Complex Reading Comprehension Through Question Decomposition

Xiao-Yu Guo , Yuan-Fang Li , and Gholamreza Haffari

Faculty of Information Technology, Monash University, Melbourne, Australia
{xiaoyu.guo,yuanfang.li,gholamreza.haffari}@monash.edu

Abstract

Multi-hop reading comprehension requires not only the ability to reason over raw text but also the ability to combine multiple evidence. We propose a novel learning approach that helps language models better understand difficult multi-hop questions and perform "complex, compositional" reasoning. Our model first learns to decompose each multi-hop question into several sub-questions by a trainable *question decomposer*. Instead of answering these sub-questions, we directly concatenate them with the original question and context, and leverage a *reading comprehension* model to predict the answer in a sequence-to-sequence manner. By using the same language model for these two components, our best *seperate/unified* t5-base variants outperform the baseline by 7.2/6.1 absolute F1 points on a hard subset of DROP dataset.

1 Introduction

Multi-hop Reading Comprehension (RC) is a challenging problem that requires compositional, symbolic and arithmetic reasoning capabilities. Facing a difficult question, humans tend to first decompose it into several sub-questions whose answers can be more easily identified. The final answer to the overall question can then be concluded from the aggregation of all sub-questions' answers. For instance, for the question in Table 1, we can naturally decompose it into three simpler sub-questions (1) "return the touchdown yards", (2) "return the fewest of #1", and (3) "return who caught #2". The tokens #1 and #2 are the answers to the first and second sub-questions respectively. Finally, the player with the touchdown of #2 is returned as the final answer.

State-of-the-art RC techniques employ large-scale pre-trained language models (LMs) such as GPT-3 (Brown et al., 2020) for their superior representation and reasoning capablities. Chain of thought prompting (Wei et al., 2022) elicits strong reasoning capability of LMs by providing intermediate reasoning steps. Least-to-most prompting (Zhou et al., 2022) further shows the feasibility of conducting decomposition and multi-hop reasoning, which happen on the decoder side together with the answer prediction procedure. However, compared to supervised learning models, both of these methods rely on extremely large LMs with tens and hundreds of **billions** of parameters to achieve competitive performance, thus requiring expensive hardware and incurring a large computation footprint.

C	First, Detroit's Calvin Johnson caught a 1-yard pass in the third quarter. The game's final points came when Mike Williams of Tampa Bay caught a 5-yard.
Q	Who caught the touchdown for the fewest yards?
$\mathbf{Q_1}$	return the touchdown yards
$\mathbf{Q_2}$	return the fewest of #1
$\mathbf{Q_3}$	return who caught #2
A	Calvin Johnson

Table 1: An example for reading comprehension. **C** is the context, **Q** is a hard multi-hop question, and $\mathbf{Q_1}$, $\mathbf{Q_2}$, $\mathbf{Q_3}$ are sub-questions annotated in BREAK dataset. **A** is the answer to **Q**.

Despite significant research on RC (Dua et al., 2019; Perez et al., 2020), those questions that require strong compositional generalisability and numerical reasoning abilities are still challenging to even the state-of-the-art models (Ran et al., 2019; Chen et al., 2020a,b; Wei et al., 2022; Zhou et al., 2022). While decomposition is a natural approach to tackle this problem, the lack of sufficient ground-truth sub-questions limits our ability to train RC models based on large LMs.

In this paper, we propose a novel low-budget

(only 1‰ parameters of GPT-3) learning approach to improve LMs' performance on hard multi-hop RC such as the Break subset of DROP (Dua et al., 2019). Our model consists of two main modules: (1) an encoder-decoder LM as a *question decomposer* and (2) another encoder-decoder LM as the *reading comprehension* model. First, we train the question decomposer to decompose a difficult multi-hop question to sub-questions from a limited amount of annotated data. Next, instead of solving these sub-questions, we train the reading comprehension model to predict the final answer by directly concatenating the sub-questions with the original question. We further propose a *unified* model that utilizes the same LM for both question decomposition and reading comprehension with task-specific prompts. With 9× weakly supervised data, we design a Hard EM-style algorithm to iteratively optimise the *unified* model.

To prove the effectiveness of our approach, we leverage two different types of LMs: T5 (Raffel et al., 2020) and Bart (Lewis et al., 2020) to build baselines and our variants. The experimental results show that without changing the model structure, our proposed variant outperforms the end-to-end baseline. By adding ground-truth sub-questions, gains on the F1 metric are 1.7 and 0.7 using T5 and Bart separately. Introducing weakly supervised training data can help improve the performance of both *separate* and *unified* variants by at least 4.4 point on F1. And our method beats the state-of-the-art model GPT-3 by a large margin.

2 Related Work

Multi-hop Reading Comprehension mentioned in this paper requires more than one reasoning or inference step to answer a question. For example, multi-hop RC in DROP (Dua et al., 2019) requires numerical reasoning such as addition, subtraction. To address this problem, Dua et al. proposed a number-aware model NAQANet that can deal with such questions for which the answer cannot be directly extracted. NumNet (Ran et al., 2019) leveraged Graph Neural Network to design a number-aware deep learning model. QDGAT (Chen et al., 2020a) distinguished number types more precisely by adding the connection with entities and obtained better performance. Nerd (Chen et al., 2020b) searched possible programs exhaustively based on the ground-truth and employed these programs as weak supervision to train the whole model.

Question Decomposition is the approach that given a complex question, break it into several simple sub-questions. These sub-questions can also be Question Decomposition Meaning Representation (QDMR) (Wolfson et al., 2020) for complex questions. Many researchers (Perez et al., 2020; Geva et al., 2021) have been trying to solve the problem by incorporating decomposition procedures. For example, Perez et al. (2020) propose a model that can break hard questions into easier sub-questions. Then, simple QA systems provide answers of these sub-questions for downstream complex QA systems to produce the final answer corresponding to the original complex question. Fu et al. (2021) propose a three-stage framework called Relation Extractor Reader and Comparator (RERC), based on complex question decomposition. Different from these approaches, we aim to improve the multi-hop capability of current encoder-decoder models without dedicated pre-designing the architecture.

Language Models like BERT (Devlin et al., 2019), GPT families (Radford et al., 2018, 2019; Brown et al., 2020), BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) are demonstrated to be effective on many NLP tasks, base on either fine-tuning or few-shot learning (Wei et al., 2022; Zhou et al., 2022), even zero-shot learning. However, LMs suffer a lot from solving multi-hop questions and logic reasoning and numerical reasoning problems. Although some research (Nye et al., 2021; Wei et al., 2022) has conducted experiments on either simple or synthetic datasets and shown the effectiveness, Razeghi et al. (2022) indicates that the model reasoning is not robust enough.

Recently, Dohan et al. (2022) points out that prompted models can be regarded as employing a unified framework a *language model cascade*. From the perspective view of probabilistic programming, several recent literature (Wei et al., 2022; Zhou et al., 2022) are formalized. In this paper, we also treat our whole process as a probabilistic model that is consistent to Dohan et al. (2022).

3 Complex Question Answering Through Decomposition

Our focus in this work is on complex questions requiring multi-hop reasoning. As such, our approach consists of the following two steps:

1. The complex question is decomposed to a sequence of sub-questions. The decomposition

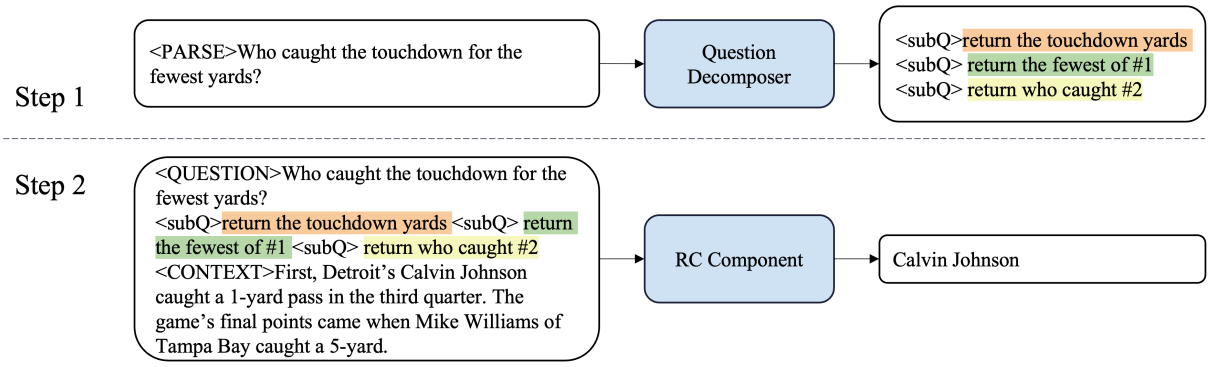

Figure 1: Our model structure on complex reading comprehension through question decomposition. Step 1: Question Decomposer generates a sequence of sub-questions; Step 2: RC component predicts the answer based on question, sub-questions and the given context. The context of this given example is truncated.

of the question is performed by the *question decomposer* component of our system.

2. The model produces the answer to the complex question leveraging the generated sub-questions to provide guidance to the reasoning of the system. This is performed by the *reading comprehension* component.

We use LMs such as T5 and Bart as the backbone[1] for both question decomposer and the reading comprehension (Figure 1). We present several variants of our model, depending whether the models for the above two steps are either separate or unified using multitask learning. As we have the ground truth question decomposition for only a subset of the training data, we treat the missing decompositions as latent variables. We then propose an algorithm based on Hard-EM (Neal and Hinton, 1998) for learning the model. The rest of this section provides more details.

Probabilistic Model. Given a question Q and a C context pair, our system generates the answer A according to the following probabilistic model:

$$P_\theta(A|Q,C) = \sum_Z P_\theta(A,Z|Q,C) \quad (1)$$

$$= \sum_Z P^{\text{dc}}_{\text{LM}}(Z|Q) \times P^{\text{rc}}_{\text{LM}}(A|Q,C,Z) \quad (2)$$

where Z denotes the unobserved decomposition of the question, $P^{\text{dc}}_{\text{LM}}(Z|Q)$ [2] denotes the question decomposer (operationalised based on one specific LM), and $P^{\text{rc}}_{\text{LM}}(A|Q,C,Z)$ denotes the reading comprehension component. In principle, the $P^{\text{dc}}_{\text{LM}}$ and $P^{\text{rc}}_{\text{LM}}$ components can be constructed using different models, so the parameters θ of the whole probabilistic model consists of those for these two models. This is denoted by the *separate* variant.

We further investigate using the same LM for both the question decomposer and reading comprehension component, which we denote by the *unified* variant in the experiments. In this case, the probabilistic model parameter θ consists of only one set of parameters corresponding to the underlying model.

Question Decomposer. To obtain high-quality sub-questions, we first train a question decomposer $P^{\text{dc}}_{\text{LM}}$ to break down difficult multi-hop questions, i.e., the first term in Equation 2. It learns the decomposition based on QDMRs (Wolfson et al., 2020). We only use the specific partition on the DROP dataset (Dua et al., 2019) and treat QDMRs as sub-questions. These sub-questions only cover around 10% QA pairs in DROP. Therefore, we need to predict decompositions for the rest of the dataset. More details will be revealed in Section 4.

Formally, given a multi-hop question Q, the question decomposer $P^{\text{dc}}_{\text{LM}}$ generates the sub-questions $Z := \{Q^1, Q^2, ..., Q^s\}$. Intuitively, We treat it as a seq2seq learning problem: our input to the encoder is "<PARSE>Q", where <PARSE> is a special token. The decoder then generates tokens of the sub-questions in auto-regressive way "<subQ>Q^1<subQ>Q^2<subQ>$\ldots Q^s$", where <subQ> is a special token [3].

[1] Our approach is general, and it can be used with other pre-trained seq2seq models and language models as well.

[2] We have made the following independence assumption: $P^{\text{dc}}_{\text{LM}}(Z|Q) \approx P^{\text{dc}}_{\text{LM}}(Z|Q,C)$.

[3] We employ the greedy search algorithm to generate the sub-questions Z. However, one can leverage other strategies like beam search to make more than one predictions.

Algorithm 1 Learning with Hard-EM

Require: an initial pre-trained LM M; the full reading comprehesion dataset $\mathcal{D}_1$; the subset with sub-question annotations $\mathcal{D}_2$.

1: Train M on $\mathcal{D}_2$ to get M^0
2: **for** iter **in** N_iters **do**
3: For all $\mathcal{D} = \mathcal{D}_1 \setminus \mathcal{D}_2$ employ M^{iter-1} to predict sub-questions and get $\mathcal{D}^{iter}$
4: Retrain M^{iter-1} on all examples: $\mathcal{D}_2 \cup \mathcal{D}^{iter}$, get updated model M^{iter}
5: **end for**

Reading Comprehension Component. To further obtain answers based on the question and generated sub-questions, the reading comprehension component $P^{\text{rc}}_{\text{LM}}$ generates the answer A, i.e., the second term in Equation 2. In stead of directly answering all the sub-questions given by the trained question decomposer, we train our RC component to predict the final answer in a sequence-to-sequence way.

Formally, given a multi-hop complex question Q and the corresponding sub-questions $Z := \{Q1, Q2, ..., Q^s\}$ generated by a trained question decomposer, our input to the RC encoder is "<QUESTION>Q<subQ>Q^1 ... <subQ>Q^s<CONTEXT>C", where <QUESTION> and <CONTEXT> are special tokens. In other words, we concatenate the multi-hop question and all the sub-questions, together with the context as the input to our RC component. The decoder then generates the tokens of the answer autoregressively.

Training and Inference. The training objective of our model is

$$\mathcal{L} = \sum_{(Q,C,A)\in\mathcal{D}_1\setminus\mathcal{D}_2} \log P_\theta(A|Q,C) + \sum_{(Q,C,Z^*,A)\in\mathcal{D}_2} \log P_\theta(A, Z^*|Q,C), \quad (3)$$

where Z^* denotes the ground truth decomposition available only for the subset of the training data referred to by $\mathcal{D}_2$. The first term of the training objective involves enumerating over all possible latent decompositions, which is computationally intractable. Therefore, we resort to Hard-EM for learning the parameters of our model (see Algorithm 1) for the unified variant. We found taking 10 iterations of the Hard-EM algorithm to be mostly sufficient for learning model parameters in our experiments.

Proportions		1%	5%	10%	50%	100%
BLEU		39.08	44.76	47.74	50.12	**54.69**
Rouge-1		77.49	81.75	83.12	84.76	**85.67**
Rouge-2		57.00	62.83	64.97	66.94	**68.61**
RougeL		67.78	72.65	74.37	76.55	**77.43**
RC	EM	26.0	26.5	27.0	**27.8**	27.2
	F1	31.3	31.3	31.6	**32.2**	32.0

Table 2: Experimental results of the Bart based question decomposer: (1) Row 1-4 show intrinsic metrics for the question decomposition by using different proportions of training instances. (2) Row 5-6 show extrinsic metrics of the RC model by using the corresponding decomposer generated sub-questions.

For the separate variant, i.e., using two different LMs for $P^{\text{dc}}_{\text{LM}}$ and $P^{\text{rc}}_{\text{LM}}$, we train the question decomposer on $\mathcal{D}_2$, and then train the reading comprehension component on $\mathcal{D}_2$ as well as $\mathcal{D}_1 \setminus \mathcal{D}_2$ augmented with the generated decomposition Z. We also compare with training the reading comprehension component on $\mathcal{D}_2$ only, in the experiments. During inference time, we first generate the question decomposition $\tilde{Z}$ according to $P^{\text{dc}}_{\text{LM}}$, and then use $\tilde{Z}$ in $P^{\text{rc}}_{\text{LM}}$ to generate the answer.

4 Experiments

4.1 Dataset

We consistently use the same notations as in Algorithm 1.

- $\mathcal{D}_1$: the DROP dataset (Dua et al., 2019) that contains 77,400/9,536 question (Q) answer (A) training/testing pairs for the reading comprehension component.
- $\mathcal{D}_2$: the BREAK dataset (Wolfson et al., 2020) [5] that contains 7,683/1,268 question (Q) decomposition (Z^*) training/testing pairs for the question decomposer [6].
- $\mathcal{D} = \mathcal{D}_1 \setminus \mathcal{D}_2$: the difference set between $\mathcal{D}_1$ and $\mathcal{D}_2$ that contains only question answer pairs without ground-truth decomposition.
- $\mathcal{D}^{iter}$: $\mathcal{D}$ with decomposition (Z) generated by the trained question decomposer.

[5] The full BREAK dataset Wolfson et al. (2020) annotated is a combination of many datasets including DROP. In this paper, we only use the DROP partition of the original BREAK.

[6] This subset of DROP contains the corresponding answers for each question. Therefore, we also use it to evaluate the RC component in our experiments.

LMs		t5-small					t5-base				
Proportions		1% [4]	5%	10%	50%	100%	1%	5%	10%	50%	100%
BLEU		11.21	44.50	50.44	60.15	62.73	34.86	52.98	57.3	62.18	**64.40**
Rouge-1		43.00	76.93	81.53	87.25	88.59	70.66	84.16	85.77	88.50	**89.27**
Rouge-2		28.18	59.13	64.33	72.60	74.76	50.57	66.86	70.24	74.24	**75.72**
RougeL		39.22	68.92	73.66	79.99	81.57	62.10	75.49	78.07	81.20	**82.53**
RC	EM	-	28.9	29.9	29.0	29.0	33.7	34.3	34.3	34.6	**34.8**
	F1	-	33.0	34.0	33.2	33.1	37.8	38.4	38.5	38.5	**38.6**

Table 3: Results of the T5 based question decomposer (left-half: t5-small, right-half: t5-base): (1) Row 1-4 show all intrinsic metrics to evaluate the question decomposer by using different proportions of training instances. (2) Row 5-6 show extrinsic metrics of the RC component by using the corresponding decomposer generated sub-questions.

Note that every question (Q) is associated with a specific context (C). With all question decomposition labelled, $\mathcal{D}_2$ is actually a subset of $\mathcal{D}_1$ and is more challenging.

4.2 Backbone and Evaluation Metric

There are three LMs of different types and sizes we employ as backbones in this paper: (1) t5-small (60M parameters), (2) t5-base (220M parameters), (3) bart-base (140M parameters). We also employ GPT-3 (175B parameters) as it is the current state-of-the-art language model in a various of natural language processing tasks.

Sub-question Decomposition We train and evaluate our question decomposer using $\mathcal{D}_2$, which was proposed to better understand difficult multi-hop questions. We report BLEU (Papineni et al., 2002) and Rouge (Lin, 2004) scores to show the intrinsic performance of the decomposer.

Reading Comprehension We evaluate our RC model on $\mathcal{D}_2$. For the Hard-EM approach, we have $\mathcal{D}_1 \setminus \mathcal{D}_2$ as weakly supervised data. We report F1 and Exact Match(EM) (Dua et al., 2019) scores in the following experiments.

4.3 Results on Decomposition

Based on Bart and T5, Table 2 and Table 3 respectively show the experimental results of the question decomposers. To comprehensively show their performance, we conducted two aspects of experiments including intrinsic decomposition evaluation and extrinsic RC evaluation.

Intrinsic Evaluation We first evaluate the quality of sub-questions generated by different question decomposers. In this part, intrinsic metrics, BLEU and Rouge scores, are shown in the first four rows of Table 2 and Table 3. And also we show the results of five decomposers trained on different proportions (1%, 5%, 10%, 50%, 100%) of the BREAK dataset $\mathcal{D}_2$'s training data. All these evaluations are conducted on the same validation set of $\mathcal{D}_2$.

Comparing column-by-column, we find that with more training data, both question decomposers achieve a better performance for both BLEU and Rouge. We also note that the rate of improvement of these metrics becomes slower when more data is added (e.g. 1% to 5% and 10% to 50%). Therefore, we posit that with more training data, the performance of the decomposer will not improve due to the capability of the LM model.

Extrinsic Evaluation Since the eventual usage of the generated sub-questions is to improve the RC component, we conduct a RC performance comparison experiments to see how can the quality of these sub-questions influence the downstream RC task. Also like the intrinsic evaluation, we show the results based on decomposers trained on different proportions of $\mathcal{D}_2$ by using two extrinsic metrics: EM and F1. All the evaluations are conducted on the same validation set of $\mathcal{D}_2$.

To clarify our settings in this part, we don't employ the ground-truth sub-questions from $\mathcal{D}_2$. Instead, we employ the sub-questions generated by five question decomposers for the RC component to predict answers. As the last two rows of both Table 2 and Table 3 show, both EM and F1 scores show a gradually increasing trend when more training instances are used to train the question decomposer. With more parameters, t5-base tends to have a better performance than t5-small.

4.4 Results on Reading Comprehension

Table 4 shows the experimental results for the downstream RC task. We show two baselines in the first place: "bart-base" and "t5-base". Without taking sub-questions as input, both are trained on the

Backbone	Variant	Training Set	F1	EM
baselines				
bart-base (Lewis et al., 2020)	-	$\mathcal{D}_2$	30.9	27.1
t5-base (Raffel et al., 2020)	-	$\mathcal{D}_2$	37.9	33.9
our bart-base variants				
w/ predicted sub-questions	*separate*	$\mathcal{D}_2$	32.0	27.2
w/ ground-truth sub-questions	*separate*	$\mathcal{D}_2$	33.2	29.0
w/ ground-truth sub-questions	*separate*	$\mathcal{D}_2, \mathcal{D}^1$	**45.0**	**40.5**
w/o Hard-EM	*unified*	$\mathcal{D}_2, \mathcal{D}^1$	44.2	39.9
w/ Hard-EM	*unified*	$\mathcal{D}_2, \mathcal{D}^{iter}$	44.3	40.0
our t5-base variants				
w/ predicted sub-questions	*separate*	$\mathcal{D}_2$	38.6	34.8
w/ ground-truth sub-questions	*separate*	$\mathcal{D}_2$	39.6	35.6
w/ ground-truth sub-questions	*separate*	$\mathcal{D}_2, \mathcal{D}^1$	**45.1**	**40.8**
w/o Hard-EM	*unified*	$\mathcal{D}_2, \mathcal{D}^1$	38.8	34.9
w/ Hard-EM	*unified*	$\mathcal{D}_2, \mathcal{D}^{iter}$	44.0	40.1
GPT-3 (zero-shot)	-	-	15.7	4.6
GPT-3 (few-shot)	-	-	34.9	27.0

Table 4: Overall results for baselines, our separate and unified variants. All models are evaluated on the same test set from $\mathcal{D}_2$.

BREAK dataset $\mathcal{D}_2$. Based on these vanilla models, we show our *separate* and *unified* approaches that use "bart-base" and "t5-base" as backbones separately in Table 4.

4.4.1 Separate Variant

Our *separate* variants are base on the architecture in Figure 1. In Table 4, we have three *separate* variants based on each backbone for comparison. Taking t5-base as one example, comparing to the t5-base, using predicted sub-questions achieves a 0.7-point gain of F1 score. Meanwhile using ground-truth sub-questions, our model outperforms the t5-base by 1.7 points of F1 score. The same improvement can be also concluded from the bart-base model. They employ $\mathcal{D}_2$ for training but their testing sets are different: predicted one use generated sub-questions while ground-truth one use sub-questions from D_2. The reason why our approach is more effective than the baseline model is that concatenating sub-questions can give LMs hints on the reasoning procedure, which helps LMs produce step-by-step thoughts implicitly.

Furthermore, we add $\mathcal{D}^1$ as the training set to train our seperate model. As it shows in Table 4, this kind of *separate* variants show the overall best performance since we have two sets of parameters separately learning question decomposition and reading comprehension. Compared to t5-base, the bart-base variant shows a higher performance gain that proves the effectiveness of our method.

4.4.2 Unified Variant

Our *unified* variants are base on the architecture in Figure 1 and one single model is used to train on both steps. In Table 4, the last two rows of each variants show the performance of our *unified* variant. Without the Hard-EM algorithm, performing multi-task learning achieves a 0.9 point improve over the T5 baseline. However, it shows a performance drop when compared to the *separate* variant with ground-truth sub-questions. This can be caused by the enlarged dataset and the additional decomposition work the *unified* variant need to handle.

When more training data is provided (i.e. $\mathcal{D}^1$ and $\mathcal{D}^{iter}$), though without ground-truth sub-questions, the *unified* variants substantially outperforms the baselines by 10.1 and 6.1 points over bart-base and t5-base model. Furthermore, when compared with the best *separate* variants, our *unified* models also show comparable performance on both F1 and EM metrics. Based on the observations of the last three rows of each backbone, it can be concluded that introducing more weakly-supervised training data can significantly help our model address the original difficult multi-hop RC task.

We also include another evaluation of employ-

Context	Question	GPT-3 (few-shot)	bart-base *separate* (best)	ground-truth answer
... notably striking out Julio Franco, at the time the oldest player in the MLB at 47 years old; Clemens was himself 43. In the bottom of the eighteenth inning, Clemens came to bat again...	Which player playing in the 2005 National League Division Series was older, Julio Franco or Roger Clemens?	Julio Franco (✓)	Julio Franco (✓)	Julio Franco
... Nyaungyan then systematically reacquired nearer Shan states. He captured Nyaungshwe in February 1601, and the large strategic Shan state of Mone in July 1603, bringing his realm to the border of Siamese Lan Na. In response, Naresuan of Siam marched in early 1605 to ...	How many years after capturing Nyaungshwe did Nyaungyan capture the large strategic Shan state of Mone?	3 years (✗)	2 (✓)	2
Kannada language is the official language of Karnataka and spoken as a native language by about 66.54% of the people as of 2011. Other linguistic minorities in the state were Urdu (10.83%), Telugu language (5.84%), Tamil language (3.45%), ...	How many in percent of people for Karnataka don't speak Telugu?	66.54% (✗)	94.04% (✗)	94.16%
A 2013 analysis of the National Assessment of Educational Progress found that from 1971 to 2008, the size of the black-white IQ gap in the United States decreased from 16.33 to 9.94 IQ points. It has also concluded however that, ...	How many IQ points did the black-white IQ gap decrease between 1971 and 2008?	16.33 (✗)	0.9 (✗)	6.39

Table 5: Correct and incorrect outputs from GPT-3 and our *separate* variant. Correct and Wrong supporting facts are annotated in the context using the corresponding color. Correct and wrong answer predictions are also marked with ✓and ✗ (the table is best seen in colours).

ing GPT-3, which is the state-of-the-art language model on many tasks and also in a large parameter scale (175B). The results are shown by last two rows in Table 4. Based on the experimental results, GPT-3 cannot even beat two baseline models under the zero-shot learning paradigm, which again shows the complexity and challenging of the task. When provided with several exemplars, it can easily outperform the bart-base model by 2.4 points on F1 score. However, even with $\times 1000$ parameters, GPT-3 is still far behind to our best variants by 10.2 F1 points.

5 Analysis and Discussions

5.1 Qualitative Analysis

In this section, we will further discuss some real-life cases generated by our proposed variants from the dataset. In Table 5, the first row shows a comparison question and both GPT-3 and our bart-base *separate* model can produce the correct answer. However, when the question requires some arithmetic operations, such as addition or subtraction, the GPT-3 model would fail to answer correctly. Our model can handle this as shown by the second row.

There are two types of failures from our variants: one is that our model cannot handle unseen numbers, and the other is arithmetic between float numbers. The unseen number case happens in the third row of Table 5. Asking for the number of a complement set, though the number 94.04% is wrongly predicted by our model, it is more close to the ground-truth (94.16%) when compared to the GPT-3, which directly predict an wrong evidence annotated with red color. Furthermore, the last row shows a subtraction question between two

overlaps		$0 \sim 25\%$	$25\% \sim 50\%$	$50\% \sim 75\%$	$75\% \sim 100\%$
uni-grams	bart-base	-	0	25.7	27.4
	unified	-	0	32.9	40.2
	separate	-	0	**35.7**	**41.3**
	GPT-3	-	**100.0**	**35.7**	26.4
bi-grams	bart-base	-	16.7	23.6	28.2
	unified	-	33.3	**29.1**	41.9
	separate	-	**50.0**	28.6	**43.2**
	GPT-3	-	44.4	**29.1**	26.2
tri-grams	bart-base	22.2	20.5	25.5	29.3
	unified	38.9	26.2	32.3	45.1
	separate	**50.0**	**30.0**	**33.4**	**45.9**
	GPT-3	**50.0**	28.0	25.8	26.8

Table 6: EM scores separately computed based on overlaps of sub-questions n-grams between training set and testing set on $\mathcal{D}_2$. Four models listed in this table are: the bart-base baseline, the best performed *separate* model, the best performed *unified* model

float numbers. Different from integer number subtraction in the second row, it is much harder to compute this arithmetic for language models. Traditionally, some symbolic methods can handle this problem very well. Tackling these problems can be interesting future work directions.

5.2 Quantitative Analysis

We look into details of $\mathcal{D}_2$ from the perspective of sub-question n-grams for both training and testing data. Intuitively, given one instance from the test set, more n-grams overlap it shows with the training set, higher the EM and F1 scores. Therefore, we further conducted the analysis and list all the statistics in Table 6.

We calculate for uni-grams, bi-grams and tri-grams for four models: bart-base baseline, the best-performed *separate* and *unified* variants proposed in Section 3 and GPT-3 with few-shot learning. The overlaps we choose is four intervals using percentages to represent. For example, $0 \sim 25\%$ overlapping on bi-grams means that the test instance have this proportion of bi-grams overlaps with all the training instances. Note that there is no overlapping for uni-grams and bi-grams in $0 \sim 25\%$.

In Table 6, we report the EM score (F1 score shows the similar results). The bart-base model show a tendency that with more overlaps across all n-grams, the performance will increase, which is consistent with our assumption. However, on the contrary, GPT-3 model show a reverse tendency that is probably due to the pre-trained corpus that shares far less n-grams with the test set. This characteristic improves the compositional generalisation ability as it outperforms the baseline model on the low-overlapping part of test set. Both of our *separate* and *unified* variants show overall improvements over the bart-base baseline. In particular, the first and second columns also show our model can better handle the low-overlapping questions, even without performance drop on the high-overlapping questions ($50\% \sim 100\%$). This experiment can further prove the compositional generalisation of our method is comparable to GPT-3.

6 Conclusion

We propose a two-step process for multi-hop reading comprehension task. The first step involves a question decomposer that maps a difficult multi-hop question into several sub-questions. The second step is to train a reading comprehension model based on (question, sub-questions, paragraph, answer) tuples. With the addition of sub-questions, our bart-/t5-base variants outperform the baseline model by 2.3/1.7 using ground truth sub-questions and 1.1/0.7 using generated ones on F1 score. Based on the hard-EM paradigm, large positive gains of another 11.1/4.4 point on F1 by the unified multi-task learning bart-/t5-base models shows the effectiveness of introducing weakly supervised training data. By further analysing the predicted examples and dataset, we also found our model can make a more comprehensive improvement compared with the SOTA GPT-3 model. But some problems like handling unseen numbers still exist and will be our future research directions.

Acknowledgements

This material is based on research sponsored by DARPA under agreement number HR001122C0029. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

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

Kunlong Chen, Weidi Xu, Xingyi Cheng, Zou Xiaochuan, Yuyu Zhang, Le Song, Taifeng Wang, Yuan Qi, and Wei Chu. 2020a. Question directed graph attention network for numerical reasoning over text. In *Proceedings of EMNLP*, pages 6759–6768.

Xinyun Chen, Chen Liang, Adams Wei Yu, Denny Zhou, Dawn Song, and Quoc V. Le. 2020b. Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *Proceedings of ICLR*.

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 NAACL HLT 2019*, pages 4171–4186.

David Dohan, Winnie Xu, Aitor Lewkowycz, Jacob Austin, David Bieber, Raphael Gontijo Lopes, Yuhuai Wu, Henryk Michalewski, Rif A. Saurous, Jascha Sohl-Dickstein, Kevin Murphy, and Charles Sutton. 2022. Language model cascades. *CoRR*, abs/2207.10342.

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 NAACL HLT 2019*, pages 2368–2378.

Ruiliu Fu, Han Wang, Xuejun Zhang, Jun Zhou, and Yonghong Yan. 2021. Decomposing complex questions makes multi-hop QA easier and more interpretable. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 169–180, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mor Geva, Tomer Wolfson, and Jonathan Berant. 2021. Break, perturb, build: Automatic perturbation of reasoning paths through question decomposition. *CoRR*, abs/2107.13935.

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, ACL 2020, Online, July 5-10, 2020*, pages 7871–7880. Association for Computational Linguistics.

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

Radford Neal and Geoffrey E. Hinton. 1998. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *CoRR*, abs/2112.00114.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. *CoRR*, abs/2002.09758.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

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

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

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. NumNet: Machine reading comprehension with numerical reasoning. In *Proceedings of EMNLP-IJCNLP*, pages 2474–2484.

Yasaman Razeghi, Robert L. Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *CoRR*, abs/2202.07206.

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. *CoRR*, abs/2201.11903.

Tomer Wolfson, Mor Geva, Ankit Gupta, Yoav Goldberg, Matt Gardner, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Trans. Assoc. Comput. Linguistics*, 8:183–198.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Olivier Bousquet, Quoc Le, and Ed Chi. 2022. Least-to-most prompting enables complex reasoning in large language models. *CoRR*, abs/2205.10625.