It Is Not About What You Say, It Is About How You Say It: A Surprisingly Simple Approach for Improving Reading Comprehension

Sagi Shaier,[▽] Lawrence E Hunter,[†] Katharina von der Wense[▽◇]
[▽]University of Colorado Boulder
[†]Independent Scholar
[◇]Johannes Gutenberg University Mainz
[▽]E-mail: {sagi.shaier, katharina.kann}@colorado.edu
[†]E-mail: Prof.Larry.Hunter@gmail.com

Abstract

Natural language processing has seen rapid progress over the past decade. Due to the speed of developments, some practices get established without proper evaluation. Considering one such case and focusing on reading comprehension, we ask our first research question: 1) How does the order of inputs – i.e., question and context – affect model performance? Additionally, given recent advancements in input emphasis, we ask a second research question: 2) Does emphasizing either the question, the context, or both enhance performance? Experimenting with 9 large language models across 3 datasets, we find that presenting the context before the question improves model performance, with an accuracy increase of up to 31%. Furthermore, emphasizing the context yields superior results compared to question emphasis, and in general, emphasizing parts of the input is particularly effective for addressing questions that models lack the parametric knowledge to answer. Experimenting with both prompt-based and attention-based emphasis methods, we additionally find that the best method is surprisingly simple: it only requires concatenating a few tokens to the input and results in an accuracy improvement of up to 36%, allowing smaller models to outperform their significantly larger counterparts.

Setting/ Emphasis	*Question: <q> Context: <c>*	*Context: <c> Question: <q>*
Question	Question: **where is the world's largest ice sheet located today.** Context: The Antarctic ice sheet is the largest single mass of ice on Earth [...]	Context: The Antarctic ice sheet is the largest single mass of ice on Earth [...]. Question: **where is the world's largest ice sheet located today.**
Context	Question: where is the world's largest ice sheet located today. Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...]**	Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...].** Question: where is the world's largest ice sheet located today.
Question+ Context	Question: **where is the world's largest ice sheet located today.** Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...]**	Context: **The Antarctic ice sheet is the largest single mass of ice on Earth [...].** Question: **where is the world's largest ice sheet located today.**

Figure 1: Example from the Natural Questions dataset in which we show the different settings we experiment with: question or context first in the input prompt, and the different substring emphasis (in bold). <q>=question string; <c>=context string.

1 Introduction

For the task of reading comprehension (RC), models receive two kinds of inputs: 1) a context, e.g., a Wikipedia article, and 2) a question that should be answered according to the context (Dzendzik et al., 2021; Zeng et al., 2020). While early efforts to address this task usually involve models that encode each of these separately (Zhang, 2019; Tay et al., 2018; Nishida et al., 2019; Clark and Gardner, 2018; Choi et al., 2017), more recently, large language models (LLMs) receive a concatenation of the two inputs (Wen et al., 2022; Huang et al., 2022; Sun et al., 2023; Bahak et al., 2023; Baek et al., 2023; Brown et al., 2020; Chowdhery et al., 2022; Chung et al., 2022).

Surprisingly, **there is no current standard of what the ordering of such input components should be**. For example, Sun et al. (2023); Nori et al. (2023); Bahak et al. (2023); Kamalloo et al. (2023); Singhal et al. (2022); Zhong et al. (2022) provide the question first in each prompt, while Cheng et al. (2023); Nori et al. (2023); Liu et al. (2023a); Baek et al. (2023); Brown et al. (2020); Singhal et al. (2022); Chowdhery et al. (2022); Chung et al. (2022) provide the context first. Moreover, **there is no current standard of how to present the two input components in general**. For example, considering the question and context strings <q> and <c>, respectively, Wen et al. (2022) add the special tokens "question:" and "context:" before the question and context, while Nori et al. (2023) use "<c>**Question:** <q>", Zhong et al. (2022) use "[Question]: <q> [Passage]: <c>", Liu et al. (2023a) use "<c> <q>", and others such as (Baek et al., 2023; Brown et al., 2020; Chowdhery et al., 2022; Chung et al., 2022), employ their own methods.

While at first sight this might not seem impor-

Findings of the Association for Computational Linguistics: ACL 2024, pages 8292–8305

tant, many works have shown that LMs can be extremely susceptible to slight variations in the input sequence (Jia and Liang, 2017; Si et al., 2019; Sen and Saffari, 2020; Shaier et al., 2023). Furthermore, recent research has found that **different presentations of inputs can help emphasize them** and improve models' ability to follow instructions (Zhang et al., 2023). Based on these observations, we ask the following research questions (RQs): 1) How does the order of inputs – i.e., question and context – affect model performance? 2) Does emphasizing either the question, the context, or both enhance performance? A summary of these questions can be seen in Figure 1.

We evaluate 9 LLMs on 3 datasets and find the following: 1) The ordering of the question and context is crucial, and improves model performance with an accuracy increase of up to 31%. 2) Both prompt-based and attention-based emphasis methods are capable of strongly improving models' performance, where emphasizing the context yields superior results compared to emphasizing the question, and in general, emphasizing parts of the input is particularly effective for addressing questions that models lack the parametric knowledge to answer. 3) The best emphasis method is surprisingly simple: it only requires a simple concatenation of a few tokens to the input and results in an accuracy improvement of up to 36%, allowing smaller models to outperform their significantly larger counterparts.

2 Related Work

Reading Comprehension Reading comprehension involves the task of understanding a given context, such as a Wikipedia passage and answering questions based on that context (Dzendzik et al., 2021; Zeng et al., 2020). To that end, researchers develop models capable of comprehending written text and extracting relevant information to accurately respond to queries (Yang et al., 2019; Wang and Pan, 2022; Touvron et al., 2023). Traditional approaches often encode the context and question separately (Zhang, 2019; Tay et al., 2018; Nishida et al., 2019; Clark and Gardner, 2018; Choi et al., 2017), while more recent advancements leverage LLMs that concatenate both inputs into a single string (Wen et al., 2022; Huang et al., 2022; Sun et al., 2023; Bahak et al., 2023; Baek et al., 2023; Brown et al., 2020; Chowdhery et al., 2022; Chung et al., 2022). These models need to possess a deep understanding of the provided context to generate accurate responses to a wide range of questions, and many have shown that they do. Achieving high performance in reading comprehension tasks requires not only effective encoding of textual information, but also sophisticated reasoning and inference abilities to derive answers from the context accurately (Xie and Xing, 2017). Therefore, ongoing research on reading comprehension focuses on improving model architectures (Dhingra et al., 2017; Indurthi et al., 2018; Wang and Pan, 2022; Touvron et al., 2023), training strategies (Gottumukkala et al., 2020; Xu et al., 2019), and evaluation metrics (Yang et al., 2018; Sugawara et al., 2017) to enhance the comprehension and reasoning capabilities of these systems. Here, we address the gap in research focused on how the inputs themselves can impact performance.

Prompt Engineering A related area – prompt engineering (Strobelt et al., 2022; Bach et al., 2022) – focuses on modifying the input prompt to improve the performance of LMs without altering their underlying architecture or training regime. And while LMs require a deep understanding of the provided context to generate accurate responses, recent studies have demonstrated that large performance enhancements can be achieved through prompt engineering alone (Brown et al., 2020; Liu et al., 2021; Wei et al., 2023; Dong et al., 2023). This approach involves various techniques such as adding different input strings (Zhang et al., 2023), providing step-by-step instructions (Wei et al., 2023), or incorporating additional contextual information into the prompt (Brown et al., 2020). By carefully crafting the input prompt, researchers aim to guide the model towards relevant information and improve its ability to comprehend and generate coherent responses.

Emphasis Methods It is important to note that researchers often do not have the ability to precisely guide the model using prompt engineering, and much of prompt development is based on intuition. That is, researchers often have to try many different prompts manually or automatically until they find those that increase performance, and often just for their specific models (Liu et al., 2021; Gao et al., 2021; Webson and Pavlick, 2022). In comparison, recent work on input emphasis, including attention steering (AS) and marked prompting (MP) (Zhang et al., 2023), have shown great success in improving models' ability to follow instructions. These methods aim to guide the focus of models towards various segments of the input sequence, by either adding tokens to the sequence or rescaling attention weights for relevant tokens. AS is closely related to work that avoids modifying models' architectures, or training regime, however, it takes a more direct approach by modifying parts of the input directly by rescaling the attention values of specific heads corresponding to specific tokens.

Interpretability Emphasis methods, such as AS, are also related to model interpretability, which is concerned with understanding the contributions of different model components, and in particular, head attribution (Geva et al., 2023). For example, Meng et al. (2023); Geva et al. (2021); Kobayashi et al. (2023) show that different knowledge from the training data is found within the feedforward layers, while others show that attention heads have similar patterns (Geva et al., 2023).

3 Models

We experiment with 9 different LLMs.

Llama-2-7B and Llama-2-13B Llama-2-7B and Llama-2-13B (Touvron et al., 2023) are LLMs which contain 7 and 13 billion parameters, respectively, and are trained on 2 trillion tokens. We use these models as they perform well on the reading comprehension task (Touvron et al., 2023) and recent work shows that their performance can be improved using emphasis methods (Zhang et al., 2023).

Falcon-7B and Falcon-7B Instruct These two models contain 7 billion parameters each, and are trained on 1.5 trillion tokens (Almazrouei et al., 2023). We opt for these models because they are newer and have demonstrated significant success across various tasks. Additionally, Falcon-7B Instruct comes with an instruct version, enabling us to compare the performance of both variations.

MPT-7B and MPT-7B Instruct These are two LLMs with 7 billion parameters, trained on 1 trillion tokens (MPT, 2023). Chosen for their recent development and proven versatility.

GPT-J-6B GPT-J-6B (Wang and Komatsuzaki, 2021) contains 6 billion parameters and is trained on the Pile dataset (Gao et al., 2020). We use this model as in addition to the fact that it has been shown to perform well on question answering tasks (De Bruyn et al., 2022), it is also often compared again our largest model – Llama-2 (Touvron et al., 2023; Zhang et al., 2023) and recent work shows that its performance can be improved using emphasis methods (Zhang et al., 2023).

GPT-2-XL GPT-2-XL (Radford et al., 2019) a LLM with 1.5 billion parameters and is trained on WebText (Radford et al., 2019). While much smaller than current state-of-the-art models, such as ChatGPT (OpenAI, 2023a) or GPT 4 (OpenAI, 2023b), we experiment with it as many low-resource settings require usage of smaller models.

GPT-2-Large Our last model, GPT-2-Large (Radford et al., 2019), contains 774 million parameters and, similar to GPT-2-XL, is trained on WebText. We use it for similar reasons as those we described in the GPT-2-XL Section.

4 Experiments

4.1 Datasets

We experiment with the following RC datasets:

Natural Questions The natural questions dataset (Kwiatkowski et al., 2019) is comprised of authentic, anonymized, and aggregated queries directed to the Google search engine. Each question is accompanied by an entire Wikipedia page, and a collection of annotated long and short answers. As entire Wikipedia pages exceed many of our models' context lengths, for each question, we use each of the long answers as the context and the corresponding short answers as the gold answers.

We utilize it due to its widespread adoption and popularity within the research community, ensuring the reproducibility and comparability of our results with existing studies. Additionally, its comprehensive coverage of diverse question types and real-world contexts allows us to further evaluate whether our findings generalize.

Stanford Question Answering Dataset (SQuAD) SQuAD (Rajpurkar et al., 2016) is composed of questions that are gathered from crowdworkers who ask questions about Wikipedia articles. We choose to use it for similar reasons described as the Natural Questions dataset.[1]

[1]We use the 1.0 version instead of the 2.0 version, as the later version contains empty strings as labels for its irrelevant contexts, which prevents us from using the closed-book setting to determine its parametric knowledge (see Section 4.5).

AdversarialQA The AdversarialQA dataset (Bartolo et al., 2020) has been constructed adversarially, based on 3 models-in-the-loop. More specifically, the authors use the same SQuAD annotation methodology and models trained on it, and explore an annotation setting where annotators are tasked with formulating questions for which the model yields incorrect predictions. Consequently, the dataset is composed solely of instances where models answer inaccurately. While not as popular as SQuAD or the Natural Questions, we utilize this dataset as this annotation methodology makes these questions unique and especially challenging.

Data Splits As the test set for each of these datasets is either private or does not contain gold answers, we randomly split the validation sets into two parts and use one half as our validation set and the other as our held-out test set. This results in roughly the following split for each dataset. Natural Questions: $307k$ train, 3915 validation, 3915 test, SQuAD: $87k$ train, 5285 validation, 5285 test, AdversarialQA: $30k$ train, 1500 validation, 1500 test.

4.2 Prompt Structure

RC datasets consist of question, context, and answer triples (q, c, a), where $q \in Q$, $c \in C$, $a \in A$. As outlined above, our RQ1 is concerned with the order in which the question and context are provided to the model: since previous work has been inconsistent in this regard, we explore which order (if any) results in higher performance.

Concretely, we compare the following two prompt structures (cf. Figure 1):

Question First Here, the question comes first in the prompt. In our concrete format, this results in the input sequence

`Question: <q> Context: <c>,`

where q and c are pairs of question and context strings, $q \in Q$, $c \in C$.

Context First In this setting, the context is the first part of the prompt. In our concrete format, this results in the input sequence

`Context: <c> Question: <q>,`

where, again, q and c are question–context pairs, $q \in Q$, $c \in C$.

4.3 Emphasis Strategies

Marked Prompting MP (Zhang et al., 2023) is a simple prompt-based approach in which we append a string to the input sequence in order to emphasize it. For example, to emphasize the questions, we can append the string **“ * ”** to

Question: <q> Context: <c>

which would result in

Question: *<q>* Context: <c>

We experiment with 4 MP methods, composed of the following start and end string pairs: [* and *, “ and ”, <emphasize> and <\emphasize>, <mark> and <\mark>]

Attention Steering In comparison to MP, AS is a more computationally-intensive method to emphasize input tokens and is attention-based.

We follow Zhang et al. (2023)'s approach known as PASTA, which requires 1) an LLM with L stacked layers, each with N multi-head attention (MHA) submodules, such as most transformer-based models (Vaswani et al., 2017); 2) input text W, and 3) a segment $w \in W$ that is found within the input text.

PASTA is composed of two parts:

1) *Attention steering*: in this part, we down-weight the attention scores of any token that is not part of the segment w, by multiplying them with a small scalar $0 \leq \alpha < 1$ for a selected $n \in N$ MHA submodules. In our experiments, we use $\alpha = 1e^{-3}$ based on Zhang et al. (2023).

2) *Model profiling*: here, we select which $n \in N$ to apply the AS to. While the original paper experiments with several selection methods, such as applying the steering to all heads, single heads, or entire layers, they obtain the best performance when selecting the intersection of the top-k best performing heads across several datasets. They select k from a small number of options, such as $\{300, 400, 500\}$ for Llama 7B. However, we find that we can improve performance by increasing this range.

In particular, from each dataset's *training split* D_{ti}, we take a small subset of examples $d_{ti} \in D_{ti}$, and apply AS to each head individually. In our experiments, we use $|d_{ti}| = 1000$ for GPT-2 large and XL, and $|d_{ti}| = 500$ for GPT-J and Llama-2, for computational reasons, after manually assessing different values which result in roughly similar models' scores. We store the performance of the model for each head, which results in $L * N$ scores for each d_{ti}. Next, on each dataset's *validation split* D_{vi} we iteratively select a k, where $0 < k \leq N * L$, and find the intersection of the top-k performing heads across all datasets $d_{ti} \in D_{ti}$. We store the scores, which results in $L * N$ scores for each k for each D_{vi}. For the test split, we use the best k based on the validation split.

Baseline: No Emphasis As a baseline, we further compare to a setting in which we do not emphasize any string and use the original prompt from Section 4.2 as inputs to the models.

4.4 Hyperparameters

We use a maximum sequence length of 512. Truncation due to this might result in an unfair comparison between the different prompt structures as either question or context might get truncated.[2]

In order to avoid this, we remove sequences that are longer than 512 tokens (about 15% of the examples in the Natural Questions dataset, less than 1% for SQuAD, and 0% for AdversarialQA).

4.5 Metrics

Accuracy Following Liu et al. (2023b); Kandpal et al. (2023); Mallen et al. (2023), we assess the performance of all models using accuracy, determining if any of the gold responses are present in the predicted output. Concretely, we feed the two prompts described in Section 4.2, such as *"Question: <q>. Context: <c>"*, to each of the models, and evaluate whether the gold label answer exists within the LLM generated answer.[3]

Context-free Accuracy We are further interested in evaluating the models' parametric knowledge. For this, we follow work by Shaier et al. (2024); Li et al. (2022); Xie et al. (2023); Roberts et al. (2020), who use a closed-book setting to evaluate models' parametric knowledge. In particular, we define *known knowledge* as questions that models answers correctly without the corresponding context and *unknown knowledge* as those they cannot.

[2]See Section 6.5 for an analysis of models with a larger context length.

[3]While this approach is popular, it is important to note that no existing evaluation metric is flawless. For instance, this approach may overlook accurate responses (e.g., because they are not an exact match to gold answers) or erroneously categorize incorrect responses as correct. To address this concern, we supplement our evaluation process by manually inspecting 100 responses from Llama 2 on the Natural Questions dataset in the no emphasis, context-first setting, to evaluate the frequency of such occurrences. We find that while this approach identifies 58.1% of the answers as correct, manual analysis identifies 82%. This highlights the gap between this popular method and human evaluation.

Perplexity Perplexity (PPL) is defined as the exponentiated average of the negative log-likelihood of a sequence. Concretely, given a sequence of tokens $X = (x_0, x_1, ..., x_t)$, the perplexity of X denoted as

$$PPL(X) = exp(-\frac{1}{t}\sum_i^t logp_\theta(x_i \mid x_{<i}))$$

where $logp_\theta(x_i \mid x_{<i})$ represents the log-likelihood of the i-th token conditioned on the preceding tokens $x_{<i}$ according to the model.

5 Results

5.1 RQ 1: Question First vs. Context First

We first analyze whether models' performance differs when given the same information, but in different order: question-first and context-first. Our results can be seen in Table 1.

No Emphasis Accuracy As we aim to understand the effect that prompt structure alone has on models' performance, for this analysis we focus on the no emphasis (NE) baseline.

Looking at the NE setting, there is a clear difference across almost all models and datasets. More specifically, prompting models with the context first strongly improves performance, with an average increase of 13.46% (49.90% in comparison to 36.44%). On the Natural Questions dataset, the highest accuracy change occurs for GPT-J: from 33.3% to 64.5% (31.2% difference). The second highest change is seen for GPT-2-XL: from 28.0% to 51.2% (23.2% difference). The third highest change occurs for Llama-2, which scores 46.3% when the question is given first but 58.1% when the context is given first (11.8% difference). Similar behavior can be seen for the SQuAD and AdversarialQA datasets as well. For example, Llama-2 changes from 60.4% to 72.9% on SQuAD, and GPT-2-XL changes from 24.8% to 31.8% on AdversarialQA. However, we do find two cases where placing the context first does not improve the results, and actually slightly reduces them: on the AdversarialQA dataset, GPT-2 large and GPT-J change from 27.7% to 26.9% and 47.2% to 46.2%, respectively.

5.2 RQ 2: Emphasis and Performance

We next analyze whether emphasizing parts of the input – the question, the context, or both – enhances models' performance. Our results can be seen again in Table 1.

Model	Emphasis Method		Natural Questions								SQuAD								AdversarialQA							
			Question First				Context First				Question First				Context First				Question First				Context First			
			No Emphasis	Emphasis			No Emphasis	Emphasis			No Emphasis	Emphasis			No Emphasis	Emphasis			No Emphasis	Emphasis			No Emphasis	Emphasis		
				Q	C	Q + C		Q	C	Q + C		Q	C	Q + C		Q	C	Q + C		Q	C	Q + C		Q	C	Q + C
Llama-2	B		46.3				58.1				60.4				72.9				42.6				49.4			
	AS			54.8	53.0	-		57.8	59.3	-		66.3	62.0	-		74.5	72.9	-		43.3	43.0	-		54.4	53.3	-
	MP	⋆		51.4	31.6	53.1		58.3	56.4	58.8		56.8	61.7	67.9		69.1	76.4	79.7		40.5	43.2	46.7		51.1	54.2	57.5
		"		48.7	54.2	54.2		56.4	58.2	59.9		61.4	71.9	72.3		72.5	76.3	78.6		42.0	48.3	48.9		50.2	56.8	56.0
		<mark>		51.7	54.1	55.1		60.0	55.5	60.5		53.3	71.5	71.8		75.4	71.3	80.4		39.0	47.3	49.3		50.7	52.4	**57.7**
		<emphasize>		47.6	54.4	53.9		**61.3**	55.5	60.2		53.8	72.2	68.0		78.1	70.4	**81.5**		37.8	49.3	46.5		51.4	50.4	56.2
GPT-J	B		33.3				64.5				45.5				61.0				47.2				46.2			
	AS			66.3	66.3	-		61.1	53.0	-		51.0	44.6	-		55.8	54.1	-		45.0	37.8	-		41.6	41.7	-
	MP	⋆		33.4	26.9	49.7		60.5	65.1	64.9		38.0	52.5	41.7		51.1	64.0	50.5		38.2	52.0	40.8		40.2	50.0	38.2
		"		39.0	63.0	62.3		66.3	65.9	66.7		34.0	56.2	49.5		61.7	61.0	66.4		35.8	53.4	50.2		48.7	49.7	52.5
		<mark>		34.3	61.6	52.9		61.5	67.8	64.4		40.5	64.2	55.9		66.8	68.5	**72.3**		41.8	64.1	52.2		57.4	55.0	60.2
		<emphasize>		38.3	**69.0**	64.2		62.7	63.6	62.9		37.1	64.7	55.9		65.0	68.1	69.5		38.4	**64.8**	57.7		57.0	52.0	59.5
GPT-2 Large	B		34.0				44.5				27.1				42.3				27.7				26.9			
	AS			**63.2**	54.8	-		54.7	45.1	-		54.5	45.2	-		46.0	43.7	-		**58.4**	44.9	-		32.8	33.6	-
	MP	⋆		22.1	44.9	30.5		43.4	42.2	41.2		23.7	30.1	39.2		39.9	43.8	44.0		22.6	30.0	38.0		25.2	27.8	27.7
		"		29.7	41.2	41.8		40.0	40.9	44.0		27.3	31.3	36.8		42.5	47.3	49.1		27.9	30.4	32.2		27.7	32.3	30.0
		<mark>		35.4	46.1	34.1		35.6	45.8	25.1		25.6	**56.5**	51.1		36.1	48.4	42.0		26.0	57.5	50.5		22.5	31.6	27.4
		<emphasize>		34.8	46.7	45.4		38.2	45.8	30.3		26.3	52.2	55.1		40.8	47.7	44.3		25.4	51.1	55.6		25.4	30.6	27.0
GPT-2 XL	B		28.0				51.2				20.5				50.1				24.8				31.8			
	AS			34.0	39.9	-		**55.9**	45.7	-		35.5	25.6	-		52.5	52.4	-		33.9	34.9	-		36.3	34.6	-
	MP	⋆		28.9	31.0	41.7		48.7	48.1	49.3		21.1	25.7	32.1		49.5	51.2	50.2		23.2	27.2	28.1		31.8	34.0	33.8
		"		30.2	35.8	43.7		50.0	46.0	46.1		23.5	29.9	37.5		49.8	51.8	51.9		25.6	28.2	30.7		32.2	33.6	33.6
		<mark>		30.1	43.3	51.0		49.8	49.5	47.0		17.4	38.3	36.2		47.3	**53.4**	49.9		19.0	**38.1**	34.8		29.8	34.8	31.6
		<emphasize>		28.4	42.3	42.9		48.2	50.4	46.1		18.2	32.3	37.9		48.7	**53.4**	50.4		20.8	32.1	34.2		30.0	35.2	32.4

Table 1: Question vs. Context Table: B=Baseline (no emphasis); AS=Attention steering; MP=Marked prompting; C=Context; Q=Question; <q>=question string; <c>=context string; The highest score for each model is in bold, the second highest on the other prompt structure is underlined. The AS method requires a substring within the input string to be emphasized, and hence, it is undefined for the Q+C setting, as in that setting the substring will be the entire input string.

Performance Improvement Across Almost All Settings We find that across all datasets, models, and prompt structures, there is a performance difference between emphasizing either the context, the question, or both, which will further be discussed in Section 6.2. However, emphasizing parts of the input is overall beneficial and can strongly improve models' NE performance. For example, on the Natural Questions dataset, every emphasis method improves Llama-2 NE performance for the question-first setting (except for emphasizing the context using MP-*). To more concretely assess the overall performance improvement emphasizing the input entails, we compare the averaged NE performance across all models, dataset, and settings, to the averaged performance over all emphasis methods, models, datasets, and settings. We find that, while the average NE performance is 43.17%, the average model performance when emphasizing the input is 47.31%.

6 Analysis and Discussion

6.1 Sequence Order Analysis

No-emphasis Perplexity To further understand the behavior we find from our analysis of RQ1 in Section 5.1, we evaluate the average perplexity of the prompts under each model for each of the two prompt structures – *Question: <q> Context: <c>* and *Context: <c> Question: <q>* –, each dataset and the NE setting. Our results can be seen in Table 2.

Model	NQ		SQuAD		AdversarialQA	
	Question First	Context First	Question First	Context First	Question First	Context First
Llama	**15.08**	15.53	11.49	**10.58**	12.89	**11.96**
GPT-J	20.16	**18.61**	13.13	**13.07**	14.52	**14.36**
GPT-2 Large	36.26	**32.22**	**20.86**	21.24	**22.88**	23.29
GPT-2 XL	30.44	**28.47**	19.02	**18.89**	20.99	**20.70**

Table 2: Model's average perplexity on each dataset, for each prompt structure, in the zero shot (no emphasis) setting. Lower is better. NQ=Natural Questions.

Across almost all dataset, models' perplexity is lower (i.e., "better") for the context-first setting, with an average reduction of 1.77 on the Natural Questions (25.48 vs. 23.70), 1.77 on SQuAD (16.12 vs. 15.94), 0.24 on AdversarialQA (17.82 vs. 17.57), and over all datasets of 0.73 (19.81 vs. 19.07). For example, the highest perplexity reduction occurs for GPT-2 large, which scores 32.22 on the Natural Questions dataset when the context is provided first, in comparison to 36.26 for the question-first setting (4.04 difference).

Perplexity vs. Accuracy Surprisingly, looking at Table 2 for the two cases above in which placing the context first does not improve accuracy (GPT-2 large and GPT-J on AdversarialQA), we find that only GPT-2 large scores higher on perplexity for the context-first setting, which could potentially explain the accuracy difference as the model finds this prompt structure more confusing on this particular dataset. However, we do not find that the perplexity was higher for the questions-first structure for GPT-J. Moreover, we find two more cases where models' perplexity was higher for one of the structures, but accuracy was higher on the same structure: Llama-2 on Natural Questions and GPT-2 large on SQuAD. This suggests that while the models do not find the context-first structure more confusing (as measured by their perplexity), they score lower on accuracy for another reason.

6.2 Emphasis Analysis

Different Emphasis Methods Affect Similar Models Differently We find that different emphasis methods affect similar models differently. On the Natural Questions dataset, while emphasizing the context using the MP-<emphasize> method on GPT-J on the question-first structure increases its NE accuracy from 33.3% to 69.0%, outperforming all other models, using the MP-* method reduces its score to 26.9%.

Similar Emphasis Methods Affect Different Models Differently We also find that similar emphasis methods affect different models differently. For example, on the AdversarialQA dataset and the context-first, context-emphasis setting, AS improves Llama-2 NE performance from 49.4% to 53.3%, and GPT-2-XL's NE performance from 26.9% to 33.6%. However, AS reduces GPT-J's performance from 46.2% to 41.7%.

Best Emphasis Methods To assess which emphasis methods are best for each model, we average the scores across all datasets and settings for each model. We find that the top 3 best emphasis methods for each model are (in decreasing order): Llama-2: (MP-", MP-<mark>, MP-<emphasize>), GPT-J: (MP-<emphasize>, MP-<mark>, MP-"), GPT-2 large: (AS, MP-<emphasize>, MP-<mark>), and GPT-2-XL: (AS, MP-<mark>, MP-<emphasize>).

Overall, across all models, datasets and settings, the best emphasis method may seem to be AS, with an average accuracy of 49.39%. This is aligned with Zhang et al. (2023)'s result, which finds that AS outperforms two MP methods on the task of instruction following.

However, looking at the top accuracies for each model on each dataset, we actually find that AS only outperforms other emphasis methods 6 out of the 24 times (4 models, 2 prompt structures for each, on 3 datasets). And from that regard, MP outperforms it (MP also scores fairly close to it overall, with the highest average accuracy of 48.68% for MP-<emphasize>).

Emphasis on C vs. Q vs. CQ To analyze which substring is better to emphasize – the context, the question, or both –, we average the performance of all models across all datasets, emphasis methods, and prompt structures. We find that the highest performance is achieved by emphasizing both context and question, with an average accuracy score of 49.49%. However, we also find that emphasizing the context is roughly just as good, with an average accuracy score of 49.21%, and that emphasizing the question falls much below both, with an average accuracy score of 43.68%.

Does Size Matter? Here, we analyze whether models' size affects their ability to be emphasized by looking at the best method for each on each setting. And while we do not find a clear pattern, we find some cases that suggest that emphasis methods are more beneficial for smaller models. For example, on the SQuAD dataset and the question-first setting, GPT-2 large improves from 27.1% to 56.5% using the MP-<mark> method (29.4% improvement), where GPT-J improves from 45.5% to 64.7% using the MP-<emphasis> method (19.2% improvement), and Llama-2 from 60.4% to 72.3% using the MP-" method (11.9% improvement).

Does Training Data Matter? To evaluate the effect training data has on the susceptibility of models for being emphasized, we compare GPT-2 large and GPT-2-XL as they are trained on the same corpus. From Table 1 we can see that, while these two models are trained on similar data, on many occasions, similar emphasis methods result in different behavior. For example, on the question-first setting and the Natural Questions dataset, while AS result in the highest performance when applied to the question on both models, for context emphasis, the best method for GPT-2 large is AS, where for GPT-2-XL the best method is MP-<mark> or MP-<emphasize>. We also do not find the same absolute improvements across the two models when looking at similar emphasis methods and similar settings. This suggests that, while the training data

Model	**Emphasis Method**	**Question Emphasis**		**Context Emphasis**	
		Accuracy	Question String Avg. Attention Score	Accuracy	Context String Avg. Attention Score
GPT 2 Large	*	22.1	0.0078	44.9	0.0041
	"	29.7	0.0078	41.2	0.0094
	mark	35.4	0.0074	46.1	0.0088
	emphasis	34.8	0.0070	46.7	0.0084
GPT 2 XL	*	28.9	0.0076	31.0	0.0039
	"	30.2	0.0075	35.8	0.0095
	mark	30.1	0.0071	43.3	0.0089
	emphasis	28.4	0.0067	42.3	0.0085

Table 3: Attention scores analysis across different models' layers and heads for different emphasis methods.

has some effect on which emphasis method is beneficial for each model, it is not the whole story.

Attention Heads Analysis To further understand why different emphasis methods result in different models' scores we evaluate the attention scores for the strings that are being emphasized by the different methods on the question-first setting. More concretely, for each MP method, we send each sentence from the Natural Questions dataset to the model. We then average the attention scores across all model's heads and layers for the tokens corresponding to the string to be emphasized – either the context or the question. Our results can be seen in Table 3.

We do not find a clear pattern that highlights whether emphasis methods result in a higher or lower attention scores for emphasis strings. For example, while GPT 2 large has an increase of accuracy from 22.1% to 29.7% when changing from the MP-* method to the MP-" method on the question-emphasis setting, the attention scores stay the same. We also see that sometimes the attention scores go up when accuracy go down, such as in GPT 2 XL, MP-mark to MP-* on question emphasis, and sometimes the attention scores go down when accuracy go up, such as in GPT 2 large, MP-" to MP-emphasis, on the context emphasis setting.

6.3 Known Vs. Unknown Knowledge

Marked Prompting We next evaluate whether MP, and specifically the best performing setting overall – context-first, question + context emphasis –, works better for addressing knowledge that models have or do not have. Our results can be seen in Table 4.

We can see that, across almost all three datasets and all models, emphasizing the input string on the unknown knowledge split results in more improvement than emphasizing the input string on the known knowledge split. For example, on Natural Questions, for unknown knowledge, Llama-2 and GPT-J improve from 46.4% and 63.2% to 49.9% and 65.5%, respectively. Where on the known knowledge split, they respectively change from 93.4% to 93.6% and from 88.5% to 85.2%.

One potential explanation for that is that models tend to already perform reasonably well on known knowledge, since they have most likely acquired that knowledge during training. However, emphasizing input strings on unknown knowledge forces the model to adapt its learned representations to handle unseen or less familiar data.

Attention Steering Next, we evaluate whether AS, and specifically the best performing setting of AS – question-first, question steering –, works better for addressing knowledge that models have or do not have. Our results can be seen in Table 5.

Across almost all three datasets and all models, steering the input string in the unknown knowledge split results in more improvement than steering it in the known knowledge split. For example, on Natural Questions, for unknown knowledge, GPT-J and GPT-2 Large improve from 27.9% and 29.4% to 59.9% and 54.6%, respectively. In contrast, on the known knowledge split, they improve from 56.4% to 76.8% and from 52.1% to 71.4%, respectively.

6.4 Can Emphasis Be Bad?

While we find that emphasizing parts of the input using various emphasis methods can be beneficial, it does require experimentation, as choosing the wrong emphasis method can actually be disadvantageous. Averaging over all datasets, models, and settings in Table 1, we find that the worse emphasis method is MP-*, only increasing the average accuracy from 43.17% to 43.66%, and at its worst setting it reduces Llama-2's baseline performance from 46.3% to 31.6% on the Natural Questions dataset in the question-first setting.

6.5 Newer Models, Instruction Tuning, and Max Context Length

In addition to our main results, we also add an analysis of five more LLMs, all of which were published in 2023 or afterwards and contain between 7B and 13B parameters. Two of the five additional LLMs were instruction-tuned, to evaluate whether such tuning affect the performance change due to different emphasis methods. Lastly, all five of the additional models were evaluated using their maximum context size (up to 4k). Our results can be

Model	Natural Questions					SQuAD					AdversarialQA				
	Knowledge Amount	Known No Emphasis	Known Emphasis	Unknown No Emphasis	Unknown Emphasis	Knowledge Amount	Known No Emphasis	Known Emphasis	Unknown No Emphasis	Unknown Emphasis	Knowledge Amount	Known No Emphasis	Known Emphasis	Unknown No Emphasis	Unknown Emphasis
Llama 2	20.0	93.4	93.6	**46.4**	**49.9**	18.1	88.6	91.9	**70.0**	**79.7**	20.5	77.9	71.7	**42.7**	**51.9**
GPT J	4.3	90.2	89.5	**63.2**	**65.5**	9.2	83.5	86.5	**58.7**	**71.2**	14.2	71.3	73.7	**42.0**	**58.0**
GPT 2 Large	1.7	**78.8**	**86.4**	43.7	43.1	4.6	79.6	84.1	**40.5**	**47.6**	11.4	64.9	61.9	**22.0**	**25.9**
GPT 2 XL	2.2	88.5	85.2	**50.2**	**48.4**	6.0	78.5	79.1	**48.2**	**50.3**	11.9	**67.5**	**69.8**	26.9	28.9

Table 4: Known vs. Unknown Table: **Marked Prompting**. We find that the best emphasizing method is marked prompting, and in particular, concatenating the string "<emphasize>" before and after the context and question strings. We use the closed-book setting to evaluate models' parametric knowledge, and compare the ZS baseline (no emphasis) to the best marked prompting approach. In bold, the largest improvement for each model on each dataset. Knowledge Amount is measured using accuracy, as the average number of questions models can successfully answer correctly without context (cf. Section 4.5).

Model / Dataset	Natural Questions				SQuAD				AdversarialQA			
	Known No Emphasis	Known Steering	Unknown No Emphasis	Unknown Steering	Known No Emphasis	Known Steering	Unknown No Emphasis	Unknown Steering	Known No Emphasis	Known Steering	Unknown No Emphasis	Unknown Steering
Llama-2	**69.1**	**81.0**	30.8	37.0	80.6	85.8	**56.7**	**62.0**	69.8	67.8	**38.0**	**38.5**
GPT-J	56.4	76.8	**27.9**	**59.9**	53.4	66.5	**44.7**	**62.6**	**63.8**	**63.8**	44.4	41.9
GPT-2 Large	52.1	71.4	**29.4**	**54.6**	47.1	68.9	**26.1**	**53.7**	42.1	59.6	**25.8**	**58.3**
GPT-2 XL	51.4	49.1	**24.2**	**33.5**	39.1	52.4	**19.3**	**34.4**	**40.7**	**50.8**	20.9	29.6

Table 5: Known vs. Unknown Table: **Attention Steering**. While attention steering does not overall perform as well as marked prompting, we also evaluate models' parametric knowledge (known vs. unknown) using the closed-book setting, and compare the ZS No Emphasis (no emphasis) to the attention steering approach where the question is presented first in the prompt and is being emphasized – as that is the best setting we find for attention steering. In bold, the largest improvement for each model on each dataset.

seen in Table 6.

Notably, 1) Our results still hold: A) the ordering of inputs plays a crucial role in all models' performances, where putting the context first strongly improves performance; B) emphasis methods also improve models' performances. 2) The context size does not play a role in the results, in the sense that our initial results and conclusions still hold. 3) Instruction-tuned models are also susceptible to input order and emphasis methods.

Model	Emphasis Method	Natural Qustions					
		Question First			Context First		
		No Emphasis	Q	C	No Emphasis	Q	C
Falcon-7B		17.0			40.2		
	*		10.2	12.8		25.0	38.6
	"		17.0	36.8		38.0	**42.4**
	mark		11.0	34.0		34.4	41.4
	emphasis		9.8	30.8		36.0	40.2
Falcon-7B Instruct		24.4			39.8		
	*		24.6	17.4		20.8	40.6
	"		29.0	34.2		16.6	36.2
	mark		25.0	**47.2**		16.0	39.8
	emphasis		16.2	42.6		12.6	38.6
MPT-7B		17.0			43.5		
	*		20.5	18.5		49.0	**53.5**
	"		16.0	42.0		36.0	37.5
	mark		34.5	49.5		29.0	46.0
	emphasis		17.5	37.5		38.0	52.0
MPT-7B Instruct		25.0			13.0		
	*		26.7	20.2		32.0	14.0
	"		15.7	29.2		8.25	12.5
	mark		15.0	26.2		15.0	13.0
	emphasis		20.5	**40.7**		20.5	13.0
Llama-13B		28.4			58.6		
	*		27.4	27.2		41.2	55.8
	"		30.4	55.0		52.0	57.0
	mark		23.4	36.8		41.6	60.0
	emphasis		26.4	53.4		49.4	**60.8**

Table 6: Analysis of newer models, two of which are instruction-tuned, where all models are evaluated using their maximum context length (up to 4k).

7 Conclusion

Focusing on reading comprehension, we evaluate 1) how the order of the question and context affects model performance; and 2) whether emphasizing either the question, the context, or both enhances performance. Experimenting with 9 LLMs across multiple datasets, we find that presenting the context before the question improves model performance, with an accuracy increase of up to 31%. Furthermore, emphasizing the context yields superior results compared to emphasizing the question, and in general, emphasizing parts of the input is particularly effective for addressing questions that models lack the parametric knowledge to answer.

Limitations

While we try to be comprehensive in our comparisons, we only evaluate one approach to represent the question – "Question: <q>", and context: "Context: <c>". However, as discussed in the Section 2, many other approaches exist. That being said, our goal is not to find the best method, but to highlight the issue that exists in the first place, which is the lack of standardization. Additionally, while we focus on reading comprehension, it is an open question if the emphasis methods and ordering also affect other domains or much larger LLMs (e.g., 70B+ parameters).

Ethics Statement

The motivation for this paper is to highlight the issue that exists in the lack of standardization of input presentation in reading comprehension, and to show that emphasizing parts of the inputs can be beneficial. We believe that it is crucial that future work continues to evaluate and improve models' performance using different settings so they can be safely used in practical scenarios.

Acknowledgments

We thank the reviewers for their comments and great suggestions. The authors acknowledge financial support from NIH grants OT2TR003422 and R01LM013400.

References

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Alshaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics.

Jinheon Baek, Alham Aji, and Amir Saffari. 2023. Knowledge-augmented language model prompting for zero-shot knowledge graph question answering. In *Proceedings of the First Workshop on Matching From Unstructured and Structured Data (MATCHING 2023)*, pages 70–98, Toronto, ON, Canada. Association for Computational Linguistics.

Hossein Bahak, Farzaneh Taheri, Zahra Zojaji, and Arefeh Kazemi. 2023. Evaluating chatgpt as a question answering system: A comprehensive analysis and comparison with existing models.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:662–678.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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*, volume 33, pages 1877–1901. Curran Associates, Inc.

Daixuan Cheng, Shaohan Huang, and Furu Wei. 2023. Adapting large language models via reading comprehension.

Eunsol Choi, Daniel Hewlett, Jakob Uszkoreit, Illia Polosukhin, Alexandre Lacoste, and Jonathan Berant. 2017. Coarse-to-fine question answering for long documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, Vancouver, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Christopher Clark and Matt Gardner. 2018. Simple and effective multi-paragraph reading comprehension. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 845–855, Melbourne, Australia. Association for Computational Linguistics.

Maxime De Bruyn, Ehsan Lotfi, Jeska Buhmann, and Walter Daelemans. 2022. Is it smaller than a tennis ball? language models play the game of twenty questions. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 80–90, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1832–1846, Vancouver, Canada. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning.

Daria Dzendzik, Jennifer Foster, and Carl Vogel. 2021. English machine reading comprehension datasets: A survey. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8784–8804, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. 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 3816–3830, Online. Association for Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ananth Gottumukkala, Dheeru Dua, Sameer Singh, and Matt Gardner. 2020. Dynamic sampling strategies for multi-task reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 920–924, Online. Association for Computational Linguistics.

Hao Huang, Xiubo Geng, Guodong Long, and Daxin Jiang. 2022. Understand before answer: Improve temporal reading comprehension via precise question understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 375–384, Seattle, United States. Association for Computational Linguistics.

Sathish Reddy Indurthi, Seunghak Yu, Seohyun Back, and Heriberto Cuayáhuitl. 2018. Cut to the chase: A context zoom-in network for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 570–575, Brussels, Belgium. Association for Computational Linguistics.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606, Toronto, Canada. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large language models struggle to learn long-tail knowledge.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2023. Analyzing feed-forward blocks in transformers through the lens of attention map.

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.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022. Large language models with controllable working memory.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023a. Evaluating the logical reasoning ability of chatgpt and gpt-4.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023b. Lost in the middle: How language models use long contexts.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023. Locating and editing factual associations in gpt.

MPT. 2023. Introducing mpt-7b: A new standard for open-source, commercially usable llms.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2335–2345, Florence, Italy. Association for Computational Linguistics.

Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. 2023. Capabilities of gpt-4 on medical challenge problems.

OpenAI. 2023a. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023b. Gpt-4 technical report.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Priyanka Sen and Amir Saffari. 2020. What do models learn from question answering datasets? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2429–2438, Online. Association for Computational Linguistics.

Sagi Shaier, Kevin Bennett, Lawrence Hunter, and Katharina Kann. 2023. Emerging challenges in personalized medicine: Assessing demographic effects on biomedical question answering systems. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 540–550, Nusa Dua, Bali. Association for Computational Linguistics.

Sagi Shaier, Lawrence E Hunter, and Katharina von der Wense. 2024. Desiderata for the context use of question answering systems.

Chenglei Si, Shuohang Wang, Min-Yen Kan, and Jing Jiang. 2019. What does bert learn from multiple-choice reading comprehension datasets?

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge.

Hendrik Strobelt, Albert Webson, Victor Sanh, Benjamin Hoover, Johanna Beyer, Hanspeter Pfister, and Alexander M. Rush. 2022. Interactive and visual prompt engineering for ad-hoc task adaptation with large language models.

Saku Sugawara, Yusuke Kido, Hikaru Yokono, and Akiko Aizawa. 2017. Evaluation metrics for machine reading comprehension: Prerequisite skills and readability. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 806–817, Vancouver, Canada. Association for Computational Linguistics.

Weiwei Sun, Hengyi Cai, Hongshen Chen, Pengjie Ren, Zhumin Chen, Maarten de Rijke, and Zhaochun Ren. 2023. Answering ambiguous questions via iterative prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7669–7683, Toronto, Canada. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Multi-granular sequence encoding via dilated compositional units for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2141–2151, Brussels, Belgium. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Wenya Wang and Sinno Pan. 2022. Deep inductive logic reasoning for multi-hop reading comprehension. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4999–5009, Dublin, Ireland. Association for Computational Linguistics.

Albert Webson and Ellie Pavlick. 2022. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, Seattle, United States. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models.

Liang Wen, Houfeng Wang, Yingwei Luo, and Xiaolin Wang. 2022. M3: A multi-view fusion and multi-decoding network for multi-document reading comprehension. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1450–1461, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. 2023. Adaptive chameleon or stubborn sloth: Unraveling the behavior of large language models in knowledge clashes.

Pengtao Xie and Eric Xing. 2017. A constituent-centric neural architecture for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1405–1414, Vancouver, Canada. Association for Computational Linguistics.

Yichong Xu, Xiaodong Liu, Yelong Shen, Jingjing Liu, and Jianfeng Gao. 2019. Multi-task learning with sample re-weighting for machine reading comprehension. 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 2644–2655, Minneapolis, Minnesota. Association for Computational Linguistics.

An Yang, Kai Liu, Jing Liu, Yajuan Lyu, and Sujian Li. 2018. Adaptations of ROUGE and BLEU to better evaluate machine reading comprehension task. In *Proceedings of the Workshop on Machine Reading for Question Answering*, pages 98–104, Melbourne, Australia. Association for Computational Linguistics.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2346–2357, Florence, Italy. Association for Computational Linguistics.

Changchang Zeng, Shaobo Li, Qin Li, Jie Hu, and Jianjun Hu. 2020. A survey on machine reading comprehension: Tasks, evaluation metrics and benchmark datasets.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. 2023. Tell your model where to attend: Post-hoc attention steering for llms.

Xuanyu Zhang. 2019. MCˆ2: Multi-perspective convolutional cube for conversational machine reading comprehension. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 6185–6190, Florence, Italy. Association for Computational Linguistics.

Wanjun Zhong, Yifan Gao, Ning Ding, Yujia Qin, Zhiyuan Liu, Ming Zhou, Jiahai Wang, Jian Yin, and Nan Duan. 2022. ProQA: Structural prompt-based pre-training for unified question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4230–4243, Seattle, United States. Association for Computational Linguistics.