Benchmarking Query-Conditioned Natural Language Inference

Marc E. Canby[1,2*] **Xinchi Chen**[2†] **Xing Niu**[2] **Jifan Chen**[2] **Bonan Min**[2]
Sergul Aydore[2] **Vittorio Castelli**[2]
[1]University of Illinois Urbana-Champaign, [2]AWS AI Labs
marcec2@illinois.edu
{xcc, xingniu, chenjf, bonanmin, saydore, vittorca}@amazon.com

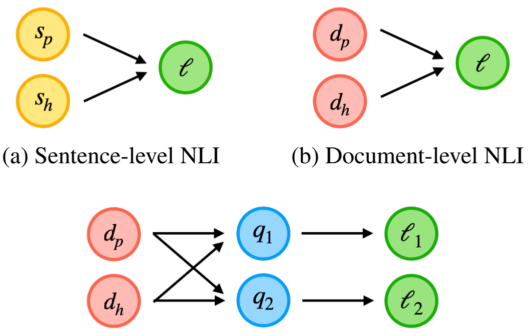
Abstract

The growing excitement around the ability of large language models (LLMs) to tackle various tasks has been tempered by their propensity for generating unsubstantiated information (hallucination) and by their inability to effectively handle inconsistent inputs. To detect such issues, we propose the novel task of *Query-Conditioned Natural Language Inference* (QC-NLI), where the goal is to determine the semantic relationship (e.g. `entailment` or `not entailment`) between two documents *conditioned on a query*; we demonstrate that many common tasks regarding inconsistency detection can be formulated as QC-NLI problems. We focus on three applications in particular: fact verification, intrinsic hallucination detection, and document inconsistency detection. We convert existing datasets for these tasks into the QC-NLI format, and manual annotation confirms their high quality. Finally, we employ zero- and few-shot prompting methods to solve the QC-NLI prediction problem for each task, showing the critical importance of conditioning on the query.

1 Introduction

Natural language inference (NLI) has become a standard method to detect inconsistencies across pairs of sentences or documents (Bowman et al., 2015; Williams et al., 2018; Yin et al., 2021; Sadat and Caragea, 2024). The need to detect inconsistencies has become even more pronounced in the era of large language models (LLM's) – much work has shown that such models produce internally contradictory text, information inconsistent with external knowledge, or information that contradicts input evidence, as in a retrieval-augmented generation (RAG) system (Zhang et al., 2023a,b).

[*] Work done during an internship at AWS AI Labs.
[†] Corresponding author.
https://github.com/amazon-science/Query-Conditioned-NLI



(a) Sentence-level NLI (b) Document-level NLI

(c) Query-conditioned NLI (**ours**)

Figure 1: **Natural language inference (NLI).** (a) Sentence-level NLI has a label ℓ indicating the semantic relationship between a premise sentence s_p and hypothesis sentence s_h. (b) Document-level NLI conditions ℓ on a premise *document* d_p and a hypothesis document d_h. (c) Query-conditioned NLI conditions label ℓ_i on premise document d_p, hypothesis document d_h, and a *query* q_i, which indicates the aspect of the documents the semantic relationship should be based on.

To detect these types of shortcomings, we propose the task of *query-conditioned NLI* and show how various inconsistency detection tasks can be solved using it. Query-conditioned NLI aims to determine the relationship between the premise and hypothesis *given a query*. The inclusion of the query distinguishes our task from standard (non-conditional) NLI tasks, which only consider the relationship of the entire premise to the entire hypothesis; these paradigms are illustrated in Figure 1. Consider the following example from traditional NLI:

> *Premise:* Jack is working in the park.
> *Hypothesis:* Jack is sleeping in the park.
> *Label:* Contradiction

A typical NLI dataset would consider these sentences a contradiction because Jack can't be sleeping if he is working. However, in conjunction with the query "Where is Jack?", the label should

Findings of the Association for Computational Linguistics: ACL 2025, pages 14808–14835
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

be `entailment` because both sentences state that Jack is in the park. Hence, the query defines the aspect of the input sentences for which the NLI label should apply. This problem is even more pronounced when the premise and hypothesis are longer documents, which often contain both consistent and inconsistent information.

In this paper, we introduce the task of query-conditional NLI and provide a benchmark dataset for it. In particular, we show that the tasks of **fact verification**, **intrinsic hallucination detection**, and **inconsistent document detection** can be cast as a QC-NLI problem, and we adapt existing datasets into our framework (examples may be seen in Table 1). Finally, we propose several prompt-based strategies to solve the QC-NLI task, showing that few-shot prompting is most effective. We also do an ablation study where the model predicts the relationship between the documents *without the query*; crucially, the drop in performance suggests that conditioning on the query is necessary to make the prediction. Finally, we note that no method exceeds 83% on any dataset, leaving room for future work to improve on the benchmark.

2 Related Work

Traditional natural language inference Our work builds on the task of natural language inference (Condoravdi et al., 2003), which is traditionally done at the sentence-level: the premise and hypothesis are each single sentences, and the task is to determine whether the premise entails, contradicts, or is neutral to the hypothesis. Typical datasets for this task are the Recognizing Textual Entailment challenge tasks (Condoravdi et al., 2003) and the Stanford Natural Language Inference dataset (Bowman et al., 2015). Our work differs from these works in two major ways: first, rather than individual sentences, we focus on paragraphs or entire documents. Second, our proposed task is *query-conditional* NLI: we are not concerned with the relationship of the entire premise to the entire hypothesis, but rather a particular aspect of them, determined by the query.

Document-level natural language inference Recent work has extended the sentence-based NLI task to documents. The first to do this is DocNLI (Yin et al., 2021), which adapted several existing datasets such that the premise and hypothesis can each consist of several sentences. However, the task still requires the entire premise to entail or not entail the entire hypothesis; therefore, any small contradiction can lead to a `not_entailment` label despite most of the text being entailed. Other document-level NLI tasks and methods have recently been proposed, such as ContractNLI (Koreeda and Manning, 2021) (and its corresponding method SpanNLI) and the method DocInfer (Mathur et al., 2022). These works attempt to identify evidence within a document (the premise) that entails or contradicts a hypothesis: in that sense, the hypothesis itself can be seen as defining the aspects of the premise that are relevant. These works differ from our QC-NLI formulation because (1) they treat the entire hypothesis as a single aspect, whereas we allow the query to identify which aspect of the hypothesis document is relevant; and (2) the primary goal of these works is source attribution within the longer document. While the latter is a potential future direction for QC-NLI, our task is mostly focused on the query-conditional aspect NLI.

Conditional natural language inference Relatively little work has looked at natural language inference conditioned on certain aspects of the premise or hypothesis. Goyal and Durrett (2020) investigate fine-grained NLI by predicting entailment decisions on dependency arcs within a sentence. Honovich et al. (2021) predicts NLI labels for sentence spans based on ground-truth knowledge, using a mixture of question-answering and NLI. CondNLI (Kim et al., 2023) also makes predictions at the sub-sentence level: this paper introduces the BioClaim dataset, which has token-level annotations indicating tokens of the premise and hypothesis that are contradictory aspects and tokens that represent their conditions. The task is to determine whether a span from the hypothesis is contradictory or neutral to the premise. However, the spans seem to be arbitrary sequences from the hypothesis and the identification of "conditions" is not clear. Our work improves upon this idea in a few ways: (1) the query indicates at a high level the aspects of the premise and hypothesis that are relevant, even if they do not exist as a single span; (2) the user indicates the "condition" via the specification of the query, rather than having a model identify individual tokens as the condition.

Inconsistency detection and fact verification The recent explosion of interest in large language models (LLM's) has led to several works on hallucination detection and other forms of inconsistency detection; a survey on hallucination detection is available by Zhang et al. (2023b). AlignScore (Zha et al., 2023), for example, develops a function to evaluate factual consistency between two texts, which is similar to our goal. Several other methods have been proposed recently, such as FactScore (Min et al., 2023), which proposes a method that breaks LLM outputs into atomic facts that can be externally verified – this is similar to conditional NLI because the individual facts can be seen as answers to an invisible query; our work builds on this by making the query a central part of the formulation. Steen et al. (2023) aim to improve robustness of NLI models in the dialgogue faithfulness setting by performing data augmentation and Monte Carlo dropout. Finally, Mündler et al. (2024) prompts an LLM to see if another LLM's output contradicts itself. These methods are highly relevant to our work, as many could be used as methods on the dataset we propose; however, none of them explicitly focus on the conditional NLI aspect, which is the main thrust of our paper.

3 Query-conditioned Natural Language Inference

3.1 Task Formulation

Query-conditioned NLI aims to identify the relationship between two documents with respect to a query. The possible labels depend on the task of interest, but a common label set is {`entailment`, `not_entailment`}. We point out a few critical aspects of QC-NLI that distinguish it from previous work.

Query-based A major distinction between QC-NLI and other NLI tasks is that the label is conditioned on the query. This effectively enables *aspect-based* NLI, as the query indicates what aspect the NLI label should be conditioned on. This is distinct from previous work, where the label is based on the entire premise and hypothesis.

Document-based We focus on tasks where the premise and/or hypothesis are multi-sentence documents. Previous document-based NLI formulations would use a `contradiction` label when even one part of the hypothesis is inconsistent with the premise; however, our query-condional formulation enables many NLI labels for the same pair of documents.

General formulation Most importantly, our formulation is general enough that several distinct tasks can be cast as a QC-NLI prediction problem, as shown in Section 3.2. Also, a system that solves QC-NLI could be used in downstream tasks: for example, in a retrieval-augmented generation (RAG) setting, retrieved documents could be reranked to penalize or reward (as in multi-view generation (Chen et al., 2024)) contradictory perspectives on the query.

3.2 Applications

We target three primary applications that can be solved by casting them as QC-NLI tasks: inconsistent document detection (e.g., among RAG input documents), intrinsic hallucination detection, and fact verification. An example of each application in QC-NLI format is given in Table 1; in Section 4, we show how datasets for these applications can be converted into this format.

Inconsistent document detection Given a query and a set of documents (e.g. a set of retrieved documents), the task is to identify the relationship between each pair of documents with respect to the query. Since the goal of this task is to detect inconsistencies, the labels are `contradiction` and `not_contradiction`.

Intrinsic hallucination detection When presented with an evidence document and a query, the task is to identify whether an LLM generates a response that is entailed by the evidence with respect to the query. This is particularly important for RAG applications where the goal is to produce an answer to the query that is entailed by the evidence.

Fact verification The task is to identify whether an authoritative source (e.g. Wikipedia article) entail or not entail a dubious source (e.g. output of LLM or untrustworthy news article) with respect to a particular aspect (query). For our purposes, the query is considered part of the task specification; though, in practice, one could use an LLM to generate questions about facts in the documents, in a manner similar to Min et al. (2023).

4 Creating and evaluating data for QC-NLI

4.1 Adapting Datasets for QC-NLI

For each task of interest, we prompt GPT-4o (OpenAI et al., 2024a) to convert an existing dataset into an appropriate QC-NLI format. We choose to transform existing datasets rather than start from scratch because they have already passed quality checks

Dataset/Task	Premise d_p	Hypothesis d_h	Query q and Label ℓ
SNLI	A lady dressed in a green and black jacket...stood at the edge of a grassy field. The horse trotted gracefully a few feet away, its mane swaying in the gentle breeze...she positioned her camera to capture the perfect moment...	In a bustling city park, a woman busily snapped photos of a horse that trotted elegantly across the grassy field. Her camera clicked rhythmically as she captured the animal's swift movements and serene grace. Amidst the distant sounds of traffic...	Who is capturing images of the horse? → `entailment`
RobustQA *Inconsistent document detection*	...if I thought a bank was going under, then yes, absolutely I would withdraw my money. Now, we can debate whether...	I probably would not take it out, since I have enough layers of backstops: Maybe if I could find a better rate. :)	would you withdraw your money from your bank if you thought it was going under? → `contradiction`
RagTruth *Hallucination detection*	passage 1:. . . The two sites...are Favotter.net and Favstar.fm. by using these two sites, you should be able to get a good idea of who is favouriting your tweets... passage 2:...Click on your Twitter name. . . Click on the Settings option...Click on the blue Notifications link... you can set up your account so that you're notified via email whenever someone marks one of your tweets as one of their favorites...	You can see who liked your tweets by using tracking websites such as Favotter.net and Favstar.fm. These sites can give you an idea of who is favouriting your tweets. Alternatively, you can also set up notifications on Twitter itself. By going to your account settings and clicking on the notifications link, you can arrange to be notified via email whenever someone likes one of your tweets.	how can you tell who liked your tweet → `entailment`
FactScore *Fact verification*	<s>Lanny Flaherty Lanny Flaherty (born July 27, 1942) is an American actor.</s><s>Career. He has given his most memorable performances in "Lonesome Dove",...	Lanny Flaherty is an American actor born on December 18, 1949, in Pensacola, Florida. He has appeared in numerous films, television shows, and theater productions...	What is Lanny Flaherty's profession? → `entailment` When was Lanny Flaherty born? → `not_entailment`

Table 1: **QC-NLI applications.** Excerpted examples from our QC-NLI dataset for three applications; relevant parts of the documents to determine label based on the query are highlighted in red or blue for emphasis. *Inconsistent document detection:* Do two retrieved evidence documents contradict or not contradict each other with respect to the query? *Hallucination detection:* Does the evidence passage entail or not entail the output of an LLM prompted on the evidence? *Fact verification:* Does a verified Wikipedia biography entail or not entail an LLM biography of the same person?

and been studied in previous works; thus, we can simply augment them with the necessary pieces to make a well-formed QC-NLI dataset without sacrificing quality.

Because the tasks and datasets are very different, we use a specialized approach for each dataset to convert it into QC-NLI format; these methods are described at a high level below, and exact prompts can be found in Appendix D. Throughout the process, we use automatic (prompt-based) quality checks to ensure the effectiveness of each step.

We use RobustQA (Han et al., 2023) for inconsistent document detection, RAGTruth (Niu et al., 2024) for hallucination detection, and FactScore (Min et al., 2023) for fact verification. In addition, we make a QC-NLI dataset based on the Stanford Natural Language Inference (SNLI, Bowman et al. (2015)) corpus for image descriptions; this QC-NLI dataset is included because it is a natural extension of traditional NLI.

Image Descriptions: SNLI (Bowman et al., 2015) Each SNLI example contains a premise (one sentence), a hypothesis (one sentence) and a label in {`entailment`, `neutral`, `contradiction`}. To convert to SNLI format, we prompt GPT-4o to write 2 paragraphs and a query, such that the answer to query on the first paragraph is the premise and the answer to query on the second paragraph is the hypothesis. Because the answers to the query on the documents are the original premise and hypothesis, the query-conditioned NLI label is the same as the original NLI label. Because of the difficulty in adapting the data points with `neutral` label, we remove those data points and instead use {`entailment`, `not_entailment`} as our final label set.

Inconsistent document detection: RobustQA (Han et al., 2023) This dataset contains a query and a collection of evidence passages that provide a "yes" or "no" answer to the query across eight different domains; these may be thought of as the input documents to a RAG system. To make a QC-NLI dataset, we simply pair up evidence documents: if they contain different perspectives to the given query (i.e. one is "yes" and the other is "no"), the label is `contradiction`; otherwise the label is `not_contradiction`. Note that many of the questions elicit open-ended opinions, which makes this an ideal dataset to study the identification of documents with different perspectives.

Hallucination detection: RAGTruth (Niu et al., 2024) RAGTruth contains a query, several evidence documents, and an output of an LLM (Llama,

Dataset	Size	Label Counts	Examples Annotated	Quality	Wilson Score Interval (WSI)	% All Agree
SNLI	4,452	`entailment: 2229` `not_entailment: 2223`	33	75.76%	0.7307 ± 0.1410	85.71% (6/7)
RobustQA	2,578	`contradiction: 1213` `not_contradiction: 1365`	33	93.94%	0.8936 ± 0.0896	71.43% (5/7)
RagTruth	829	`entailment: 695` `not_entailment: 134`	33	81.82%	0.7850 ± 0.1289	71.43% (5/7)
FactScore	13,796	`entailment: 9568` `not_entailment: 4228`	51	84.31%	0.8191 ± 0.0992	85.71% (6/7)

Table 2: **QC-NLI datasets.** Dataset statistics including test set size and label counts. Also shows annotation results, including the overall quality (percentage of correct labels among the examples annotated), the Wilson Score Interval (WSI) for the overall quality, and the percentage of the seven examples common to all four annotators where *all* annotators agree.

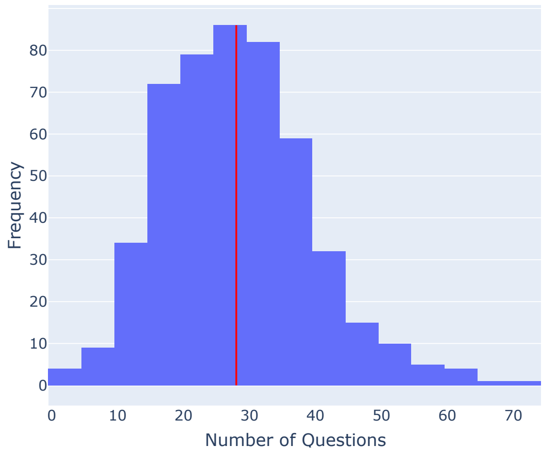


Figure 2: **Number of queries per document pair in FactScore.** The QC-NLI FactScore dataset has many queries per each pair of documents (d_p, d_h), depicted by the histogram. The red line indicates that there are an average of 28 queries per document pair.

Mistral, GPT) when asked to answer the query based on the evidence. Thus, the evidence collectively forms the premise, and the output of the LLM is the hypothesis. RAGTruth also annotates segments of the output as containing baseless or contradictory information with respect to the evidence. To make a QC-NLI dataset, for each document pair we prompt GPT-4o to see if the segments are relevant to the query; if no segment is relevant, then we do not use the example. If any segment is relevant to the query and none of those segments are marked as baseless or contradictory to the evidence, then we assign an `entailment` label; otherwise we assign a `not_entailment` label. The segment annotations are not included in our final dataset; we simply use them to determine the QC-NLI relation between each pair of documents given the corresponding query.

Fact verification: FactScore (Min et al., 2023) An example from FactScore contains a "ground truth" Wikipedia biography, a (potentially suspect) biography written by an LLM (ChatGPT, InstructGPT, or PerplexityAI), and a series of facts in the LLM output annotated by whether or not they are supported in the Wikipedia biography. To convert to QC-NLI format, we simply convert each fact into a question by prompting GPT-4, and use the Wikipedia biography as the premise and the LLM biography as the hypothesis. If the fact is marked as "supported" in Factscore, then we use the `entailment` label; otherwise we use `not_entailment`.

4.2 Data Statistics

Table 2 depicts the size of each test set and label distribution. Across all datasets, our QC-NLI benchmark contains 21,655 examples. The RobustQA dataset is the only dataset that contains multiple document pairs per query, reflecting its use in determining whether documents offer a contradictory perspective to a given query. FactScore, on the other hand, contains multiple queries per document pair, illustrated in Figure 2. For this dataset, there are an average of 28 queries per document pair, so that the query is crucial in determining the NLI label. Additional dataset statistics may be found in Appendix A, and a selection of full examples may be seen in Appendix B.

4.3 Data Evaluation

Because the datasets are converted to QC-NLI format automatically via prompting, we wish to know if the NLI labels are correct for each (d_p, d_h, q)

triplet. Thus, for each dataset, some of the authors manually label several examples, of which 7 examples are common to all annotators. To reduce bias, annotators do not see the true label. Each annotator is onboarded with 5 practice examples, where they are asked to label an example and then compare it with a reference label and explanation. Exact annotator instructions are in Appendix C.

The annotator labels are compared with the dataset labels, and an overall quality is estimated. Additionally, we calculate a 95% Wilson Score Interval (WSI) (Wilson, 1927) for this quantity; we choose this method because it works well for small annotation sizes. Finally, because there are a small number of annotators and only 7 common examples to each of them, we report the percentage of the common examples on which all the annotators agree on the label.

The results of this evaluation are shown in Table 2. All datasets have an overall quality above 75%, with RobustQA having near 94% data quality. Additionally, for all datasets, there is high inter-annotator agreement, indicating fairly consistent annotation standards.

5 Solving the QC-NLI Benchmark

In addition to developing QC-NLI datasets for QC-NLI for various downstream tasks, we present some *prediction* baselines: given two documents and a query, how do the documents relate with respect to the query?

We use three prompting methods: zero-shot prompting, few-shot prompting, and QA+NLI prompting. In all cases, we ask for the LLM to explain its reasoning before making a final prediction.

Zero-shot prompting The LLM is given a description of the task and an explanation of the labels. It is then asked to predict the NLI label from the documents and the query.

Few-shot prompting This is the same as zero-shot, except that a few examples of the task (manually selected from a separate train partition) are provided. At least one example is given for each label (typically 2-3 examples total).[1] Each case also contains brief reasoning to explain the correct answer. For FactScore, where the dataset contains multiple queries for the same pair of passages, our few shot examples use a single pair of passages with a different query for each label; with this format, we hope to illustrate the importance of using the query (and not just the passages) in forming a prediction.

QA + NLI prompting Here, we first ask an LLM to answer the query on each document independently. Then, *without access to the documents or the query*, we ask the LLM to predict the NLI label based only on the two answers. We note that this approach may fail in cases where there are multiple ways to answer the question; consider the (sentence-level) example below:

> *Premise:* The only girl in the class won the track meet and the spelling bee.
> *Hypothesis:* Charlotte, the only girl in the class, won the track meet.
> *Query:* Who won the track meet?

Possible answers based only on the premise (*P1* and *P2*) or only on the hypothesis (*H1* and *H2*) are

> *P1:* The only girl in the class.
> *P2:* The girl who won the spelling bee.
> *H1:* Charlotte.
> *H2:* The only girl in the class.

If, for example, the QA system returns *P1* and *H1*, an NLI system will be unable to make an accurate prediction; however, if *P2* and *H2* are returned, it could correctly predict `entailment`. Thus, without the context of the documents, an NLI system may fail based on the answers alone.

No query predictions We additionally run the zero-shot and few-shot prompting methods *without the query*: a poor performance in this setting compared to the prompts with the query means that the query is crucial to solve the task.

Prediction models We use GPT-4o, GPT-4, GPT-3.5-turbo, Gemini 1.5-pro, and Gemini-1.5-flash (OpenAI et al., 2024a,b; Team et al., 2024) to make predictions.

6 Experimental Results

Because the label distributions of the datasets are often unbalanced (see Table 2), we use *balanced accuracy* (BA, the average of the true positive

[1]For SNLI and RagTruth, two `not_entailment` examples are provided, one indicating clear contradiction and the other indicating a neutral relationship. We also provide two `not_entailment` examples for FactScore, where one indicates clear contradiction and the other demonstrates the case where one document does not contain the answer to the query.

Dataset	Model	Quality/ Ceiling	Query			No Query	
			Zero-shot	Few-shot	QA+NLI	Zero-shot	Few-shot
SNLI	GPT-4o	0.7576	0.6586	0.6877	**0.6967**	0.5626	0.5220
	GPT-4		**0.6812**	0.6642	0.6680	0.5630	0.5388
	GPT-3.5		0.6776	**0.7465**	0.6810	0.6552	0.5824
	Gem 1.5 Pro		0.6460	**0.6862**	0.6501	0.5188	0.5422
	Gem 1.5 Flash		0.5738	**0.6729**	0.6687	0.5231	0.5440
	Average		*0.6474*	***0.6915***	*0.6729*	*0.5645*	*0.5459*
RobustQA	GPT-4o	0.9394	0.6269	0.6478	**0.6660**	0.5976	0.5964
	GPT-4		0.5530	**0.6536**	0.6218	0.5556	0.6454
	GPT-3.5		0.6241	0.6271	**0.6391**	0.6077	0.4064
	Gem 1.5 Pro		0.6580	**0.6811**	0.6326	0.6208	0.6692
	Gem 1.5 Flash		0.5857	**0.6290**	0.5964	0.5607	0.5743
	Average		*0.6095*	***0.6477***	*0.6312*	*0.5885*	*0.5783*
RagTruth	GPT-4o	0.8182	0.6605	**0.7823**	0.6238	0.6153	0.7112
	GPT-4		0.6733	**0.8102**	0.6102	0.6477	0.7625
	GPT-3.5		**0.6023**	0.5526	0.5649	0.5382	0.5361
	Gem 1.5 Pro		0.6694	**0.7951**	0.5568	0.6505	0.7324
	Gem 1.5 Flash		0.6605	**0.6902**	0.5864	0.6147	0.6480
	Average		*0.6532*	***0.7261***	*0.5884*	*0.6133*	*0.678*
FactScore	GPT-4o	0.8431	**0.8189**	**0.8189**	0.7835	0.6926	0.5322
	GPT-4		0.7769	**0.7981**	0.7016	0.6632	0.6286
	GPT-3.5		**0.7593**	0.7407	0.7029	0.6537	0.5113
	Gem 1.5 Pro		**0.8263**	0.8209	0.7245	0.6433	0.6115
	Gem 1.5 Flash		0.7920	**0.8130**	0.7665	0.6779	0.6896
	Average		*0.7947*	***0.7983***	*0.7358*	*0.6661*	*0.5946*

Table 3: **All Results.** Balanced accuracy for each dataset and model for each type of prompting experiment. The overall quality found in the annotation study (Section 4.3) is also shown; this may be interpreted as a ceiling for prediction accuracy. The best score in each row is **bolded**. Note that random guessing for balanced accuracy would result in a score around 0.5.

and true negative rates) to evaluate the predictions. Note that $0 \leq \text{BA} \leq 1$, with 0.5 indicating random guessing.

6.1 Results when conditioning on the query

Table 3 shows balanced accuracy for each of the prompting methods using the GPT and Gemini family models. Clearly, few-shot prompting works best for most of the datasets and models, always achieving the highest average score among the methods where the query is provided. Not surprisingly, few-shot prompting generally beats zero-shot prompting; but importantly, few-shot prompting is generally better than the composition of QA and NLI prompts. While not better on average than few-shot prompting, the QA+NLI method works fairly well for the SNLI and RobustQA datasets, but very poorly for RagTruth and FactScore.

6.2 Results without conditioning on the query

Table 3 shows that there is a drop in accuracy among the methods that include the query and their corresponding no-query methods. Notably, the drop is very high across all model types for SNLI and FactScore, but it is not as high for RobustQA and RagTruth. The potential reasons for this are explored in Section 7.

Since the FactScore dataset has multiple queries for the same pair of documents, we additionally run a majority voting *oracle*. This simply returns the label that appears most commonly for each document pair; when there is a tie, the correct label is chosen, so this metric is an upper-bound for query-less prediction. We find that the no-query oracle gives a balanced accuracy of 0.75 for FactScore. Importantly, both zero- and few-shot prompting methods exceed this number when the query *is* provided, further illustrating the importance of conditioning on the query to solving the task.

7 Discussion

Importance of the query The dramatic drop in performance between methods that condition the prediction on the query and those that do not underscores the importance of the query. Notably, the drop is not as significant for RobustQA and RagTruth as it is for SNLI and Factscore. For RobustQA, each document provides either a "yes" or "no" perspective; thus, the query is not strictly necessary to solve the task. For RagTruth, the entire hypothesis document is the output of an LLM conditioned on the premise *and the query* – hence, providing the query to the QC-NLI task is not as helpful since the hypothesis already addresses it. However, even for these datasets, conditioning on the query is more helpful than not.

On the other hand, SNLI and FactScore exhibit dramatic decreases in performance when removing the query from the prediction task. For SNLI, this is partly due to dataset construction, where we encourage the documents to describe distinct settings. Thus, the extraneous material does not provide much information about the NLI label, meaning that the query defines the relevant aspect. Unlike SNLI, FactScore actually has *multiple* queries (and labels) per pair of documents. Thus, the query is, by design, critical to determining the label: even a majority voting oracle only achieves a balanced accuracy of 0.75, which is over 4 points lower than the average balanced accuracy on the query-conditioned methods.

The prospect of QA+NLI methods As mentioned in Section 6, few-shot prompting generally outperforms both zero-shot prompting and QA-NLI for all datasets and models. However, QA+NLI (where the system first answers the question on each document, and then makes an NLI prediction based only on the answers) performs quite well for SNLI and RobustQA. Section 5 makes the point that this type of system may not work well, depending on how the QA system answers the question; so, why does this work for SNLI and RobustQA?

For RobustQA, the questions can all be answered with "yes" or "no", and each document gives one of these perspectives (see Section 4.1). Therefore, a QA system on each of these documents will return an affirmative or negative answer, making it easy to determine the label and explaining the rather small drop in performance between the few-shot and QA+NLI systems on this dataset. The SNLI dataset does not contain yes-no questions, but contains rather literal questions about image descriptions. Hence, it is also feasible to first answer the question on each image and then perform NLI.

RagTruth, on the other hand, contains much more difficult queries. Most queries are highly open-ended and involve checking multiple sentences in the hypothesis against the premise. Thus, a QA system that does not return all necessary parts of the documents will not have the necessary information to solve the NLI task. Thus, the *documents* themselves are important, and this is an important aspect of our QC-NLI formulation.

It is less clear why QA+NLI is not as effective as the end-to-end methods for FactScore, since the questions ask about atomic facts. An analysis of the dataset shows that the queries are typically based on the *hypothesis* document only (as the goal is to check whether various facts in the hypothesis are entailed by the premise); so a possibility is that the QA result on the premise is not very informative. Further, the lower performance of QA+NLI compared to end-to-end prompting is consistent with other works that suggest that unified approaches outperform cascaded systems (Arya et al., 2023; Chowdhury et al., 2025).

This discussion suggests that some of our datasets are easier than others for QC-NLI. Future systems for the QC-NLI benchmark thus should be aware of this, and special focus may be necessary for those datasets that contain multiple sentences or facts that need to be checked across the documents.

A long way to go Finally, none of the methods achieve a balanced accuracy greater than 82.63% (Gemini 1.5 Pro on FactScore); this is only slightly below FactScore's ceiling accuracy of 84.31%, which is the quality of the dataset determined by our annotation study. Most models achieve 60-70% on the various datasets, slightly lower than their respective ceiling accuracies. Even so, while zero- and few-shot prompting methods work well in many cases, some datasets still appear very hard to solve. RobustQA, in particular, has a long way to go: the ceiling is 93.94%, but our best method achieves 68.11%. Future work can look at more sophisticated methods for solving these tasks, such as conditional neural architectures (Deshpande et al., 2023; Yoo et al., 2024; Tu et al., 2024).

8 Conclusion & Future Work

In this paper, we have proposed the novel task of Query-Conditioned Natural Language Inference (QC-NLI) and shown how it can be useful for three different applications of interest in the NLP community: fact verification, hallucination detection, and inconsistency detection for RAG. We adapt existing datasets to create a high quality benchmark for this task, where the premise and hypothesis are each multi-sentence documents. We then propose several baselines: zero-shot prompting, few-shot prompting, and QA+NLI (the composition of question-answering and NLI). The results have two primary takeaways: the *query* is important for solving the task (as evinced by the marked drop in accuracy when the query is removed from the prompts); and (2) that the *documents* are important to solving the task in the sense that first answering the query on the documents and then performing NLI inference is not always sufficient.

There are many avenues for future work. Perhaps most important is the development of higher-performing methods for QC-NLI: in particular, the best method on the RobustQA dataset performs about 15 points lower than its ceiling. More varied prompt strategies could improve performance and robustness. Further, training (or fine-tuning) models on this task may be important to improving performance, and conditional methods could be critical. Another direction would be the use of a QC-NLI system in downstream applications: for example, eliciting more holistic answers from RAG systems by providing documents that proffer different perspectives with respect to the query. In these ways, QC-NLI is a useful benchmark with much potential in the NLP community.

9 Limitations

The primary limitations of our work can be broken into two categories: shortcomings of our QC-NLI dataset and limitations of the methods to solve the QC-NLI task. One limitation of the dataset is that we only used GPT-4o (rather than several LLMs) to create it from existing datasets; further, it does not contain segment annotations indicating which parts of the documents are necessary to solve the task, which could hinder the development of methods. Further, despite relatively high annotation quality (between 75% and 94% depending on the dataset), many data points may have the wrong label; while this is a problem with any automated data generation approach, it may skew the results when attempting to solve the task. Finally, we only provide an English dataset for query-conditioned NLI; while our data adaptation method could also be applied to other languages, we do not include it in this paper.

Regarding methods for solving the task, our paper only considers prompt-based approaches; considering more varied prompt templates could improve robustness, and it is possible that the benchmark is easier to solve using parametric NLI models. Further, we do not show the use of a QC-NLI system in solving downstream tasks, such as the mitigation of hallucination or the improvement of RAG generation under inconsistent inputs; instead, we leave these studies as future work.

References

Lalaram Arya, Amartya Roy Chowdhury, and S. R. Mahadeva Prasanna. 2023. Direct vs cascaded speech-to-speech translation using transformer. In *Speech and Computer*, pages 258–270, Cham. Springer Nature Switzerland.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Guanhua Chen, Wenhan Yu, and Lei Sha. 2024. Unlocking multi-view insights in knowledge-dense retrieval-augmented generation. *Preprint*, arXiv:2404.12879.

Amartya Roy Chowdhury, Tonmoy Rajkhowa, and Sanjeev Sharma. 2025. Towards multilingual spoken visual question answering system using cross-attention. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9165–9175, Abu Dhabi, UAE. Association for Computational Linguistics.

Cleo Condoravdi, Dick Crouch, Valeria De Paiva, Reinhard Stolle, and Daniel G. Bobrow. 2003. Entailment, intensionality and text understanding. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning -*, volume 9, pages 38–45, Not Known. Association for Computational Linguistics.

Ameet Deshpande, Carlos Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. 2023. C-STS: Conditional Semantic Textual Similarity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5669–5690, Singapore. Association for Computational Linguistics.

Tanya Goyal and Greg Durrett. 2020. Evaluating Factuality in Generation with Dependency-level Entailment. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3592–3603, Online. Association for Computational Linguistics.

Rujun Han, Peng Qi, Yuhao Zhang, Lan Liu, Juliette Burger, William Yang Wang, Zhiheng Huang, Bing Xiang, and Dan Roth. 2023. RobustQA: Benchmarking the Robustness of Domain Adaptation for Open-Domain Question Answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4294–4311, Toronto, Canada. Association for Computational Linguistics.

Or Honovich, Leshem Choshen, Roee Aharoni, Ella Neeman, Idan Szpektor, and Omri Abend. 2021. Q^{2}: Evaluating Factual Consistency in Knowledge-Grounded Dialogues via Question Generation and Question Answering. *arXiv preprint*. ArXiv:2104.08202 [cs].

Youngwoo Kim, Razieh Rahimi, and James Allan. 2023. Conditional Natural Language Inference. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6833–6851, Singapore. Association for Computational Linguistics.

Yuta Koreeda and Christopher Manning. 2021. ContractNLI: A Dataset for Document-level Natural Language Inference for Contracts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1907–1919, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Puneet Mathur, Gautam Kunapuli, Riyaz Bhat, Manish Shrivastava, Dinesh Manocha, and Maneesh Singh. 2022. DocInfer: Document-level Natural Language Inference using Optimal Evidence Selection. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 809–824, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Niels Mündler, Jingxuan He, Slobodan Jenko, and Martin Vechev. 2024. Self-contradictory Hallucinations of Large Language Models: Evaluation, Detection and Mitigation. *arXiv preprint*. ArXiv:2305.15852 [cs].

Cheng Niu, Yuanhao Wu, Juno Zhu, Siliang Xu, Kashun Shum, Randy Zhong, Juntong Song, and Tong Zhang. 2024. RAGTruth: A Hallucination Corpus for Developing Trustworthy Retrieval-Augmented Language Models. *arXiv preprint*. ArXiv:2401.00396 [cs].

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan

Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024b. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Mobashir Sadat and Cornelia Caragea. 2024. MSciNLI: A diverse benchmark for scientific natural language inference. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1610–1629, Mexico City, Mexico. Association for Computational Linguistics.

Julius Steen, Juri Opitz, Anette Frank, and Katja Markert. 2023. With a Little Push, NLI Models can Robustly and Efficiently Predict Faithfulness. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 914–924, Toronto, Canada. Association for Computational Linguistics.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Sercinoglu, Zach Gleicher, Juliette Love, Paul Voigtlaender, Rohan Jain, Gabriela Surita, Kareem Mohamed, Rory Blevins, Junwhan Ahn, Tao Zhu, Kornraphop Kawintiranon, Orhan Firat, Yiming Gu, Yujing Zhang, Matthew Rahtz, Manaal Faruqui, Natalie Clay, Justin Gilmer, JD Co-Reyes, Ivo Penchev, Rui Zhu, Nobuyuki Morioka, Kevin Hui, Krishna Haridasan, Victor Campos, Mahdis Mahdieh, Mandy Guo, Samer Hassan, Kevin Kilgour, Arpi Vezer, Heng-Tze Cheng, Raoul de Liedekerke, Siddharth Goyal, Paul Barham, DJ Strouse, Seb Noury, Jonas Adler, Mukund Sundararajan, Sharad Vikram, Dmitry Lepikhin, Michela Paganini, Xavier Garcia, Fan Yang, Dasha Valter, Maja Trebacz, Kiran Vodrahalli, Chulayuth Asawaroengchai, Roman Ring, Norbert Kalb, Livio Baldini Soares, Siddhartha Brahma, David Steiner, Tianhe Yu, Fabian Mentzer, Antoine He, Lucas Gonzalez, Bibo Xu, Raphael Lopez Kaufman, Laurent El Shafey, Junhyuk Oh, Tom Hennigan, George van den Driessche, Seth Odoom, Mario Lucic, Becca Roelofs, Sid Lall, Amit Marathe, Betty Chan, Santiago Ontanon, Luheng He, Denis Teplyashin, Jonathan Lai, Phil Crone, Bogdan Damoc, Lewis Ho, Sebastian Riedel, Karel Lenc, Chih-Kuan Yeh, Aakanksha Chowdhery, Yang Xu, Mehran Kazemi, Ehsan Amid, Anastasia Petrushkina, Kevin Swersky, Ali Khodaei, Gowoon Chen, Chris Larkin, Mario Pinto, Geng Yan, Adria Puigdomenech Badia, Piyush Patil, Steven Hansen, Dave Orr, Sebastien M. R. Arnold, Jordan Grimstad, Andrew Dai, Sholto Douglas, Rishika Sinha, Vikas Yadav, Xi Chen, Elena Gribovskaya, Jacob Austin, Jeffrey Zhao, Kaushal Patel, Paul Komarek, Sophia Austin, Sebastian Borgeaud, Linda Friso, Abhimanyu Goyal, Ben Caine, Kris Cao, Da-Woon Chung, Matthew Lamm, Gabe Barth-Maron, Thais Kagohara, Kate Olszewska, Mia Chen, Kaushik Shivakumar, Rishabh Agarwal, Harshal Godhia, Ravi Rajwar, Javier Snaider, Xerxes Dotiwalla, Yuan Liu, Aditya Barua, Victor Ungureanu, Yuan Zhang, Bat-Orgil Batsaikhan, Mateo Wirth, James Qin, Ivo Danihelka, Tulsee Doshi, Martin Chadwick, Jilin Chen, Sanil Jain, Quoc Le, Arjun Kar, Madhu Gurumurthy, Cheng Li, Ruoxin Sang, Fangyu Liu, Lampros Lamprou, Rich Munoz, Nathan Lintz, Harsh Mehta, Heidi Howard, Malcolm Reynolds, Lora Aroyo, Quan Wang, Lorenzo Blanco, Albin Cassirer, Jordan Griffith, Dipanjan Das, Stephan Lee, Jakub Sygnowski, Zach Fisher, James Besley, Richard Powell, Zafarali Ahmed, Dominik Paulus, David Reitter, Zalan Borsos, Rishabh Joshi, Aedan Pope, Steven Hand, Vittorio Selo, Vihan Jain, Nikhil Sethi, Megha Goel, Takaki Makino, Rhys May, Zhen Yang, Johan Schalkwyk, Christina Butterfield, Anja Hauth, Alex Goldin, Will Hawkins, Evan Senter, Sergey Brin, Oliver Woodman, Marvin Ritter, Eric Noland, Minh Giang, Vijay Bolina, Lisa Lee, Tim Blyth, Ian Mackinnon, Machel Reid, Obaid Sarvana, David Silver, Alexander Chen, Lily Wang, Loren Maggiore, Oscar Chang, Nithya Attaluri, Gregory Thornton, Chung-Cheng Chiu, Oskar Bunyan, Nir Levine, Timothy Chung, Evgenii Eltyshev, Xiance Si, Timothy Lillicrap, Demetra Brady, Vaibhav Aggarwal, Boxi Wu, Yuanzhong Xu, Ross McIlroy, Kartikeya Badola, Paramjit Sandhu, Erica Moreira, Wojciech Stokowiec, Ross Hemsley, Dong Li, Alex Tudor, Pranav Shyam, Elahe Rahimtoroghi, Salem Haykal, Pablo Sprechmann, Xiang Zhou, Diana Mincu, Yujia Li, Ravi Addanki, Kalpesh Krishna, Xiao Wu, Alexandre Frechette, Matan Eyal, Allan Dafoe, Dave Lacey, Jay Whang, Thi Avrahami, Ye Zhang, Emanuel Taropa, Hanzhao Lin, Daniel Toyama, Eliza Rutherford, Motoki Sano, HyunJeong Choe, Alex Tomala, Chalence Safranek-Shrader, Nora Kassner, Mantas Pajarskas, Matt Harvey, Sean Sechrist, Meire Fortunato, Christina Lyu, Gamaleldin Elsayed, Chenkai Kuang, James Lottes, Eric Chu, Chao Jia, Chih-Wei Chen, Peter Humphreys, Kate Baumli, Connie Tao, Rajkumar Samuel, Cicero Nogueira dos Santos, Anders Andreassen, Nemanja Rakićević, Dominik Grewe, Aviral Kumar, Stephanie Winkler, Jonathan Caton, Andrew Brock, Sid Dalmia, Hannah Sheahan, Iain Barr, Yingjie Miao, Paul Natsev, Jacob Devlin, Feryal Behbahani, Flavien Prost, Yanhua Sun, Artiom Myaskovsky, Thanumalayan Sankaranarayana Pillai, Dan Hurt, Angeliki Lazaridou, Xi Xiong, Ce Zheng, Fabio Pardo, Xiaowei Li, Dan Horgan, Joe Stanton, Moran Ambar, Fei Xia, Alejandro Lince, Mingqiu Wang, Basil Mustafa, Albert Webson, Hyo Lee, Rohan Anil, Martin Wicke, Timothy Dozat, Abhishek Sinha, Enrique Piqueras, Elahe Dabir, Shyam Upadhyay, Anudhyan Boral, Lisa Anne Hendricks, Corey Fry, Josip Djolonga, Yi Su, Jake Walker, Jane Labanowski, Ronny Huang, Vedant Misra, Jeremy

Chen, RJ Skerry-Ryan, Avi Singh, Shruti Rijhwani, Dian Yu, Alex Castro-Ros, Beer Changpinyo, Romina Datta, Sumit Bagri, Arnar Mar Hrafnkelsson, Marcello Maggioni, Daniel Zheng, Yury Sulsky, Shaobo Hou, Tom Le Paine, Antoine Yang, Jason Riesa, Dominika Rogozinska, Dror Marcus, Dalia El Badawy, Qiao Zhang, Luyu Wang, Helen Miller, Jeremy Greer, Lars Lowe Sjos, Azade Nova, Heiga Zen, Rahma Chaabouni, Mihaela Rosca, Jiepu Jiang, Charlie Chen, Ruibo Liu, Tara Sainath, Maxim Krikun, Alex Polozov, Jean-Baptiste Lespiau, Josh Newlan, Zeyncep Cankara, Soo Kwak, Yunhan Xu, Phil Chen, Andy Coenen, Clemens Meyer, Katerina Tsihlas, Ada Ma, Juraj Gottweis, Jinwei Xing, Chenjie Gu, Jin Miao, Christian Frank, Zeynep Cankara, Sanjay Ganapathy, Ishita Dasgupta, Steph Hughes-Fitt, Heng Chen, David Reid, Keran Rong, Hongmin Fan, Joost van Amersfoort, Vincent Zhuang, Aaron Cohen, Shixiang Shane Gu, Anhad Mohananey, Anastasija Ilic, Taylor Tobin, John Wieting, Anna Bortsova, Phoebe Thacker, Emma Wang, Emily Caveness, Justin Chiu, Eren Sezener, Alex Kaskasoli, Steven Baker, Katie Millican, Mohamed Elhawaty, Kostas Aisopos, Carl Lebsack, Nathan Byrd, Hanjun Dai, Wenhao Jia, Matthew Wiethoff, Elnaz Davoodi, Albert Weston, Lakshman Yagati, Arun Ahuja, Isabel Gao, Golan Pundak, Susan Zhang, Michael Azzam, Khe Chai Sim, Sergi Caelles, James Keeling, Abhanshu Sharma, Andy Swing, YaGuang Li, Chenxi Liu, Carrie Grimes Bostock, Yamini Bansal, Zachary Nado, Ankesh Anand, Josh Lipschultz, Abhijit Karmarkar, Lev Proleev, Abe Ittycheriah, Soheil Hassas Yeganeh, George Polovets, Aleksandra Faust, Jiao Sun, Alban Rrustemi, Pen Li, Rakesh Shivanna, Jeremiah Liu, Chris Welty, Federico Lebron, Anirudh Baddepudi, Sebastian Krause, Emilio Parisotto, Radu Soricut, Zheng Xu, Dawn Bloxwich, Melvin Johnson, Behnam Neyshabur, Justin Mao-Jones, Renshen Wang, Vinay Ramasesh, Zaheer Abbas, Arthur Guez, Constant Segal, Duc Dung Nguyen, James Svensson, Le Hou, Sarah York, Kieran Milan, Sophie Bridgers, Wiktor Gworek, Marco Tagliasacchi, James Lee-Thorp, Michael Chang, Alexey Guseynov, Ale Jakse Hartman, Michael Kwong, Ruizhe Zhao, Sheleem Kashem, Elizabeth Cole, Antoine Miech, Richard Tanburn, Mary Phuong, Filip Pavetic, Sebastien Cevey, Ramona Comanescu, Richard Ives, Sherry Yang, Cosmo Du, Bo Li, Zizhao Zhang, Mariko Iinuma, Clara Huiyi Hu, Aurko Roy, Shaan Bijwadia, Zhenkai Zhu, Danilo Martins, Rachel Saputro, Anita Gergely, Steven Zheng, Dawei Jia, Ioannis Antonoglou, Adam Sadovsky, Shane Gu, Yingying Bi, Alek Andreev, Sina Samangooei, Mina Khan, Tomas Kocisky, Angelos Filos, Chintu Kumar, Colton Bishop, Adams Yu, Sarah Hodkinson, Sid Mittal, Premal Shah, Alexandre Moufarek, Yong Cheng, Adam Bloniarz, Jaehoon Lee, Pedram Pejman, Paul Michel, Stephen Spencer, Vladimir Feinberg, Xuehan Xiong, Nikolay Savinov, Charlotte Smith, Siamak Shakeri, Dustin Tran, Mary Chesus, Bernd Bohnet, George Tucker, Tamara von Glehn, Carrie Muir, Yiran Mao, Hideto Kazawa, Ambrose Slone, Kedar Soparkar, Disha Shrivastava, James Cobon-Kerr, Michael Sharman, Jay Pavagadhi, Carlos Araya, Karolis Misiunas, Nimesh Ghelani, Michael Laskin, David Barker, Qiujia Li, Anton Briukhov, Neil Houlsby, Mia Glaese, Balaji Lakshminarayanan, Nathan Schucher, Yunhao Tang, Eli Collins, Hyeontaek Lim, Fangxiaoyu Feng, Adria Recasens, Guangda Lai, Alberto Magni, Nicola De Cao, Aditya Siddhant, Zoe Ashwood, Jordi Orbay, Mostafa Dehghani, Jenny Brennan, Yifan He, Kelvin Xu, Yang Gao, Carl Saroufim, James Molloy, Xinyi Wu, Seb Arnold, Solomon Chang, Julian Schrittwieser, Elena Buchatskaya, Soroush Radpour, Martin Polacek, Skye Giordano, Ankur Bapna, Simon Tokumine, Vincent Hellendoorn, Thibault Sottiaux, Sarah Cogan, Aliaksei Severyn, Mohammad Saleh, Shantanu Thakoor, Laurent Shefey, Siyuan Qiao, Meenu Gaba, Shuo yiin Chang, Craig Swanson, Biao Zhang, Benjamin Lee, Paul Kishan Rubenstein, Gan Song, Tom Kwiatkowski, Anna Koop, Ajay Kannan, David Kao, Parker Schuh, Axel Stjerngren, Golnaz Ghiasi, Gena Gibson, Luke Vilnis, Ye Yuan, Felipe Tiengo Ferreira, Aishwarya Kamath, Ted Klimenko, Ken Franko, Kefan Xiao, Indro Bhattacharya, Miteyan Patel, Rui Wang, Alex Morris, Robin Strudel, Vivek Sharma, Peter Choy, Sayed Hadi Hashemi, Jessica Landon, Mara Finkelstein, Priya Jhakra, Justin Frye, Megan Barnes, Matthew Mauger, Dennis Daun, Khuslen Baatarsukh, Matthew Tung, Wael Farhan, Henryk Michalewski, Fabio Viola, Felix de Chaumont Quitry, Charline Le Lan, Tom Hudson, Qingze Wang, Felix Fischer, Ivy Zheng, Elspeth White, Anca Dragan, Jean baptiste Alayrac, Eric Ni, Alexander Pritzel, Adam Iwanicki, Michael Isard, Anna Bulanova, Lukas Zilka, Ethan Dyer, Devendra Sachan, Srivatsan Srinivasan, Hannah Muckenhirn, Honglong Cai, Amol Mandhane, Mukarram Tariq, Jack W. Rae, Gary Wang, Kareem Ayoub, Nicholas FitzGerald, Yao Zhao, Woohyun Han, Chris Alberti, Dan Garrette, Kashyap Krishnakumar, Mai Gimenez, Anselm Levskaya, Daniel Sohn, Josip Matak, Inaki Iturrate, Michael B. Chang, Jackie Xiang, Yuan Cao, Nishant Ranka, Geoff Brown, Adrian Hutter, Vahab Mirrokni, Nanxin Chen, Kaisheng Yao, Zoltan Egyed, Francois Galilee, Tyler Liechty, Praveen Kallakuri, Evan Palmer, Sanjay Ghemawat, Jasmine Liu, David Tao, Chloe Thornton, Tim Green, Mimi Jasarevic, Sharon Lin, Victor Cotruta, Yi-Xuan Tan, Noah Fiedel, Hongkun Yu, Ed Chi, Alexander Neitz, Jens Heitkaemper, Anu Sinha, Denny Zhou, Yi Sun, Charbel Kaed, Brice Hulse, Swaroop Mishra, Maria Georgaki, Sneha Kudugunta, Clement Farabet, Izhak Shafran, Daniel Vlasic, Anton Tsitsulin, Rajagopal Ananthanarayanan, Alen Carin, Guolong Su, Pei Sun, Shashank V, Gabriel Carvajal, Josef Broder, Iulia Comsa, Alena Repina, William Wong, Warren Weilun Chen, Peter Hawkins, Egor Filonov, Lucia Loher, Christoph Hirnschall, Weiyi Wang, Jingchen Ye, Andrea Burns, Hardie Cate, Diana Gage Wright, Federico Piccinini, Lei Zhang, Chu-Cheng Lin, Ionel Gog, Yana Kulizhskaya, Ashwin Sreevatsa, Shuang Song, Luis C. Cobo, Anand Iyer, Chetan Tekur, Guillermo Garrido, Zhuyun Xiao, Rupert Kemp, Huaixiu Steven Zheng, Hui Li, Ananth Agarwal, Christel Ngani, Kati Goshvadi, Rebeca Santamaria-Fernandez, Woj-

ciech Fica, Xinyun Chen, Chris Gorgolewski, Sean Sun, Roopal Garg, Xinyu Ye, S. M. Ali Eslami, Nan Hua, Jon Simon, Pratik Joshi, Yelin Kim, Ian Tenney, Sahitya Potluri, Lam Nguyen Thiet, Quan Yuan, Florian Luisier, Alexandra Chronopoulou, Salvatore Scellato, Praveen Srinivasan, Minmin Chen, Vinod Koverkathu, Valentin Dalibard, Yaming Xu, Brennan Saeta, Keith Anderson, Thibault Sellam, Nick Fernando, Fantine Huot, Junehyuk Jung, Mani Varadarajan, Michael Quinn, Amit Raul, Maigo Le, Ruslan Habalov, Jon Clark, Komal Jalan, Kalesha Bullard, Achintya Singhal, Thang Luong, Boyu Wang, Sujeevan Rajayogam, Julian Eisenschlos, Johnson Jia, Daniel Finchelstein, Alex Yakubovich, Daniel Balle, Michael Fink, Sameer Agarwal, Jing Li, Dj Dvijotham, Shalini Pal, Kai Kang, Jaclyn Konzelmann, Jennifer Beattie, Olivier Dousse, Diane Wu, Remi Crocker, Chen Elkind, Siddhartha Reddy Jonnalagadda, Jong Lee, Dan Holtmann-Rice, Krystal Kallarackal, Rosanne Liu, Denis Vnukov, Neera Vats, Luca Invernizzi, Mohsen Jafari, Huanjie Zhou, Lilly Taylor, Jennifer Prendki, Marcus Wu, Tom Eccles, Tianqi Liu, Kavya Kopparapu, Francoise Beaufays, Christof Angermueller, Andreea Marzoca, Shourya Sarcar, Hilal Dib, Jeff Stanway, Frank Perbet, Nejc Trdin, Rachel Sterneck, Andrey Khorlin, Dinghua Li, Xihui Wu, Sonam Goenka, David Madras, Sasha Goldshtein, Willi Gierke, Tong Zhou, Yaxin Liu, Yannie Liang, Anais White, Yunjie Li, Shreya Singh, Sanaz Bahargam, Mark Epstein, Sujoy Basu, Li Lao, Adnan Ozturel, Carl Crous, Alex Zhai, Han Lu, Zora Tung, Neeraj Gaur, Alanna Walton, Lucas Dixon, Ming Zhang, Amir Globerson, Grant Uy, Andrew Bolt, Olivia Wiles, Milad Nasr, Ilia Shumailov, Marco Selvi, Francesco Piccinno, Ricardo Aguilar, Sara McCarthy, Misha Khalman, Mrinal Shukla, Vlado Galic, John Carpenter, Kevin Villela, Haibin Zhang, Harry Richardson, James Martens, Matko Bosnjak, Shreyas Rammohan Belle, Jeff Seibert, Mahmoud Alnahlawi, Brian McWilliams, Sankalp Singh, Annie Louis, Wen Ding, Dan Popovici, Lenin Simicich, Laura Knight, Pulkit Mehta, Nishesh Gupta, Chongyang Shi, Saaber Fatehi, Jovana Mitrovic, Alex Grills, Joseph Pagadora, Tsendsuren Munkhdalai, Dessie Petrova, Danielle Eisenbud, Zhishuai Zhang, Damion Yates, Bhavishya Mittal, Nilesh Tripuraneni, Yannis Assael, Thomas Brovelli, Prateek Jain, Mihajlo Velimirovic, Canfer Akbulut, Jiaqi Mu, Wolfgang Macherey, Ravin Kumar, Jun Xu, Haroon Qureshi, Gheorghe Comanici, Jeremy Wiesner, Zhitao Gong, Anton Ruddock, Matthias Bauer, Nick Felt, Anirudh GP, Anurag Arnab, Dustin Zelle, Jonas Rothfuss, Bill Rosgen, Ashish Shenoy, Bryan Seybold, Xinjian Li, Jayaram Mudigonda, Goker Erdogan, Jiawei Xia, Jiri Simsa, Andrea Michi, Yi Yao, Christopher Yew, Steven Kan, Isaac Caswell, Carey Radebaugh, Andre Elisseeff, Pedro Valenzuela, Kay McKinney, Kim Paterson, Albert Cui, Eri Latorre-Chimoto, Solomon Kim, William Zeng, Ken Durden, Priya Ponnapalli, Tiberiu Sosea, Christopher A. Choquette-Choo, James Manyika, Brona Robenek, Harsha Vashisht, Sebastien Pereira, Hoi Lam, Marko Velic, Denese Owusu-Afriyie, Katherine Lee, Tolga Bolukbasi, Alicia Parrish, Shawn Lu, Jane Park, Balaji Venkatraman, Alice Talbert, Lambert Rosique, Yuchung Cheng, Andrei Sozanschi, Adam Paszke, Praveen Kumar, Jessica Austin, Lu Li, Khalid Salama, Bartek Perz, Wooyeol Kim, Nandita Dukkipati, Anthony Baryshnikov, Christos Kaplanis, XiangHai Sheng, Yuri Chervonyi, Caglar Unlu, Diego de Las Casas, Harry Askham, Kathryn Tunyasuvunakool, Felix Gimeno, Siim Poder, Chester Kwak, Matt Miecnikowski, Vahab Mirrokni, Alek Dimitriev, Aaron Parisi, Dangyi Liu, Tomy Tsai, Toby Shevlane, Christina Kouridi, Drew Garmon, Adrian Goedeckemeyer, Adam R. Brown, Anitha Vijayakumar, Ali Elqursh, Sadegh Jazayeri, Jin Huang, Sara Mc Carthy, Jay Hoover, Lucy Kim, Sandeep Kumar, Wei Chen, Courtney Biles, Garrett Bingham, Evan Rosen, Lisa Wang, Qijun Tan, David Engel, Francesco Pongetti, Dario de Cesare, Dongseong Hwang, Lily Yu, Jennifer Pullman, Srini Narayanan, Kyle Levin, Siddharth Gopal, Megan Li, Asaf Aharoni, Trieu Trinh, Jessica Lo, Norman Casagrande, Roopali Vij, Loic Matthey, Bramandia Ramadhana, Austin Matthews, CJ Carey, Matthew Johnson, Kremena Goranova, Rohin Shah, Shereen Ashraf, Kingshuk Dasgupta, Rasmus Larsen, Yicheng Wang, Manish Reddy Vuyyuru, Chong Jiang, Joana Ijazi, Kazuki Osawa, Celine Smith, Ramya Sree Boppana, Taylan Bilal, Yuma Koizumi, Ying Xu, Yasemin Altun, Nir Shabat, Ben Bariach, Alex Korchemniy, Kiam Choo, Olaf Ronneberger, Chimezie Iwuanyanwu, Shubin Zhao, David Soergel, Cho-Jui Hsieh, Irene Cai, Shariq Iqbal, Martin Sundermeyer, Zhe Chen, Elie Bursztein, Chaitanya Malaviya, Fadi Biadsy, Prakash Shroff, Inderjit Dhillon, Tejasi Latkar, Chris Dyer, Hannah Forbes, Massimo Nicosia, Vitaly Nikolaev, Somer Greene, Marin Georgiev, Pidong Wang, Nina Martin, Hanie Sedghi, John Zhang, Praseem Banzal, Doug Fritz, Vikram Rao, Xuezhi Wang, Jiageng Zhang, Viorica Patraucean, Dayou Du, Igor Mordatch, Ivan Jurin, Lewis Liu, Ayush Dubey, Abhi Mohan, Janek Nowakowski, Vlad-Doru Ion, Nan Wei, Reiko Tojo, Maria Abi Raad, Drew A. Hudson, Vaishakh Keshava, Shubham Agrawal, Kevin Ramirez, Zhichun Wu, Hoang Nguyen, Ji Liu, Madhavi Sewak, Bryce Petrini, DongHyun Choi, Ivan Philips, Ziyue Wang, Ioana Bica, Ankush Garg, Jarek Wilkiewicz, Priyanka Agrawal, Xiaowei Li, Danhao Guo, Emily Xue, Naseer Shaik, Andrew Leach, Sadh MNM Khan, Julia Wiesinger, Sammy Jerome, Abhishek Chakladar, Alek Wenjiao Wang, Tina Ornduff, Folake Abu, Alireza Ghaffarkhah, Marcus Wainwright, Mario Cortes, Frederick Liu, Joshua Maynez, Andreas Terzis, Pouya Samangouei, Riham Mansour, Tomasz Kępa, François-Xavier Aubet, Anton Algymr, Dan Banica, Agoston Weisz, Andras Orban, Alexandre Senges, Ewa Andrejczuk, Mark Geller, Niccolo Dal Santo, Valentin Anklin, Majd Al Merey, Martin Baeuml, Trevor Strohman, Junwen Bai, Slav Petrov, Yonghui Wu, Demis Hassabis, Koray Kavukcuoglu, Jeff Dean, and Oriol Vinyals. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Jingxuan Tu, Keer Xu, Liulu Yue, Bingyang Ye, Kyeongmin Rim, and James Pustejovsky. 2024. Linguistically Conditioned Semantic Textual Similarity. *arXiv preprint*. ArXiv:2406.03673 [cs].

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Edwin B. Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

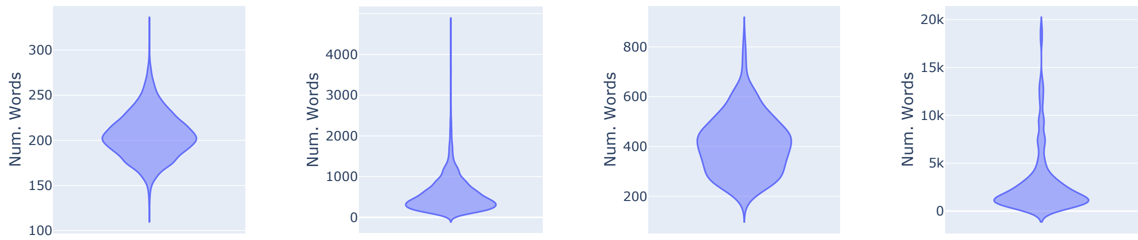
Wenpeng Yin, Dragomir Radev, and Caiming Xiong. 2021. DocNLI: A Large-scale Dataset for Document-level Natural Language Inference. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4913–4922, Online. Association for Computational Linguistics.

Young Hyun Yoo, Jii Cha, Changhyeon Kim, and Taeuk Kim. 2024. Hyper-CL: Conditioning Sentence Representations with Hypernetworks. *arXiv preprint*. ArXiv:2403.09490 [cs].

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. AlignScore: Evaluating Factual Consistency with a Unified Alignment Function. *arXiv preprint*. ArXiv:2305.16739 [cs].

Shuo Zhang, Liangming Pan, Junzhou Zhao, and William Yang Wang. 2023a. The Knowledge Alignment Problem: Bridging Human and External Knowledge for Large Language Models. *arXiv preprint*. ArXiv:2305.13669 [cs].

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023b. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. *arXiv preprint*. ArXiv:2309.01219 [cs].



(a) SNLI (b) RobustQA (c) RagTruth (d) FactScore

Figure 3: **Lengths of examples in each dataset.** Violin plots of the number of words in the combined premise and hypothesis documents for each dataset.

A Additional Dataset Statistics

In addition to the statistics reported in Section 4.2, we report violin plots of the number of words in the combined premise and hypothesis documents of each dataset. This is depicted in Figure 3. The results show that SNLI and RagTruth each have fewer than 1,000 words in their documents while RobustQA and FactScore are much larger; in some cases FactScore has nearly 20,000 words.

B Dataset Examples

For each QC-NLI dataset, we list at least one example for each label, using the labels assigned by annotators (see Section 4.1).

B.1 SNLI

Example: `entailment`

> **Document 1 (Premise):** A lady dressed in a green and black jacket paired with blue pants stood at the edge of a grassy field. The horse trotted gracefully a few feet away, its mane swaying in the gentle breeze. With the sun shining brightly overhead, she positioned her camera to capture the perfect moment. Nearby, the sounds of birds chirping added to the serene ambiance. Leaning slightly to adjust her angle, she focused intently on the horse, ensuring the details were perfectly framed in her lens. A sense of enthusiasm and concentration was clearly visible on her face as she clicked several shots.
>
> **Document 2 (Hypothesis):** In a bustling city park, a woman busily snapped photos of a horse that trotted elegantly across the grassy field. Her camera clicked rhythmically as she captured the animal's swift movements and serene grace. Amidst the distant sounds of traffic and children playing nearby, she remained focused, adjusting her camera settings as needed. Her interest was solely on the horse, which seemed to pose naturally for the snapshots. This was a peaceful afternoon, defined by her pastime with her camera.
>
> **Query:** Who is capturing images of the horse?

Example: `not_entailment`

> **Document 1 (Premise):** Under the warm afternoon sun, the golden sand stretches along the shoreline where two golden retrievers appear to be playing joyfully. Waves roll gently onto the beach as one of the dogs nuzzles a blue ball, its coat glistening in the sunlight. With a playful bark, it grabs the ball and trots over to its companion, dropping it at its feet. The second dog wags its tail eagerly, picking up the ball with its mouth. Together, they move back and forth, the ball frequently exchanging paws between them.
>
> **Document 2 (Hypothesis):** On a bright, sunny day in a bustling city park, the sound of children playing competes with the barks of two golden retrievers. The dogs wander along the grassy

expanse, giving off an impression that they should be enjoying their games. One dog holds a prized blue ball in its mouth, its eyes darting at its companion. However, both dogs stand in a tense standoff, each keenly holding onto their own space and refusing to let the ball pass between them. This standstill extends into a patience-testing silence, broken only by the distant hum of city life.

Query: What is the interaction between the dogs involving the ball?

B.2 RobustQA

Example: `contradiction`

Document 1 (Premise): Great question! Taking a few minutes to think through this logically should come up with something. First of all, realise that a larger battery with a higher amperage (CCA) will not cause any problems on the drawing end. In other words, you're not sending too much electricity to some component or something. The higher amperage simply means that more energy is available, not that more energy is flowing through the components. I'm sure you already knew that; just making sure everyone is on the same page. Secondly, you've got to realise that it doesn't take any more energy to charge up a bigger battery than a smaller battery. That is, provided they have had equal draws. The difference, of course, is that the bigger battery can get uncharged further, resulting in a larger draw on the alternator. Basically, this means that your alternator will have to work harder to recharge the battery, but simply because the larger battery was able to output more juice. Since the alternator wear is usually due to age, the answer is yes, the bigger battery will wear down your alternator more. That being said, it probably won't be significant, unless you regularly leave your radio running overnight, because your alternator is always running when the car is running; the only difference is that it will be demanding more current from the alternator to recharge the battery. Therefore, as far as I and my research are concerned, the wear is Probably insignificant.

Document 2 (Hypothesis): In fact the larger capacity battery may, under certain circumstances, increase your alternator's life. Many vehicles with enhanced sound systems install a larger battery to smooth out the load spikes on high-draw portions of the music. A typical electrical system upgrade path would start with a higher-capacity battery, then a higher-capacity alternator or even dual alternators, then dual high-capacity batteries.

Query: will fitting a physically bigger battery reduce the working life of the alternator?

Example: `not_contradiction`

Document 1 (Premise): Yes you should be able to in my opinion, as an above poster mentioned the water doesn't actually stop you from being able to talk or generating the pitches needed to cast a spell. While you are talking underwater, from the point the air goes through you vocal box, and the sounds are shaped with your tongue and lips all the way up to hitting the Water outside of your mouth it is going to sound exactly as it does out of water. (While you talk you're going to generate a bubble of air in front of you that the sound can travel normally a little ways through.) After it hits the water the sound losses most of its energy due to the much higher impedance of water compared to the sound in air. (It's still there though and the same exact pitch, just very quite and harder for use to hear because our ears work differently under water.) But the point still stands that until it hits that water it sounds 100

Document 2 (Hypothesis): Yes To play Devil's Advocate here, it is not strictly RAW to disallow spellcasting. Of course, it is a sensible house rule – but it would be a house rule if you disallowed it. Jeremy Crawford says you can This tweet from Jeremy Crawford explicitly states that being underwater doesn't interfere with spellcasting. There is no conditional "Yes, if they can breathe underwater" JC says you can, but only if you can breathe underwater? Another tweet from Jeremy Crawford says that, if you can breathe underwater, you can perform the verbal components of spells. Fair enough. However, this is NOT the same as "if you can't breathe underwater, you can't perform the verbal components of spells" either. Just as saying "if you can sing, you have a voice" is true, but "if you can't sing, you don't have a voice" is not necessarily true. Again, strictly speaking, nothing is disallowing spellcasting here yet. The PHB says you have to be able to talk? As @NautArch has shown, the PHB does mention a rule on V components of spells that seems like it should affect spellcasting. Most spells require the chanting of mystic words. The words themselves aren't the source of the spell's power; rather, the particular combination of sounds, with specific pitch and resonance, sets the threads of magic in motion. Thus, a character who is gagged or in an area of silence, such as one created by the silence spell, can't cast a spell with a verbal component. This question on Quora asks if we can talk normally underwater. Well, the answer is yes, we can speak normally. The question is just, can the person you're speaking to understand you? Well, in spellcasting, nobody needs to understand you. You just need to produce mystic words that form a combination of sounds, with a specific pitch and resonance. Note that you can always do this underwater, it's just that the sound is formed in your larynx and becomes distorted as soon as it touches the water. But the rules don't say "the sounds must reach outside your larynx" or "others must hear you clearly". You can technically still do it. Moreover, every spellcaster will likely have different ways of casting the same spell, just because they naturally have different voices. It is not against the rules to consider that there are multiple ways you can set pitch and resonance, but still cause the weaves of magic to be set in motion in the same way. So, sound can still travel through water. Why can't a magic user speak those mystic words in a way that, when the sound travels underwater, the specific pitch and resonance still matches what is needed to pull off the spell? RAW, this is not illegal. But Gagged prohibits spellcasting, so why doesn't being underwater? There are many ways to wave this away. Any answer I give will not be RAW, and is in DM fiat territory absolutely. Nonetheless, you can argue that when you are gagged, your tongue cannot move about and you cannot shape the sounds and words precisely because of this, whereas being underwater does not forbid this. You can also say that being gagged restricts your jaw movement, but being underwater doesn't, so you still retain enough control to be able to cast while submerged. Sensible House Rules Casting underwater is different from casting in air, this is true. How you handle this is up to you. This Enworld discussion shows a few ways other DMs handle it, in the order of their appearance in that thread: Spellcasting is totally disallowed underwater unless the caster can speak underwater Allow spellcasting underwater without penalties, as there is no rule actually forbidding it Have the caster perform a check. On a failure, the spell slot is not wasted, but the action is lost. But only do this if: 1) there was a way around this issue, or 2) being in the water is intended to be a penalty. Otherwise, just let the casters cast normally. If spellcasting is penalized underwater, non-casters must be similarly penalized Allow one spell to be cast, but then immediately have the caster start drowning Require a concentration check before casting a spell Disallow spellcasting for a one-off encounter, but allow it if the characters are expected to be under the water for extended periods Another definitive JC Tweet: You can cast a single spell underwater, but afterwards you begin drowning if you can't breathe underwater. Otherwise, no rule prevents verbal components from working underwater. Thanks to @mxyzplk for bringing this tweet to my attention! Extra Note: In that Twitter thread, Dan Dillon asks: Is the intent that if you cast a verbal component spell you're no longer holding your breath (and now on Con mod +1 rnds)? To which Jeremy replies "yes." But technically it is Con mod rounds only, without the +1. The +1 only happens when you've held your breath and have Con mod + 1 minutes before you start drowning.

Query: can spells with a verbal component be cast underwater if the caster cant breathe water?

B.3 RagTruth

Example: `entailment`

Document 1 (Premise): passage 1:Monoamniotic twins occur when the split takes place after the ninth day after fertilization. Conjoined twins: When the division of the developing zygote into 2 embryos occurs, 99% of the time it is within 8 days of fertilization. Mortality is highest for conjoined twins due to the many complications resulting from shared organs.\n\n passage 2:Dizygotic twin pregnancies are slightly more likely when the following factors are present in the woman: 1 She is of West African descent (especially Yoruba) 2 She is between the age of 30 and 40 years. 3 She is greater than average height and weight. 4 She has had several previous pregnancies.\n\n passage 3:Twins are two offspring produced by the same pregnancy. Twins can be either monozygotic (identical), meaning that they develop from one zygote, which splits and forms two embryos, or dizygotic (fraternal), meaning that they develop from two different eggs. In fraternal twins, each twin is fertilized by its own sperm cell. In contrast, a fetus that develops alone in the womb is called a singleton, and the general term for one offspring of a multiple birth is multiple. Non-related look-alikes whose resemblance parallels that of twins are referred to as doppelgangers.\n\n

Document 2 (Hypothesis): Twins can happen in two ways: monozygotic (identical) twins occur when one zygote splits into two embryos, while dizygotic (fraternal) twins happen when two different eggs are fertilized by two separate sperm cells. The likelihood of dizygotic twins is influenced by factors such as West African descent, being between 30 and 40 years old, being taller and heavier than average, and having had several previous pregnancies. Conjoined twins occur when the division of the zygote happens within 8 days of fertilization and can result in complications due to shared organs.

Query: how do twins happen

Example: `not_entailment`

Document 1 (Premise): passage 1:Children proudly wave the American flag during a children naturalization ceremony in July 2011 at a stadium in Manchester, NH. (Photo courtesy of USCIS). Immigration has shaped the United States as a nation since the first newcomers arrived over 400 years ago.\n\n passage 2:Prompt: What were the long-term economic, social, and political effects of Reconstruction? The United States was challenged with many issues after the Civil War like crop lien work contracts, segregation, and unresolved problems with the seceded states. This period was called Reconstruction. After the Civil War, African Americans were free but with no place to live in or to work at, they settled with their former 'masters'. African Americans were technically free, but no one wanted to hire a colored man, so they were put on crop lien work contracts.\n\n passage 3:The War also, for the first time, brought economic unity to the US. The South joined the market system, abandoning their former semi-subsistence economy. Industrialization and large economic enterprises clearly had arrived to stay. Politically, important changes had accompanied the maintenance of the national unity.\n\n

Document 2 (Hypothesis): Based on the provided passages, here are the answers to the questions about the political, economic, and social effects of the United States:\n\n Question 1: What political effects have on the United States?\n Answer: The Reconstruction period after the Civil War brought significant political changes to the United States. The South joined the market system, and industrialization and large economic enterprises became established, leading to economic unity. Additionally, politically important changes accompanied the maintenance of national unity.\n \n Question 2: What economic effects have on the United States?\n Answer: The Reconstruction period after the Civil War brought economic unity to the United States, as the South joined the market system and abandoned their former semi-subsistence economy.\n \n Question 3: What social effects have on the United States?\n Answer: The passage does not provide information about specific social effects on the United States.\n \n Note: Based on the provided passages, the answers are limited to the time period after the Civil War, and do not cover the broader history of immigration or other time periods.

Query: what political, economic and social effects have on the united states

B.4 FactScore

This dataset has multiple queries per pair of passages, each with its own NLI label. Thus, here we show one pair of passages and a query for each label.

Document 1 (Premise): <s>Paul O'Neill (racing driver) Paul O'Neill (born 31 December 1979) is a British auto racing driver, and the half-brother of English singer Melanie C.</s><s>Career.</s><s>Career.:Early years. Born in Widnes, Cheshire, O'Neill was originally attracted to racing after seeing Nigel Mansell win the 1987 British Grand Prix, but did not pursue a career initially, studying for A-Levels but not completing them. After attending the Jim Russell racing school, he began racing in the MGF cup in 1999 and 2000, before entering the production class of the British Touring Car Championship in 2001. He did not win a race, but took 3 class podiums and 2 class poles en route to 8th in the class.</s><s>Career.:British Touring Car Championship.</s><s>Career.:British Touring Car Championship.:Vauxhall (2002–2003). His efforts in the production class were enough to earn him a drive with Team Egg Sport in their semi-works Touring Class Vauxhall Astra Coupe for 2002. He was 8th in this championship, and then stepped up to the factory Vauxhall team for 2003, finishing 4th in the series.</s><s>####SPECIAL####SEPARATOR####Career.:British Touring Car Championship.:Tech-Speed and Motorbase (2004–2007). He was not expected to retain the drive for 2004, but early that year he discovered that he had diabetes, and his racing licence was temporarily withdrawn. Once the condition was under control he did not immediately resume racing, instead helping his former team Tech-Speed prepare their bio-ethanol powered car, tutoring driver Fiona Leggate and fulfilling the post of race engineer. He also raced the guest car in the Porsche Carrera Cup at Oulton Park in 2006 along with some Ginettas Racing. Following Leggate's premature departure from Tech-Speed, O'Neill signed for the team to compete in the final two race weekends of the 2006 British Touring Car Championship season. This car is powered by sugar beet, which he is not permitted to eat because of his diabetes. He scored a 10th place in his first race back. Notably, while most drivers at this September Brands Hatch meeting were slower than they had been at the April meeting due to different track conditions, O'Neill was over a second faster than Leggate's time from that meeting in some practice sessions. At Silverstone he was hampered by engine issues in practice but took 2 top 10 finishes####SSPECIAL####SSEPARATOR####S on raceday. He did not keep the drive for 2007, Techspeed opting instead to unite with the Turkish Arkas team and run Erkut Kızılırmak. He competed in the final round of 2007 with Motorbase Performance in a SEAT Toledo.</s><s>Career.:British Touring Car Championship.:Tech-Speed (2009–2011). O'Neill returned to Tech-Speed for the 2009 British Touring Car Championship season, partnering Martyn Bell in a two-car team. He scored a third place at Snetterton, and as of round seven at Knockhill he had scored points in 11 successive races. He was placed 10th in the Drivers Championship. O'Neill remained with Tech-Speed for the 2010 & 2011 seasons, now joined by John George. He finished 9th (2010) and 10th (2011) claiming 5 podiums.</s><s>Career.:British Touring Car Championship.:Speedworks Motorsport (2012). He was without a regular drive at the start of the 2012 season, but drove the Speedworks Motorsport Toyota Avensis at Croft and Knockhill, deputising for the team's regular driver Tony Hughes, who was unable to take part due to his business commitments. He was unable####SSPECIAL####SSEPARATOR####S to secure the finances required to stay with the team for 2013.</s><s>Career.:British Touring Car Championship.:Tony Gilham Racing (2013). Having been without a drive since the start of the 2013 season, O'Neill joined Tony Gilham Racing for the Knockhill round driving an RCIB Insurance Racing Vauxhall Insignia when regular driver Jack Goff decided to miss the round.</s><s>Career.:British GT Championship. O'Neill entered the Croft round of the British GT Championship towards the end of 2007 alongside co-driver Steve Clark in a Team RPM Viper Competition Coupe GT3. He returned to the series in 2008 with Team RPM driving a Viper Competition Coupe in the GT3 class. He competed in the first eight races of the season, including a win in the second race at Rockingham. He finished the season 16th in the GT3 standings.</s><s>Career.:Other activities. O'Neill is currently a motor racing instructor at the Oulton Park circuit in Cheshire. Since 2012, he has been part of the presenting team for the ITV4 television coverage of the BTCC. In 2013 he appeared on the British television series Top Gear, driving a Volkswagen Beetle in a taxi race. In 2014, he appeared in####SSPECIAL####SSEPARATOR####S series 2 of Classic Car Rescue, driving and valuing the restored cars. He also competed in one round of the Mazda MX-5 SuperCup at Oulton Park, taking two wins. Paul O'Neill is a Patron of The Sporting Memories Foundation and on 10 September 2017 completed the Great North Run to raise funds for the charity. O'Neill is also a driver coach for Britcar class champion Tim Docker. O'Neill volunteered as a marshal at the seventh round of the Britcar Endurance Championship at Oulton Park. O'Neill was the driver of the car for a new world record set by golfer Marcus Armitage for the longest drive into a moving car. Paul is the co-founder of 29 Productions - a bespoke multi camera streaming and production company.</s>

Document 2 (Hypothesis): Paul O'Neill was a British racing driver born on December 30, 1979, in Widnes, England. He began his racing career in karting before moving up to car racing in the late 1990s. O'Neill competed in a variety of championships throughout his career, including the British Touring Car Championship (BTCC), the Porsche Carrera Cup, and the British GT Championship.\n \n O'Neill's most successful years came in the BTCC, where he competed for over a decade. He recorded his first win in the championship in 2002 and went on to achieve a further three victories. O'Neill was known for his aggressive driving style and his ability to push his car to its limits.Away from racing, O'Neill was also a popular figure in the media, serving as a commentator and pundit for the BTCC on ITV4. He was also a regular guest on BBC Radio 5 Live's motorsport coverage.\n \n Tragically, Paul O'Neill passed away on July 9, 2018, at the age of 38, following a battle with cancer. He was remembered by his fellow drivers and fans for his passion, dedication, and great personality.

Query 1: Where did O'Neill's most successful years occur?

Label 1: `not_entailment`

Query 2: In which racing event did O'Neill compete?

Label 2: `entailment`

C Annotator Instructions

The following displays the instructions given to annotators.

C.1 SNLI

```
    This dataset is very straightforward; each document is rather short and
     ↪  describes a scene that may be found in a picture.

Label set:
entailment: The answer to d1 with respect to the query entails the answer to d2 with
↪  respect to the query.
not\_entailment: All other cases. E.g. “neutral” or “contradiction”. You should
↪  treat this as a catch-all category.
```

C.2 RobustQA

```
This dataset targets the RAG retrieval setting: do two documents offer a
↪  contradictory perspective with respect to a query? The good news: all documents
↪  offer a “yes” or “no” perspective; so it should not take long to figure this out
↪  by skimming the documents. This is the only dataset where the label set uses
↪  “contradiction” rather than “entailment”.
```

```
Label set:
contradiction: The answer to d1 with respect to the query contradicts the answer to
↪ d2 with respect to the query. This typically means one document offers a “yes”
↪ answer and the other a “no”.
not_contradiction: All other cases.
```

C.3 RagTruth

```
This dataset targets the RAG hallucination setting: is d2 (the output of an LLM
↪ prompted on d1 and a query) entailed by d1 with respect to the query? This
↪ dataset is challenging to annotate because d2 contains many facts with respect
↪ to the query. In order for the label to be “entailment”, every fact relevant to
↪ the query in d2 should be corroborated by something in d1; otherwise it should
↪ be "not_entailment". You may find it easier to copy d1 into a text editor and
↪ Cmd+F to search for something specific.

Label set:
entailment: The answer to d1 with respect to the query entails the answer to d2 with
↪ respect to the query.
not_entailment: All other cases. E.g. “neutral” or “contradiction”. You should
↪ treat this as a catch-all category.
```

C.4 FactScore

```
All three FactScore datasets (ChatGPT, InstructGPT, and PerplexityAI) target the
↪ fact verification setting: is a particular fact in d2 (determined by the query)
↪ entailed by d1? This dataset is challenging to annotate because d1 (a Wikipedia
↪ bio) is very long. Here is the strategy I suggest:
    1. First find the fact referenced by the query within d2.
    2. Copy d1 into a text editor and search for keywords related to that fact (e.g.
    ↪ a name, a year, etc.). Determine your label based off that.
You should not need to read all of d1 in order to determine the label.

Label set:
entailment: The answer to d1 with respect to the query entails the answer to d2 with
↪ respect to the query.
not_entailment: All other cases. E.g. “neutral” or “contradiction”. You should
↪ treat this as a catch-all category.
```

D Dataset Adapation

In this section, we list the prompt templates we used to adapt each dataset into its QC-NLI variant.

D.1 SNLI

Prompt 1: Write paragraphs about premise and hypothesis

```
I will provide you with two sentences sentence1 and sentence2, where sentence1
↪ entails sentence2. I want you to do the following. Write paragraph1 and
↪ paragraph2 and a query, such that the answer to the query for paragraph1 is
↪ sentence1, and the answer to the query for paragraph2 is sentence2. The goal is
↪ to preserve the relationship between the answers (entailment). Adhere to the
↪ following guidelines:
    1. The paragraphs should consist of 4-10 sentences and should typically describe
    ↪ scenes that could be depicted in photographs.
    2. The paragraphs should not be written as if they are image captions. For
    ↪ example, they should not reference "a photo", "a scene", or "an image".
```

```
    3. Similarly, the query should be written as if it is about an image caption.
    ↪  Thus, it should avoid referring to the paragraph as "a photo", "a scene",
    ↪  or "an image".
    4. Both paragraphs must address the query you provide.
    5. Critically, each sentence should be the most specific answer possible to the
    ↪  query for its respective paragraph. For example, suppose sentence1 is "The
    ↪  children are talking to the animals", and the question you propose is "Who
    ↪  are the children talking to?" The paragraph should not say "The children
    ↪  are talking to giraffes", because "giraffes" are more specific than
    ↪  "animals". In that case, on the question of who the children are talking to,
    ↪  the paragraph should not be more specific than "animals".
    6. Since sentence1 entails sentence2, it is important that the answer to the
    ↪  query for paragraph1 entails the answer to the query for paragraph2.

Fill in the XML tags below. Note that answer1 and answer2 denote the answer to the
↪  query for paragraph1 and paragraph2 respectively.

<sentence1>{sent1}</sentence1>
<sentence2>{sent2}</sentence2>
<paragraph1></paragraph1>
<paragraph2></paragraph2>
<query></query>
<answer1>{sent1}</answer1>
<answer2>{sent2}</answer2>
```

Prompt 2: Do premise and hypothesis answer the query?

```
Pretend you are looking at a photograph depicting a scene, and someone asks you a
↪  question about it. I will provide you a sentence. I want you to tell me if the
↪  sentence would be a possible direct answer to the question for the hypothetical
↪  photograph. Since this is based on a photograph, you may not have all the
↪  information about the scene depicted; therefore, an answer may be plausible
↪  even if it is not extremely specific or detailed. The most important thing is
↪  that it provides a direct answer to the question, and if it does you should
↪  return "yes". Otherwise, return "no". If you are not sure, return "maybe".

Fill in the XML tags below (put your reasoning in the reasoning tags).

<question>{query}</question>
<sentence>{answer}</sentence>

Is the sentence a possible answer to the question?

<reasoning></reasoning>
<response></response>
```

Prompt 3: Is question about location?

```
I will provide you with a question about a hypothetical photograph. I simply want
↪  you to tell me if question is about the location, or setting, of the photo. If
↪  the question is about the location, then you should return "yes"; otherwise
↪  return "no". If you are unsure, return "maybe".
```

```
For example, if the question is ”Where are the children standing?” or ”What is the setting of the image?”, return ”yes”.
If the question is ”Who is talking with the woman?” or ”What is the man holding?”, return ”no”.

Fill in the XML tags below (put your reasoning in the reasoning tags).

<question>{query}</question>

Is the query asking about a location or setting?

<reasoning></reasoning>
<answer></answer>
```

Prompt 4: Modify location of paragraph (if question is not about location)

```
I will provide you with a paragraph about a hypothetical image. I simply want you to rewrite the paragraph, changing the location (or setting) to something different. You should otherwise not change the content of the paragraph.

For example, if the paragraph is set at the beach, you might change it to a pool. If it is set in a church, you might change it to a mosque.

Importantly, you must make sure that the answer to the question ”{query}” is still ”{sentence}”.

Fill in the XML tags below.

<paragraph>{paragraph}</paragraph>

Now change the setting of the paragraph, while making sure that the answer to the question ”{query}” is still ”{sentence}”.

<modified_paragraph></modified_paragraph>
```

Prompt 5: Do premise and hypothesis answer the query for the paragraphs?

```
I will provide you with a paragraph and a query. I simply want you to tell me if the paragraph contains an answer to the query. It may be that the answer to the query is complicated, spread over many sentences, or answerable in many ways. In any of these cases you should return ”yes”. If the query is based on a premise incompatible with the content of the paragraph, or if the query is irrelevant to the content of the paragraph, then you should return ”no”. If you are not sure, return ”maybe”.

Fill in the XML tags below (put your reasoning in the reasoning tags).

<paragraph>{paragraph}</sentence1>
<query>{query}</query>

Does the paragraph contain an answer to the query?

<reasoning></reasoning>
<answer></answer>
```

D.2 RobustQA

Prompt 1: Assign "yes" or "no" label to examples

```
I will provide you with a question, and an answer which either provides a positive
↪  ("yes") or negative ("no") opinion. I simply want you to tell me whether the
↪  opinion is "yes" or "no". A clear giveaway would be if the answer itself has
↪  these labels in it, especially if the answer starts with one of these labels.
↪  Fill in the reasoning and response XML tags below - the response must be "yes"
↪  or "no".

<question>{query}</question>
<answer>{answer}</answer>

<reasoning></reasoning>
<response></response>
```

D.3 RagTruth

Prompt 1: LLM generation (hypothesis) contains answer to query

```
I will provide you with a query and a response to the query written by an LLM. Note
↪  that the query is about a specific document, but you do not need to know the
↪  contents of that document to perform the following task. I just want you to
↪  tell me if the response contains an answer to the query. Because you do not have
↪  the document, you won't be able to tell if the answer is correct or not, but
↪  that is not relevant. You just need to tell me if the passage I provide you
↪  contains a potential answser to the query.

For example, if the query is "What is the store's name?", and the passage contains a
↪  sentence like "The store's name is Portle", then you should return "yes".
↪  Otherwise, return "no".

Fill in the XML tags for reasoning and answer below. Your answer must be "yes" or
↪  "no".

<query>{query}</query>
<response>{response}</response>

Does the response contain an answer to the query?

<reasoning></reasoning>
<answer></answer>
```

Prompt 2: Is annotated segment relevant to the query?

```
I will provide you with a query and a response to the query written by an LLM. Note
↪  that the query is about a specific document, but you do not need to know the
↪  contents of that document to perform the following task. I also will provide
↪  you with a snippet from the response. I just want you to tell me if snippet is a
↪  potential answer to the query; the rest of the response is simply provided as
↪  context. Because you do not have the underlying document, you won't be able to
↪  tell if the snippet is a correct correct or not, but that is not what you are
↪  being asked. You just need to tell me if the snippet is a potential answer to
↪  the query.
```

```
For example, if the query is "What is the name of the store in Mabileen?", and the snippet is "The store's name is Portle", then you should return "yes". If the snippet is something like "There is no store in Mabileen", you should return "yes". However, if the snippet is "Mabileen has 100 residents", you should return "no", since that is not an answer to the query.

Fill in the XML tags for reasoning and answer below. Your answer must be "yes" or "no".

<query>{query}</query>
<response>{response}</response>
<snippet>{snippet}</snippet>

Is the snippet a potential answer to the query?

<reasoning></reasoning>
<answer></answer>
```

D.4 FactScore

Prompt 1: Convert fact to question

```
I will provide you with an atomic fact. I want you to write a simple question such that it could be answered by the fact. For example, if the fact is "The sky is blue.", then the question would be "What color is the sky?" Fill in the XML tags below, including your reasoning.

<fact>{fact}</fact>

<reasoning></reasoning>

<question></question>
```

E QC-NLI Prediction

Zero-shot

```
I will provide you with two documents (document1 and document2) as well as a query about the documents. Treat document1 as the premise and document2 as the hypothesis. I want you to tell me how document1 relates to document2 with respect to the query. Specifically, you should return one of the following NLI labels: "entailment" or "not_entailment". You can assume that both documents contain the complete information needed to address the query.

The labels are defined as follows:
        - entailment: Return 'entailment' only if the answer to the query for document2 (the hypothesis) is necessarily true given the answer to the query for document1 (the premise).
        - not_entailment: Return 'not_entailment' only if the answer to the query for document2 (the hypothesis) does not entail the answer to the query for document1 (the premise).
```

```
<document1>{doc1}</document1>

<document2>{doc2}</document2>

<query>{query}</query>

Now, does document1 entail or not entail document2 with respect to the query? First,
↪  provide your reasoning in the reasoning tags. Then, provide your answer
↪  ("entailment" or "not_entailment") in the answer tags. Make sure your final
↪  answer is one of these answer choices.

<reasoning></reasoning>

<answer></answer>
```

Few-shot

```
I will provide you with two documents (document1 and document2) as well as a query
↪  about the documents. Treat document1 as the premise and document2 as the
↪  hypothesis. I want you to tell me how document1 relates to document2 with
↪  respect to the query. Specifically, you should return one of the following NLI
↪  labels: "entailment" or "not_entailment". You can assume that both documents
↪  contain the complete information needed to address the query.

The labels are defined as follows:
        - entailment: Return 'entailment' only if the answer to the query for
        ↪  document2 (the hypothesis) is necessarily true given the answer to the
        ↪  query for document1 (the premise).
        - not_entailment: Return 'not_entailment' only if the answer to the query
        ↪  for document2 (the hypothesis) does not entail the answer to the query
        ↪  for document1 (the premise).

I will provide you with a few examples:

<example_{ex1_label}>
<document1>{ex1_document1}</document1>

<document2>{ex1_document2}</document2>

<query>{ex1_query}</query>

Now, does document1 entail or not entail document2 with respect to the query? First,
↪  provide your reasoning in the reasoning tags. Then, provide your answer
↪  ("entailment" or "not_entailment") in the answer tags. Make sure your final
↪  answer is one of these answer choices.

<reasoning>{ex1_reasoning}</reasoning>

<answer>{ex1_label}</answer>
</example_{ex1_label}>

<example_{ex2_label}>
```

```
<document1>{ex2_document1}</document1>

<document2>{ex2_document2}</document2>

<query>{ex2_query}</query>

Now, does document1 entail or not entail document2 with respect to the query? First,
↪  provide your reasoning in the reasoning tags. Then, provide your answer
↪  ("entailment" or "not_entailment") in the answer tags. Make sure your final
↪  answer is one of these answer choices.

<reasoning>{ex2_reasoning}</reasoning>

<answer>{ex2_label}</answer>
</example_{ex2_label}>

Now it's your turn!

<document1>{doc1}</document1>

<document2>{doc2}</document2>

<query>{query}</query>

Now, does document1 entail or not entail document2 with respect to the query? First,
↪  provide your reasoning in the reasoning tags. Then, provide your answer
↪  ("entailment" or "not_entailment") in the answer tags. Make sure your final
↪  answer is one of these answer choices.

<reasoning></reasoning>

<answer></answer>
```

QA+NLI: Question-answering

```
You are a reliable question-answering system. I will provide you with a document
↪  and a question. I simply want you to provide an answer the question based on the
↪  document using one or more full sentences. It is very important that your
↪  answer be based only on the document: do not apply any knowledge beyond what is
↪  contained in the document. Be very concise: you do not need to provide more
↪  information than that which answers the question.

Fill in the resasoning and answer XML tags.

<document>{document}</document>
<question>{query}</question>

<reasoning></reasoning>
<answer></answer>
```

The answers to each question are then fed into the zero-shot prompt template.