum_{i=1}^{n} \max_{j \in [1,m]}(s_{ij}^{ret})}{n}$$

$$\text{Recall} = \frac{\sum_{j=1}^{m} \max_{i \in [1,n]}(s_{ij}^{rel})}{m}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

We use micro-averaged scores for these metrics.

E Additional Experiment Results

E.1 Late Ensemble

By late ensemble, we aggregate the outputs from models of different paradigms to boost performance. We experiment with a simple voting strategy. If a span is predicted as an answer by more than one model, we add it to the final prediction set. If a span is part of another span, we consider them equivalent and take the longer one. In rare cases where the four models predict totally different answers, we add them all to the final prediction set.

Our voting strategy leads to improvements of 1.0%, 1.2%, and 1.3% in PM F1 on DROP, Quoref, and MultiSpanQA, respectively, over the best-performing models in Table 5. Yet, this strategy might discard many correct answers. In the future, we can explore more sophisticated strategies. For example, similar to the idea of Mixture of Experts (Jacobs et al., 1991), the system can evaluate the probability that the instance belongs to a certain category and then adjust the weight of the model based on its capabilities in this category.

E.2 Early Fusion

In Table 13, we report the performance of different strategies for early fusion on different types of instances. In Table 14, we list the prompts used for our pilot study on GPT-3.5.

F Licenses of Scientific Artifacts

The license for Quoref and DROP is CC BY 4.0. The license for HuggingFace Transformers is Apache License 2.0. Other datasets and models provide no licenses.

Model	BART-base				BART-large			
	p-dep.	q-dep.			p-dep.	q-dep.		
		All	w/-clue	w/o-clue		All	w/-clue	w/o-clue
DROP								
Vanilla GENERATION	72.18	74.77	76.19	72.62	78.57	81.65	83.42	77.31
+NUMPRED (multitask)	**72.45**	75.37	76.80	70.58	80.35	**84.24**	**86.06**	77.88
+NUMPRED (pipeline)	70.58	**75.72**	**77.14**	**76.77**	76.79	82.66	84.34	**79.65**
+ITERATIVE (normal)	71.82	74.55	75.97	68.26	78.07	81.90	83.56	74.03
+ITERATIVE (number)	71.90	75.27	76.66	72.93	**80.58**	83.05	84.91	72.66
Quoref								
Vanilla GENERATION	66.31	77.41	78.38	52.63	76.41	**88.51**	**88.90**	79.76
+NUMPRED (multitask)	67.54	79.37	80.15	58.30	**77.73**	87.88	88.11	**82.35**
+NUMPRED (pipeline)	66.26	**82.88**	**83.55**	65.20	75.37	87.71	88.22	77.36
+ITERATIVE (normal)	**69.40**	82.68	83.24	**68.24**	73.13	87.43	87.93	77.20
+ITERATIVE (number)	65.79	77.16	77.97	55.63	77.69	88.09	88.59	75.41
MultiSpanQA								
Vanilla GENERATION	80.57	78.05	81.73	75.85	84.52	**84.96**	88.78	**81.65**
+NUMPRED (multitask)	**81.08**	78.65	81.09	**77.73**	**84.83**	84.80	**89.66**	81.06
+NUMPRED (pipeline)	60.24	63.56	68.67	58.25	67.27	71.21	74.53	69.33
+ITERATIVE (normal)	80.46	78.06	81.81	74.78	83.16	81.78	84.84	80.87
+ITERATIVE (number)	80.15	**79.63**	**83.47**	76.17	83.49	83.06	86.08	80.44

Table 13: The performance (PM F1) of different strategies for early fusion on different types of instances. `p-dep.` denotes `passage-dependent`. `q-dep.` denotes `question-dependent`.

Vanilla GPT-3.5

Answer the question based on the given context. Each question has more than one answer. Please give all the answers and separate them with a semicolon.
Context: Laura Horton is a fictional character from the NBC soap opera , Days of Our Lives , a long - running serial drama about working class life in the fictional , United States town of Salem . Created by writer Peggy Phillips , the role was originated by actress Floy Dean on June 30 , 1966 till October 21 , 1966 . Susan Flannery stepped into the role from November 22 , 1966 to May 27 , 1975 . Susan Oliver briefly stepped into the role from October 10 , 1975 , to June 9 , 1976 , followed by Rosemary Forsyth from August 24 , 1976 , to March 25 , 1980 .
Question: who played laura horton on days of our lives
Answers: Floy Dean; Susan Flannery; Susan Oliver; Rosemary Forsyth

Following the example above and answer the following multi-answer question. Please give all the answers and separate them with a semicolon.
Context: `{context}`
Question: `{question}`
Answers:

GPT-3.5 + NUMPED

Answer the question based on the given context. Each question has more than one answer. Please predict the number of answers first, then give all the answers and separate them with a semicolon.
Context: Laura Horton is a fictional character from the NBC soap opera , Days of Our Lives , a long - running serial drama about working class life in the fictional , United States town of Salem . Created by writer Peggy Phillips , the role was originated by actress Floy Dean on June 30 , 1966 till October 21 , 1966 . Susan Flannery stepped into the role from November 22 , 1966 to May 27 , 1975 . Susan Oliver briefly stepped into the role from October 10 , 1975 , to June 9 , 1976 , followed by Rosemary Forsyth from August 24 , 1976 , to March 25 , 1980 .
Question: who played laura horton on days of our lives
Answers: The number of answers is 4: Floy Dean; Susan Flannery; Susan Oliver; Rosemary Forsyth

Following the example above and answer the following multi-answer question. Please predict the number of answers first, then give all the answers and separate them with a semicolon.
Context: `{context}`
Question: `{question}`
Answers:

Table 14: The one-shot prompts for GPT-3.5 to answer multi-answer questions in MultiSpanQA.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The Limitation Section

☒ A2. Did you discuss any potential risks of your work?
The dataset annotation and the methods in this work do not pose any ethical or security-related risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4, 5

☑ B1. Did you cite the creators of artifacts you used?
Section 4, 5

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix F

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Our annotation is a categorization of the instances in previously published datasets, which have been peer-reviewed and are publicly available.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4, Appendix B

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 4, Appendix B

C ☑ Did you run computational experiments?

Section 5, 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5, Appendix D

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5, Appendix D

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5, Appendix D

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5, Appendix D

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Section 4

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
The instructions are stated in Section 4.2. The dataset in this work do not pose any ethical or security-related risks.

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Appendix B

☑ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Appendix B

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Ethics review is not required for this research in the country where this work is carried out. We carefully checked that there are no ethical problems in our research.

☑ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Appendix B