GroundCocoa: A Benchmark for Evaluating Compositional & Conditional Reasoning in Language Models

Harsh Kohli
kohli.120@osu.edu

Sachin Kumar
kumar.1145@osu.edu

Huan Sun
sun.397@osu.edu

Abstract

The rapid progress of large language models (LLMs) has seen them excel and frequently surpass human performance on standard benchmarks. This has enabled many downstream applications, such as LLM agents, to rely on their reasoning to address complex task requirements. However, LLMs are known to unexpectedly falter in simple tasks and under seemingly straightforward circumstances - underscoring the need for better and more diverse evaluation setups to measure their true capabilities. To this end, we choose to study compositional and conditional reasoning, two aspects that are central to human cognition, and introduce GroundCocoa - a lexically diverse benchmark connecting these reasoning skills to the real-world problem of flight booking. Our task involves aligning detailed user preferences with available flight options presented in a multiple-choice format. Results indicate a significant disparity in performance among current state-of-the-art LLMs with even the best performing model, GPT-4 Turbo, not exceeding 67% accuracy despite advanced prompting techniques.

1 Introduction

Conditional and compositional reasoning are central to navigating and interacting with complex systems through decision-making processes (Oaksford and Chater, 2010; Simon and Newell, 1971). Conditional reasoning refers to the understanding and application of logical rules, often structured in “if-then” forms, which are fundamental to evaluating potential scenarios and anticipating outcomes in daily decision-making. Compositional reasoning involves solving complex problems by integrating solutions to simpler sub-problems in a structured manner. This cognitive process is crucial for understanding the relationships between different components of a task. We evaluate how effectively current LLMs exhibit these cognitive abilities, which are essential for both human and artificial intelligence. To that end, we introduce **GroundCocoa**[1], a benchmark designed to assess compositional and conditional reasoning within a grounding task.

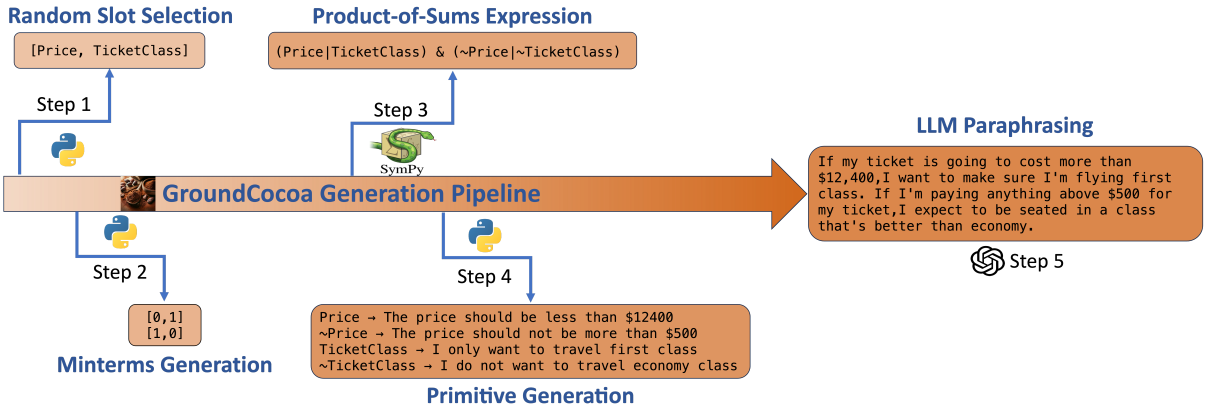
Set within a real-world inspired flight reservation scenario, GroundCocoa comprises questions framed as user needs. Finding and booking flights is a complex task where user requirements might be many and highly convoluted. While we use flight booking to illustrate our idea, the compositional primitives forming our user requirements test for general skills such as temporal reasoning (e.g., "I want a flight departing after 5 pm") and mathematical reasoning (e.g., "Ticket price should be under $1000"). Thus, we posit that results and insights derived from evaluating LLMs on GroundCocoa should largely be applicable to other domains.

We leverage a controllable method, illustrated in Figure 1, to create samples of varying complexity. Our data generation process (§2) consists of a 5-stage pipeline including online scraping, constraint generation, symbolic logic to impose conditionality, paraphrasing user requirements, and matching generated requirements to available flight options. To test for robustness, we allow requirements to freely condition on one another and impose no restrictions on their nature. Additionally, we isolate a subset of more atypical queries that contain unconventional user needs (e.g., “I want at least 2 layovers”) and evaluate their impact on model performance.

In addition to the release of the dataset and accompanying results, our contribution also includes the data generation pipeline which can be used to controllably generate samples of increasing complexity to challenge more advanced models in the future. Through slight modifications to the data scrapers and the primitive rule-set (described in Section 2), the method can also be extended to incorporate other domains for a more diverse eval-

[1] https://osu-nlp-group.github.io/GroundCocoa/

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 8280–8295
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics



```
If my ticket is going to cost more than
$12,400,I want to make sure I'm flying first
class. If I'm paying anything above $500 for
my ticket,I expect to be seated in a class
that's better than economy.
```

```
Price → The price should be less than $12400
~Price → The price should not be more than $500
TicketClass → I only want to travel first class
~TicketClass → I do not want to travel economy class
```

Figure 1: Stepwise depiction of GroundCocoa query generation using 2 slots and 2 minterms.

uation setup. Statistics of GroundCocoa are shown in Table 1. Our key findings are as follows:

1. Accuracy among contemporary LLMs varies greatly, ranging from a little better than random guess to about 67% on a five-option multiple-choice question task. Within this spectrum, GPT-4 Turbo (OpenAI, 2023) stands out, demonstrating a superior capacity of the GPT line of models to adapt and excel in novel reasoning tasks. However, conditional reasoning poses a significant challenge to all evaluated models, even on samples of relatively lower complexity.

2. Incorporating prompting techniques such as Chain of Thought (COT) (Wei et al., 2022) and Least-to-Most (L2M) prompting (Zhou et al., 2023) leads to mixed results, with only a modest performance improvement in some cases. Prior research has noted that although these methods help decompose problems into steps, LLMs struggle as the complexity of the individual steps grows (Hendrycks et al., 2021b; Madaan and Yazdanbakhsh, 2022; Nogueira et al., 2021; Qian et al., 2023). These assertions hold true in our observations.

3. Including unconventional user requirements leads to a drop in accuracy of as much as 6% in GPT-4 Turbo, indicating a training bias towards more typical needs.

2 Approach

Figure 1 illustrates our proposed approach for generating a user requirement. The task involves matching this generated requirement against 5 flight options where only 1 of the options satisfies the generated criteria. Our 5-stage data creation pipeline is detailed in subsequent sections. In the process of generating a natural language user requirement for flight booking, we are faced with the following considerations:

Conditionality of Constraints. We aim to challenge contemporary models in their ability to reason through scenarios characterized by conditional complexity. This is done through mutual dependence of flight attributes which we refer to as *slots*. As illustrated in the final requirement (Step 5) of Figure 1, there is an interdependence between the values for price and ticket class. This is a direct result of the generated minterm table. A minterm is a specific type of logical expression that represents exactly one row in a truth table where the function evaluates to true (1). In the context of GroundCocoa, each minterm represents a specific combination of flight attributes, or 'slots', that satisfies a particular user requirement. We represent this interdependence in logical form through a Product-of-Sums (POS) expression which consists of multiple OR operations (sums) which are later combined through AND operations (products). This process is further explained in a subsequent section (§2.2). The inclusion of OR operations between slots introduces conditional complexity to our user requirement, necessitating consideration of potential slot values in if-then scenarios. On the other hand, a greater number of AND conditions implies a higher number of variables that a model has to simultaneously reason over resulting in increased compositional complexity.

Satisfiability of POS Expression. While generating the logical form for a user requirement, we must ensure satisfiability of the generated POS expression. For this, we use SymPy (Meurer et al., 2017), an open-source Python symbolic mathematics library which generates an optimal POS expression given a minterm table (§2.2).

Fuzziness in Slot Values. Corresponding to each occurrence of a slot in the POS expression there has to be a unique constraint. For the example in Figure 1, the two constraints on the price slot are {<12400, <500}. We impose these constraints randomly through specialized rule-based systems corresponding to each slot. However, these might cause the final user criteria to become impossible to satisfy even if the corresponding POS expression is satisfiable. Thus, for a generated user requirement we perform checks to ensure that there exists at least 1 route that satisfies the criteria and at least 4 that do not so that there are at least 1 positive and 4 negative options for a generated requirement.

In addition to the test set, we also include a separate validation set which may be used for tuning hyperparameters. The pipeline may be reused to generate more complex samples in the future, and could also be extended to other domains through a slight modification of the data collection (§2.1) and primitive generation (§2.3) stages.

2.1 Flight Data Collection

We start with a list of the top 50 busiest airports by passenger traffic derived from Wikipedia. We choose source and destination airports randomly from this list. A fixed departure date is also chosen randomly from the future and set for each flight search. The source, destination, and travel date are input to Google Flights. We then sample a small number of flights from the search results. The sampled flights are chosen from each of economy, business, and first class and, for each flight option, all the relevant details such as the number of layovers, price, departure and arrival times etc. are saved. A sample flight schema with all the elements is provided in Appendix A. We use Selenium Webdriver for scraping this data.

2.2 Product-of-Sums Generation

To generate a POS expression, we first randomly select a small number of flight attributes or *slots*. The complete set of slots S is as follows:

S={airline, ticket class, departure time, arrival time, total travel time, number of layovers, average carbon emission difference, travel date, price, layover locations, layover times}

We vary the number of slots between 2 and 6 in order to generate samples of differing complexity. We then randomly generate 2-3 “minterms”, the list of all input combinations of slots that generate a true (1). A higher number of minterms results in a greater conditional complexity. The slot symbols and generated minterms are input to SymPy which uses a redundant-group eliminating algorithm to output the smallest POS expression consistent with the minterm table.

2.3 Primitive Generation

Corresponding to each slot, we have developed a rule-based system that randomly imposes constraints on its values. These constraints are converted to natural language through templates. Since a POS expression may contain a negation, we generate two primitives at each turn - one for the constraint and one for its negation. A sample primitive for total travel time is shown in Table 2.

TravelTime	Travel Time should be more than 22 hours and 30 minutes.
¬TravelTime	Travel Time should not be more than 22 hours and 30 minutes.

Table 2: Sample primitive for total travel time.

At this stage, we also isolate samples that include any one of the following three primitives - (1) carbon emissions must be above the average for that route, (2) price of the flight must be above a minimum threshold, and (3) number of layovers on the route should be greater than a minimum. While this list is not exhaustive, such samples (henceforth referred to as "atypical" queries) are able to successfully encapsulate contrarian needs that are unlikely to manifest often during pretraining.

2.4 LLM Paraphrasing and Human Validation

We paraphrase the user requirement derived from rule templates to make them more natural-sounding while preserving the original intent and meaning. LLM paraphrasing is carried out in two distinct steps described below. The exact prompts and an example of intermediate results are provided in Appendix D. We manually verify each query to ensure it is consistent with the primitives and make changes wherever necessary.

1. Individual primitives are substituted into each sum term and combined using templated rules. We then use GPT-4 Turbo to paraphrase each of the sum terms.

2. Next, we combine the individual sum terms into a product (logical AND). This is done by

Statistic	(slot, minterm) configurations						**Total**
	(2,2)	(3,2)	(4,2)	(4, 3)	(5,2)	(6,2)	
Test Samples	1511	1083	710	723	451	371	**4849**
Test Unique Queries	124	136	117	129	121	101	**728**
Val. Samples	17	17	8	5	2	3	**52**
Val. Unique Queries	1	1	1	1	1	1	**6**
Avg. Query Length	65.04	88.33	103.88	119.14	124.56	148.87	**95.95**
Avg. Context Length	-	-	-	-	-	-	**1252.27**
Vocab Size	-	-	-	-	-	-	**4200**

Table 1: Key Statistics of GroundCocoa.

merging the paraphrases of sum terms, separated by periods. The resulting flight requirement is again paraphrased with GPT-4 Turbo.

2.5 Option Matching

We match the generated user requirements with the flight data collected in Section 2.1. Each route between the source and destination represents a potential choice in our multiple-choice dataset. Choices are divided into subsets containing one positive (matching the user requirement) and four negative (not matching the user requirement) options. This is done to ensure that each multiple-choice question has only a single correct answer for ease of evaluation. Many such subsets may be created from a single user requirement and, consequently, our dataset consists of queries repeated multiple times with differing choices. Table 1 contains details of the number of unique queries and overall samples corresponding to all the slot/minterm configurations in GroundCocoa.

3 Results

To measure performance on GroundCocoa, we test several models of different sizes including both open-source and closed-source LLMs - LLAMA 2-chat (Touvron et al., 2023) / LLAMA 3-Instruct (Dubey et al., 2024), Mixtral 8x7B - Instruct (Jiang et al., 2024) / Mistral 7B Instruct (Jiang et al., 2023), Gemini Pro (Team et al., 2023), and GPT-4 Turbo. Results from our experiments are shown in Table 3. We have 3 different evaluation setups for the our models - direct prompting, chain-of-thought (CoT) (Wei et al., 2022) prompting, and least-to-most (L2M) prompting (Zhou et al., 2023). We aim to evaluate the intrinsic reasoning ability of current LLMs and, thus, exclude methods such as Program of Thoughts (Chen et al., 2023) and Program-aided Language Models (Gao et al., 2023) that offload the critical reasoning component to an external engine such as a Python interpreter.

3.1 Direct Prompting

The models are presented with a sample from GroundCocoa consisting of a user requirement and 5 flight options in a zero-shot manner. The smaller models in our experiments are only evaluated in this setting.

3.2 Chain-of-Thought Prompting

Since our task involves grounding user requirements to each answer choice, the CoT explanations are provided for each flight option given the user requirement. Thus, our standard CoT (CoT-full) consists of 5 distinct explanations. On GPT-4, we empirically observe that the large resulting context length can prove detrimental to model performance with the models often confusing between the requirements and options of the test case and the exemplar. To address this, we try a different prompting strategy (CoT-partial) with only two flight choices (1 positive and 1 negative) for the in-context example. Due to limitations on context length (4096 tokens) we are unable to run LLAMA 2-chat 70B on CoT-full. The exact prompts are given in Appendix C. Results from our experiments are shown in Table 3. As alluded to previously, GroundCocoa poses a significant challenge for each of the evaluated models, even with CoT prompting. The CoT-partial strategy leads to better results than CoT-full in 3 out of 4 cases, and best results are obtained using GPT-4 Turbo with CoT-partial. It is noteworthy, though, that there exists a marked difference in performance between competing models. Such variation represents a significant departure from the usual performance patterns observed in popular benchmarks such as MMLU (Hendrycks et al., 2021a), HellaSwag (Zellers et al., 2019), ARC Reasoning Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2021), and

GSM-8K (Cobbe et al., 2021) among others, where results are much more comparable.

3.3 Least-to-Most Prompting

Finally, we do a limited evaluation with least-to-most prompting which carries out task decomposition through an iterative prompting procedure. The problem (user requirement) is broken down into multiple sub-problems and each sub-problem is solved iteratively through successive prompts. The number of decomposition steps (turns) required in L2M scales linearly with the compositional complexity of each sample. The large number of turns per sample leads to a higher inference cost. We thus test each of our larger models using L2M using a subset of 200 samples from GroundCocoa - the corresponding rows are marked with an asterisk (*) in Table 3. Results indicate that GroundCocoa remains a challenging benchmark despite such multi-turn prompting methods for problem decomposition.

	Regular	Atypical	Total
Open-source Models			
LLAMA 2-chat 7B	14.56	14.66	14.60
LLAMA 3.1-chat 8B	33.66	35.25	34.29
Mistral 7B Instruct	25.70	26.10	25.86
LLAMA 2-chat 13B	16.33	16.06	16.23
Mixtral 8x7B-Instruct	45.79	42.48	44.48
Mixtral 8x7B-Instruct + CoT-full	34.38	32.65	33.69
Mixtral 8x7B-Instruct + CoT-partial	41.38	39.85	40.15
Mixtral 8x7B-Instruct + L2M*	22.32	15.90	19.50
LLAMA 2-chat 70B	24.13	21.63	23.13
LLAMA 2-chat 70B + CoT-partial	25.73	23.97	25.03
LLAMA 3.1-chat 70B	59.57	55.64	58.01
LLAMA 3.1-chat 70B + CoT-full	58.37	57.67	58.09
LLAMA 3.1-chat 70B + CoT-partial	60.22	58.66	59.60
LLAMA 3.1-chat 70B + L2M*	68.18	50.89	58.50
Closed-source Models			
Gemini Pro	42.79	40.46	41.86
Gemini Pro + CoT-full	41.14	40.87	41.04
Gemini Pro + CoT-partial	34.82	33.85	34.44
Gemini Pro + L2M*	42.86	40.46	41.90
GPT-4 Turbo	64.66	58.81	62.34
GPT-4 Turbo + CoT-full	65.07	61.51	63.66
GPT-4 Turbo + CoT-partial	67.77	65.62	66.92
GPT-4 Turbo + L2M*	46.43	53.41	49.50

Table 3: Accuracy (%) on GroundCocoa.

4 Analysis

Beyond assessing the overall model performance, we also investigate the consequences of varying the complexity of user criteria and presenting relatively unconventional user needs.

4.1 Impact of Increasing Complexity

In our analysis, we observe the performance of GPT-4 Turbo, the best-performing model from among those tested on GroundCocoa across different levels of conditional and compositional complexity. In their recent work on assessing the limitations of transformer on compositional tasks, Dziri et al. (2023) use computational graphs as approximations of the underlying reasoning processes in such models. They define the terms *reasoning depth*, the length of the deepest layer in the computational graph from the source nodes, and *reasoning width*, the mode of number of nodes in each layer - indicating the extent of multi-hop reasoning and compositional parallelism required to solve a given problem. Considering the characteristics of GroundCocoa we focus on reasoning width - the number of variables a model has to simultaneously reason over for a given problem. Intuitively, this may be represented by the number of *slots* used during the generation of a particular sample as described in Section 2.2. However, keeping the number of rows in the minterm table constant while increasing the slots may often lead to lower conditional complexity as the number of slots is increased.

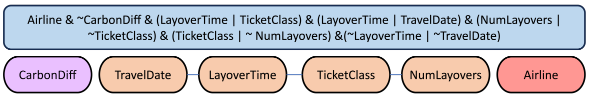

Airline & ~CarbonDiff & (LayoverTime | TicketClass) & (LayoverTime | TravelDate) & (NumLayovers | ~TicketClass) & (TicketClass | ~ NumLayovers) &(~LayoverTime | ~TravelDate)
CarbonDiff
TravelDate
LayoverTime
TicketClass
NumLayovers
Airline

Figure 2: POS expression and its dependency graph.

In order to effectively gauge the compositional and conditional complexity of a sample in our dataset, we define a dependency graph derived from the POS expression corresponding to that sample. Vertices represent slots and a dependency (edge) is created when a particular slot co-occurs with another slot within a sum term in the POS. A sample POS expression and its corresponding dependency graph are shown in Figure 2. The graph has 3 connected components with the largest connected component (LCC) of size 4. The maximum degree is 2 which corresponds to the two connections for nodes LayoverTime and TicketClass.

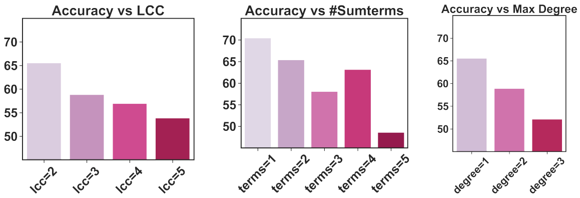


Figure 3: Increasing complexity in evaluation samples.

Given a fixed schema for the flight options, the

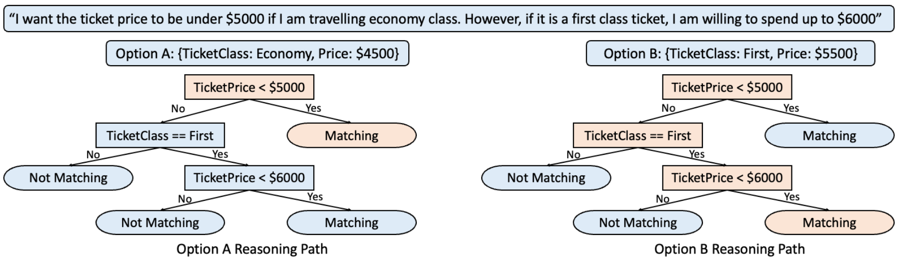


Figure 4: Sample user requirement and two hypothetical flight options.

number of sum terms in the POS expression as well as the LCC in the dependency graph are indicative of the *reasoning width* and, in turn, the compositional complexity of the user criteria. The LCC is the length of the largest chain of slots - the possible values of which are dependent on one another through OR conditions (represented by edges in the dependency graph). This metric effectively reflects the breadth of parallel computation or *reasoning width* required to accurately infer the given user criteria. Since increased branching in the dependency graph suggests a greater conditional complexity in user criteria, we also analyze model performance with increasing *maximum degree* of the dependency graph. This gives us the extent of conditioning on a single slot value. In Figure 3 we observe the decline in model performance with increased complexity as indicated by these factors.

4.2 Quantifying Confusion in Answer Choices through Entropy

Numerous recent studies have explored how deep learning models, specifically transformer-based architectures, achieve success by exploiting shortcuts (Geirhos et al., 2020; Liu et al., 2022; Tang et al., 2023; Du et al., 2023) and relying on spurious correlations present in the training data (Zhang et al., 2023; Saparov and He, 2023; Saparov et al., 2023). Recently, Dziri et al. (2023) utilized relative information gain of individual output elements in partially correct answers to explain surface pattern understanding in LLMs. In the same vein, we employ entropy as a metric to measure the confusion that might be caused due to conditions in the user query for a given flight option. We do this in an attempt to demystify how language models may succeed at some and fail at other queries with similar levels of complexity. To illustrate this, we take an example user requirement, and two hypothetical and simplified flight options as shown in Figure 4. Additionally, we show the reasoning path that must be navigated in each case for a successful outcome.

	Option A	Option B
Price < $5000	1	0
TicketClass = Economy	1	0
Price < $6000	1	1
TicketClass = First	0	1
p_{sat}	0.75	0.5
$p_{\bar{sat}}$	0.25	0.5
Entropy	0.81125	1.0

Table 4: Satisfaction of primitives and entropy.

We observe how option B in our example leads to a more convoluted reasoning path, whereas the model is able to bypass considerable conditional overhead in the case of Option A. For the purpose of quantifying this more generally, we observe the compositional primitives (values attached to individual slots in the POS expression) in each sample and attach a binary value indicating if the primitive is satisfied. For the example in Figure 4, we show the primitives and the corresponding values of both options in Table 4. We also show the probability of a primitive being satisfied(p_{sat}) and being unsatisfied($p_{\bar{sat}}$) by the flight option under consideration, as well as the final entropy.

Entropy due to user criteria for each option can then be computed using the formula in Equation 1. Higher uncertainty leads to greater entropy in Option B as opposed to Option A, indicating a greater conditional overhead.

$$H(X) = -(p_{sat} log p_{sat} + p_{\bar{sat}} log p_{\bar{sat}}) \quad (1)$$

In our analysis, we take the entropy values of the correct answer choice for each sample. Figure 5 shows the densities of entropy values for the correct and wrong predictions of GPT-4 Turbo. While correct predictions exceed wrong predictions at lower entropy values, an abrupt surge in wrong predictions is observed at higher entropy levels. Thus, entropy gives us yet another measure of conditional complexity from the perspective of the answer choices rather than just the query, and helps explain why a model might exhibit inconsistent results across user queries of similar complexity.

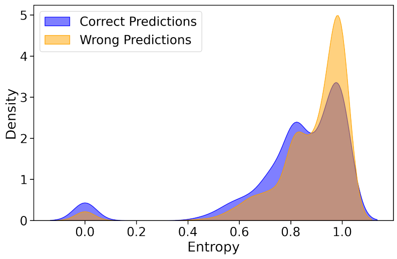


Figure 5: Effect of increasing entropy in answer choices.

4.3 Robustness to Unconventional User Needs

Several contemporary studies have sought to examine the robustness of language models by studying their resilience to out-of-distribution data (Koh et al., 2021; Wang et al., 2023) or through adversarial attacks and input perturbations (Gardner et al., 2020; Goel et al., 2021; Subhash et al., 2023; Sanyal et al., 2022; Yuan et al., 2023). In our work, we challenge models through atypical user requirements in order to assess bias from pretraining and robustness to unorthodox and nontraditional queries. We segregate queries into "Regular" and "Atypical" groups as described in Section 2.3. In Table 3, we contrast model performance on samples that describe such unconventional user needs versus those that do not. While most models in our testing show a decay in performance, the impact is more noticeable on better performing models such as GPT-4 Turbo. The in-context example used for all queries when testing with CoT includes two such primitives (ticket price $>$ 1800, carbon emission above average). We observe that the decline in performance is less pronounced with CoT.

5 Related Work

Reasoning Challenges in NLP. Our work extends the existing line of research on evaluating natural language processing (NLP) systems on different facets of reasoning - most notably commonsense question-answering (Talmor et al., 2019; Huang et al., 2019), physical reasoning (Bisk et al., 2020), social interaction (Sap et al., 2019), mathematical reasoning (Cobbe et al., 2021; Amini et al., 2019; Miao et al., 2020; Hendrycks et al., 2021b), story completion (Zellers et al., 2019), temporal reasoning (Zhou et al., 2019; Tan et al., 2023) abductive reasoning (Bhagavatula et al., 2020) and pronoun resolution (Sakaguchi et al., 2021). Different from these benchmarks, GroundCocoa introduces a unique and substantial challenge for LLMs in the form of conditional and compositional reasoning.

Among these, ConditionalQA (Sun et al., 2022) is arguably the most comparable to GroundCocoa in terms of the skills it assesses. While ConditionalQA focuses on the reading comprehension of conditionally-complex policy documents, GroundCocoa further tests the alignment/grounding ability of language models as (conditionally complex) user preferences have to be matched with multiple (5) flight schemas. The user requirements in GroundCocoa are deliberately generated to introduce conditional complexity through our pipeline. These factors result in a greater number of reasoning paths (reasoning width) and the number of variables the model has to simultaneously consider when answering a question. Our method provides a controllable way to adjust for this complexity by a simple adjustment of parameters such as slots/minterms as described in Section 2.2. Thus, GroundCocoa can be scaled to more complex examples in the future and also adapted to different domains.

Benchmarks on Propositional Logic. GroundCocoa also aligns with the considerable body of work on evaluating logical reasoning in language models. The RuleTaker (Clark et al., 2021) and ProofWriter (Tafjord et al., 2021) datasets proposed a modern approach to evaluating logical reasoning through a task involving assignment of binary labels to candidate implications following a set of premises expressed in natural language. The datasets emulate a *linear* deductive chain of reasoning of varying depths given a set of facts and rules, with ProofWriter augmenting this task through intermediate conclusions and proof generation. LogicNLI (Tian et al., 2021) provides a more comprehensive diagnostic benchmark involving reasoning through all seven fundamental logics (conjunction, disjunction, negation, implication, equation, universal and existential quantifiers). It contains an additional "paradox" label implying a situation where both the hypothesis as well as its negative proposition can be simultaneously entailed to the premise through different reasoning paths. This facilitates a non-linear reasoning, but is still limited to two contradictory reasoning paths. The FOLIO (Han et al., 2022) dataset boasts a higher vocabulary size due to a hybrid annotation approach but again consists of linear reasoning chains. Along similar lines, ProntoQA (Saparov and He, 2022) proposes a first-order logic benchmark using a linear ontology which might be fictional. This is done to prevent LLMs from predicting correct outcomes through spurious correlations in their pretraining corpus.

The benchmarks described here are primarily focused on the evaluation of deductive reasoning. In contrast, GroundCocoa offers a more realistic grounding task with an emphasis on if-then reasoning which leads to many candidate reasoning paths for each answer choice. While deductive reasoning may involve a broader range of logical structures, conditional reasoning is a subset which deals specifically with the relationships and implications of conditional statements. Our dataset consists of a large vocabulary size and context length per sample, leading to greater linguistic diversity, and a higher reasoning width than other benchmarks in logical reasoning. Questions are designed to test for robustness against rare and unconventional user requirements and bring to the fore model bias from pretraining data. Also, unlike most other benchmarks, we do not attempt to evaluate logical reasoning in isolation - our task might require abilities such as temporal or mathematical reasoning.

Compositional Generalization. Samples in GroundCocoa consist of novel combination of primitives expressed as user requirements in a flight-booking task. Such reasoning falls under the umbrella of compositional generalization - an area that has garnered increasing interest recently. Hosseini et al. (2022) highlight the relative generalization gap with in-context learning between in-distribution and out-of-distribution samples in various semantic parsing tasks. Dziri et al. (2023) demonstrates how transformer-based LLMs may solve compositional tasks by reducing them to linearized subgraph matching. By establishing a computational graph for each problem, the authors are able to define computational complexity by metrics such as the reasoning depth and width which correspond to levels in multi-hop reasoning and average parallelism respectively. Unsurprisingly, increased task complexity leads to a rapid decay in model performance under various settings.

Our findings largely concur with previous literature on compositional reasoning. However, results on GroundCocoa reveal that even the most advanced LLMs struggle at relatively low levels of compositional complexity when juxtaposed with conditional reasoning and grounding. While Dziri et al. (2023) demonstrated their results using problems such as multi-digit multiplication, dynamic programming, and Einstein's puzzle - we release a new dataset that is anchored on a practical, real world use-case of parsing complex user criteria and grounding to a fixed schema representing a flight option. GroundCocoa contains a high semantic coverage and we posit that it would be of interest to the NLP community as a hard evaluation set to benchmark compositional generalization in LLMs.

Dialogue-State Tracking. Finally, while our task is reminiscent of a single turn in a dialogue state tracking system, it goes one step further to test a language model's grounding ability to match a flight schema with the user query. Most schema-guided dialogue datasets (Rastogi et al., 2020; Lee et al., 2022) consist of fixed slot values and filtering of available options is handled through external systems (e.g. api's). Slot values in GroundCocoa are fuzzy due to conditional constraints on the primitives - in Figure 4, TicketPrice may take on different values based on TicketClass. GroundCocoa consists of examples with varying levels of compositional complexity due to long and complex user requirements. This differentiates it from the majority of schema-guided dialogue datasets where the primary objective is goal identification and tagging of slot values. These tasks, while challenging in their own respect, do not engage a models' compositional reasoning ability to the same extent.

6 Conclusion

Modern LLMs have demonstrated remarkable advancements in many tasks including those that are inherently compositional and necessitate conditional reasoning such as mathematical problem solving, and code generation and interpretation. However, discerning genuine reasoning from mere rote learning and shallow understanding continues to be a focal point of study. Though proficient at answering questions of seemingly greater complexity, we show that they can struggle on the same skills when presented with an unfamiliar task setting. While problem size does have an impact, even the less complex samples in our dataset are challenging to the best language models today.

Beyond introducing a new benchmark dataset, we conduct a thorough analysis of the effects of increasing complexity, including advanced prompting techniques, and robustness to atypical queries.

Our results uncover a substantial disparity in the performance of competing language models, a distinction that is not as pronounced in most other evaluation benchmarks and highlights their respective abilities in tackling novel challenges. Our data generation process is largely automatic, with human validation at the last step. In addition to the dataset and the evaluation script, we release code for the data generation which can be easily extended to generate more examples, and increase diversity (through different slots) as well as complexity. With minor modifications, the task can be further complicated by incorporating queries with multiple answers and questions that require other forms of logical reasoning such as aggregation (e.g., "Give me the cheapest flight matching my criteria?") and existential quantification (e.g., "Is there a first class seat under $5000?"), greater world knowledge (e.g., "I'd like to avoid layovers in Europe") etc., which we leave for future work.

7 Limitations

GroundCocoa consists entirely of samples in the flight-booking domain. This scenario is popular and widely used in training and evaluation benchmarks for dialogue state tracking, planning etc. Due to the general nature of the primitives used in our flight requirements, we are confident that the results and insights would be applicable to a wide array of domains. However, this has not been empirically validated and we leave the extension of GroundCocoa to other domains as a topic for future research.

To isolate unconventional user requirements, we identify primitives that are uncommon in typical flight reservation scenarios (e.g., "I want more than two layovers"). However, the criteria for segregation involves a degree of subjectivity. Furthermore, conventional primitives can be combined in unconventional ways using conditional formats (e.g., "If the flight is after 7 pm, I want the carbon emissions to be below average"), which our approach for identifying unconventional requirements does not account for. Consequently, further investigation is needed to evaluate model robustness to unconventional requirements that significantly deviate from patterns likely encountered in training data.

Finally, we assess the performance of LLMs using both CoT and L2M prompting techniques. However, L2M requires several decomposition steps, resulting in multiple prompts to the various LLMs for each test sample. Given the high inference cost associated with this approach, our evaluation is limited to a subset of 200 samples. While the results suggest that GroundCocoa remains a challenging benchmark even with L2M prompting, they do not offer a full assessment of individual LLM performance under this setting.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Chandra Bhagavatula, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Wen tau Yih, and Yejin Choi. 2020. Abductive commonsense reasoning. In *International Conference on Learning Representations*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Peter Clark, Oyvind Tafjord, and Kyle Richardson. 2021. Transformers as soft reasoners over language. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3882–3890.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Commun. ACM*, 67(1):110–120.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: program-aided language models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Matt Gardner, Yoav Artzi, Victoria Basmov, Jonathan Berant, Ben Bogin, Sihao Chen, Pradeep Dasigi, Dheeru Dua, Yanai Elazar, Ananth Gottumukkala, et al. 2020. Evaluating models' local decision boundaries via contrast sets. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1307–1323.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

Karan Goel, Nazneen Fatema Rajani, Jesse Vig, Zachary Taschdjian, Mohit Bansal, and Christopher Ré. 2021. Robustness gym: Unifying the nlp evaluation landscape. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*, pages 42–55.

Simeng Han, Hailey Schoelkopf, Yilun Zhao, Zhenting Qi, Martin Riddell, Luke Benson, Lucy Sun, Ekaterina Zubova, Yujie Qiao, Matthew Burtell, David Peng, Jonathan Fan, Yixin Liu, Brian Wong, Malcolm Sailor, Ansong Ni, Linyong Nan, Jungo Kasai, Tao Yu, Rui Zhang, Shafiq Joty, Alexander R. Fabbri, Wojciech Kryscinski, Xi Victoria Lin, Caiming Xiong, and Dragomir Radev. 2022. Folio: Natural language reasoning with first-order logic. *arXiv preprint arXiv:2209.00840*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Arian Hosseini, Ankit Vani, Dzmitry Bahdanau, Alessandro Sordoni, and Aaron Courville. 2022. On the compositional generalization gap of in-context learning. In *Proceedings of the Fifth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 272–280, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. 2021. Wilds: A benchmark of in-the-wild distribution shifts. In *International Conference on Machine Learning*, pages 5637–5664. PMLR.

Harrison Lee, Raghav Gupta, Abhinav Rastogi, Yuan Cao, Bin Zhang, and Yonghui Wu. 2022. Sgd-x: A benchmark for robust generalization in schema-guided dialogue systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10938–10946.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2022. Transformers learn shortcuts to automata. In *The Eleventh International Conference on Learning Representations*.

Aman Madaan and Amir Yazdanbakhsh. 2022. Text and patterns: For effective chain of thought, it takes two to tango. *arXiv preprint arXiv:2209.07686*.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, Amit Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. 2017. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, and Jimmy Lin. 2021. Investigating the limitations of transformers with simple arithmetic tasks. *arXiv preprint arXiv:2102.13019*.

Mike Oaksford and Nick Chater. 2010. Cognition and conditionals: Probability and logic in human thought.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Jing Qian, Hong Wang, Zekun Li, Shiyang Li, and Xifeng Yan. 2023. Limitations of language models in arithmetic and symbolic induction. pages 9285–9298.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Soumya Sanyal, Zeyi Liao, and Xiang Ren. 2022. RobustLR: A diagnostic benchmark for evaluating logical robustness of deductive reasoners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9614–9631, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Abulhair Saparov and He He. 2023. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*.

Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Seyed Mehran Kazemi, Najoung Kim, and He He. 2023. Testing the general deductive reasoning capacity of large language models using OOD examples. *CoRR*, abs/2305.15269.

Herbert A. Simon and Allen Newell. 1971. Human problem solving: The state of the theory in 1970. *American Psychologist*, 26:145–159.

Varshini Subhash, Anna Bialas, Weiwei Pan, and Finale Doshi-Velez. 2023. Why do universal adversarial attacks work on large language models?: Geometry might be the answer. In *The Second Workshop on New Frontiers in Adversarial Machine Learning*.

Haitian Sun, William Cohen, and Ruslan Salakhutdinov. 2022. ConditionalQA: A complex reading comprehension dataset with conditional answers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3627–3637, Dublin, Ireland. Association for Computational Linguistics.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Ruixiang Tang, Dehan Kong, Longtao Huang, and Hui Xue. 2023. Large language models can be lazy learners: Analyze shortcuts in in-context learning. pages 4645–4657.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Jidong Tian, Yitian Li, Wenqing Chen, Liqiang Xiao, Hao He, and Yaohui Jin. 2021. Diagnosing the first-order logical reasoning ability through LogicNLI. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3738–3747, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jindong Wang, HU Xixu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, et al. 2023. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zhangdie Yuan, Songbo Hu, Ivan Vulić, Anna Korhonen, and Zaiqiao Meng. 2023. Can pretrained language models (yet) reason deductively? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1439–1454.

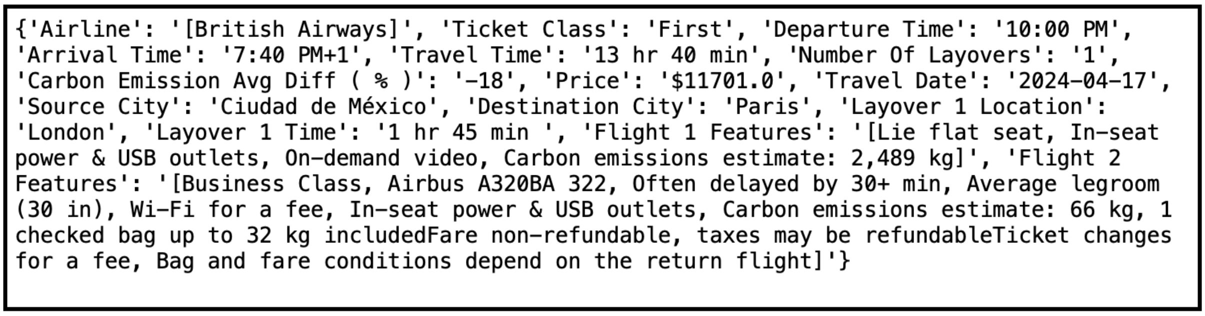
Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800.

Honghua Zhang, Liunian Harold Li, Tao Meng, Kai-Wei Chang, and Guy Van den Broeck. 2023. On the paradox of learning to reason from data. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI)*.

Ben Zhou, Daniel Khashabi, Qiang Ning, and Dan Roth. 2019. “going on a vacation” takes longer than “going for a walk”: A study of temporal commonsense understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

A Sample Flight Schema

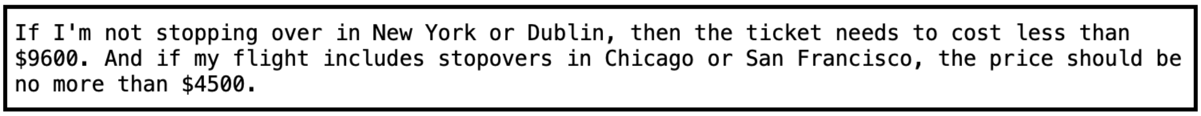


```
{'Airline': '[British Airways]', 'Ticket Class': 'First', 'Departure Time': '10:00 PM',
'Arrival Time': '7:40 PM+1', 'Travel Time': '13 hr 40 min', 'Number Of Layovers': '1',
'Carbon Emission Avg Diff ( % )': '-18', 'Price': '$11701.0', 'Travel Date': '2024-04-17',
'Source City': 'Ciudad de México', 'Destination City': 'Paris', 'Layover 1 Location':
'London', 'Layover 1 Time': '1 hr 45 min ', 'Flight 1 Features': '[Lie flat seat, In-seat
power & USB outlets, On-demand video, Carbon emissions estimate: 2,489 kg]', 'Flight 2
Features': '[Business Class, Airbus A320BA 322, Often delayed by 30+ min, Average legroom
(30 in), Wi-Fi for a fee, In-seat power & USB outlets, Carbon emissions estimate: 66 kg, 1
checked bag up to 32 kg includedFare non-refundable, taxes may be refundableTicket changes
for a fee, Bag and fare conditions depend on the return flight]'}
```

Figure 6: Schema for British Airways flight between Mexico City and Paris on 04/24/2024

B Samples from different slot/minterm configurations

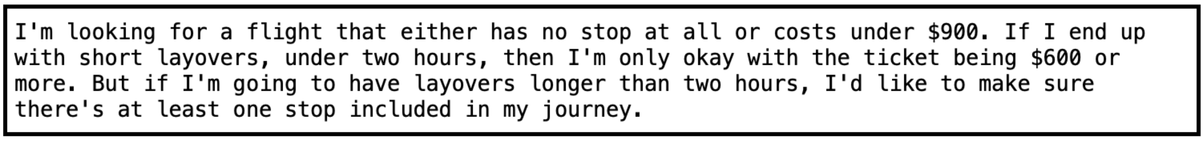
B.1 2 slots, 2 minterms



If I'm not stopping over in New York or Dublin, then the ticket needs to cost less than $9600. And if my flight includes stopovers in Chicago or San Francisco, the price should be no more than $4500.

Figure 7: Sample query in GroundCocoa with 2 slots and 2 minterms.

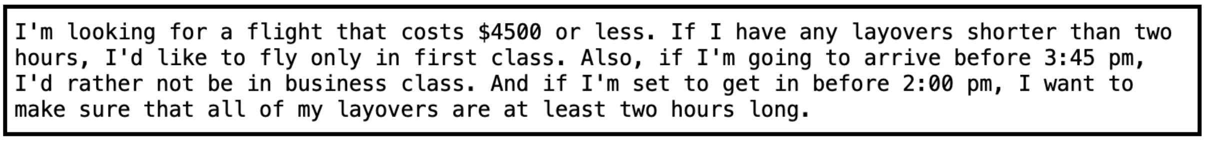
B.2 3 slots, 2 minterms



I'm looking for a flight that either has no stop at all or costs under $900. If I end up with short layovers, under two hours, then I'm only okay with the ticket being $600 or more. But if I'm going to have layovers longer than two hours, I'd like to make sure there's at least one stop included in my journey.

Figure 8: Sample query in GroundCocoa with 3 slots and 2 minterms.

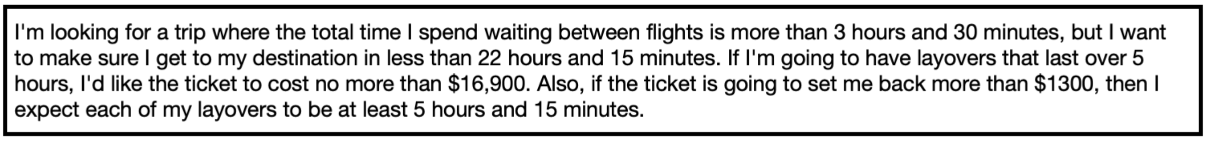
B.3 4 slots, 2 minterms



I'm looking for a flight that costs $4500 or less. If I have any layovers shorter than two hours, I'd like to fly only in first class. Also, if I'm going to arrive before 3:45 pm, I'd rather not be in business class. And if I'm set to get in before 2:00 pm, I want to make sure that all of my layovers are at least two hours long.

Figure 9: Sample query in GroundCocoa with 4 slots and 2 minterms.

B.4 4 slots, 3 minterms



I'm looking for a trip where the total time I spend waiting between flights is more than 3 hours and 30 minutes, but I want to make sure I get to my destination in less than 22 hours and 15 minutes. If I'm going to have layovers that last over 5 hours, I'd like the ticket to cost no more than $16,900. Also, if the ticket is going to set me back more than $1300, then I expect each of my layovers to be at least 5 hours and 15 minutes.

Figure 10: Sample query in GroundCocoa with 4 slots and 3 minterms.

B.5 5 slots, 2 minterms

```
I'd like all my layovers to be in Bangkok and they should be no shorter than 5 hours and 30
minutes. I'd rather not travel in economy class. If it's not possible for me to get to my
destination by 18:45, then I'd like to leave between 18:15 and 19:45. Also, I need to make
sure I arrive by 11:45, unless my flight takes off before 7:30 or after 8:15.
```

Figure 11: Sample query in GroundCocoa with 5 slots and 2 minterms.

B.6 6 slots, 2 minterms

```
I need to get to my destination by 7:45 PM on April 17th, 2024. I'm looking for a flight
that's not better for the environment, with carbon emissions above the average. I'd like to
leave before 3:00 PM, but if that's not possible, I'll take a direct flight. Also, if I'm
not flying business class, I need to take off by 2:45 PM at the latest. And if my flight is
a direct flight, I'd rather not be in first class.
```

Figure 12: Sample query in GroundCocoa with 6 slots and 2 minterms.

C Prompts used in Model Evaluation

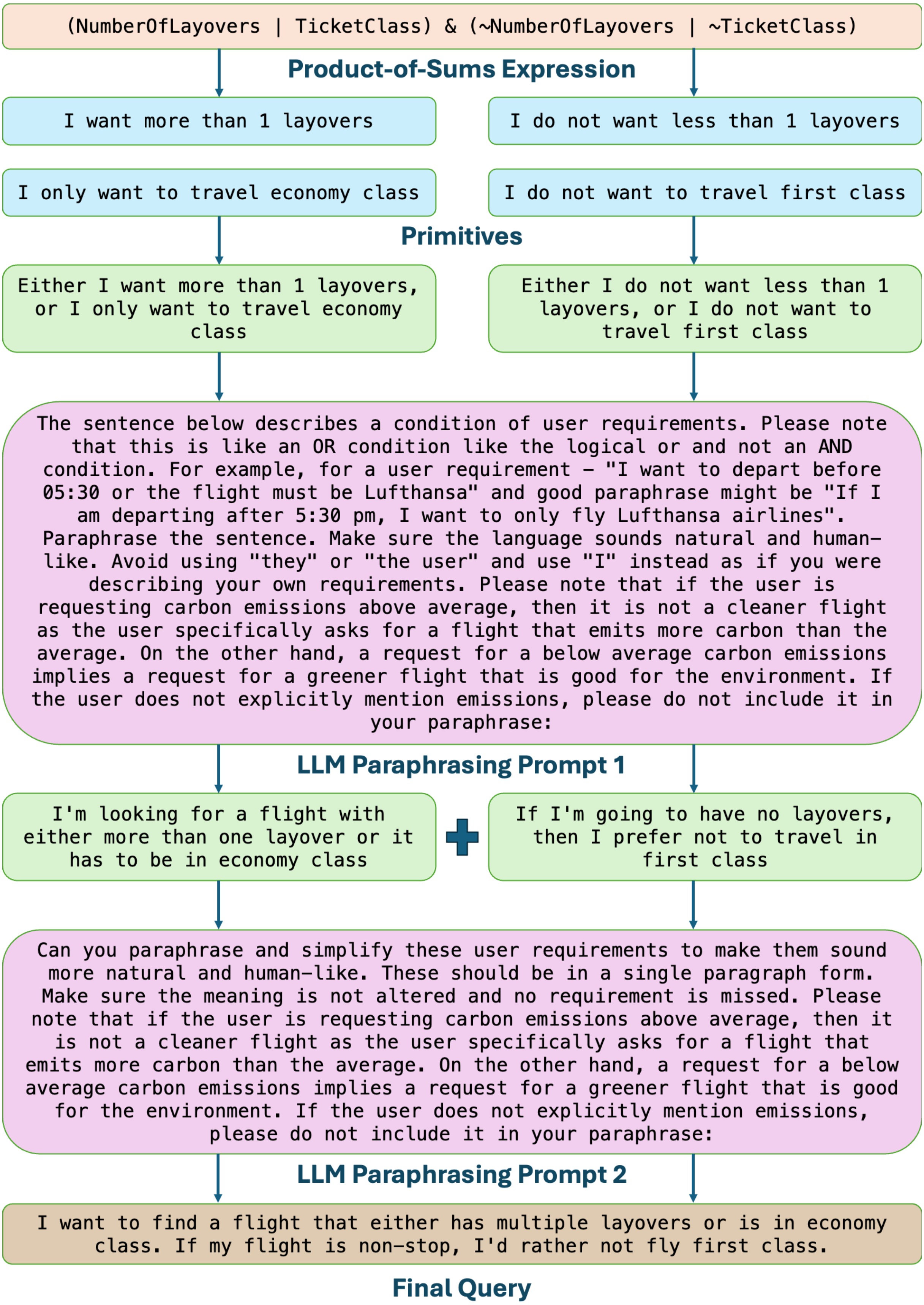
```
A user has specified certain criteria for booking a flight. Below are five different flight
options labeled as 'A', 'B', 'C', 'D', and 'E'. Review these options and select the one
that best matches the user requirements. Respond with a single option and the phrase 'The
answer is Option ' followed by the correct letter - 'A', 'B', 'C', 'D', or 'E' -

Q. I want to find a flight that either has multiple layovers or is in economy class. If my
flight is non-stop, I'd rather not fly first class.

Option A:
.
.
.
```

Figure 13: Evaluation prompt without CoT.

D End-to-End Generation Process



(NumberOfLayovers | TicketClass) & (~NumberOfLayovers | ~TicketClass)

Product-of-Sums Expression

I want more than 1 layovers

I do not want less than 1 layovers

I only want to travel economy class

I do not want to travel first class

Primitives

Either I want more than 1 layovers, or I only want to travel economy class

Either I do not want less than 1 layovers, or I do not want to travel first class

The sentence below describes a condition of user requirements. Please note that this is like an OR condition like the logical or and not an AND condition. For example, for a user requirement – "I want to depart before 05:30 or the flight must be Lufthansa" and good paraphrase might be "If I am departing after 5:30 pm, I want to only fly Lufthansa airlines". Paraphrase the sentence. Make sure the language sounds natural and human-like. Avoid using "they" or "the user" and use "I" instead as if you were describing your own requirements. Please note that if the user is requesting carbon emissions above average, then it is not a cleaner flight as the user specifically asks for a flight that emits more carbon than the average. On the other hand, a request for a below average carbon emissions implies a request for a greener flight that is good for the environment. If the user does not explicitly mention emissions, please do not include it in your paraphrase:

LLM Paraphrasing Prompt 1

I'm looking for a flight with either more than one layover or it has to be in economy class

If I'm going to have no layovers, then I prefer not to travel in first class

Can you paraphrase and simplify these user requirements to make them sound more natural and human-like. These should be in a single paragraph form. Make sure the meaning is not altered and no requirement is missed. Please note that if the user is requesting carbon emissions above average, then it is not a cleaner flight as the user specifically asks for a flight that emits more carbon than the average. On the other hand, a request for a below average carbon emissions implies a request for a greener flight that is good for the environment. If the user does not explicitly mention emissions, please do not include it in your paraphrase:

LLM Paraphrasing Prompt 2

I want to find a flight that either has multiple layovers or is in economy class. If my flight is non-stop, I'd rather not fly first class.

Final Query

Figure 15: End-to-End query generation with 2 slots and 2 minterms.

```
Consider the given example and answer the question that follows.

A user has specified certain criteria for booking a flight. Below are two different flight
options labeled as 'A' and 'B'. Review these options and select the one that best matches
the user requirements. Respond with a single option and the phrase 'The answer is Option '
followed by the correct letter - 'A', or 'B'

Q. I'd like my flight to have layovers that are no shorter than an hour each. It's important
to me that my flight is environmentally friendly, with carbon emissions lower than the
average, unless my layovers are exclusively in Atyrau and Dubai. I want to get to my
destination before noon, but if that's not doable, then I don't want to take off between
7:45 pm and 10:45 pm. Ideally, my flight should depart between 1:30 pm and 3:30 pm, but if
that can't happen, I want to pay a fare of $1800 or more. I'm looking to spend less than
$1600 on my ticket, but I insist on a flight that emits more carbon than average if that's
not an option. Also, if I'm set to arrive before noon, I'd prefer not to have any layovers
in Frankfurt am Main.

Option A:{'Airline': '[Emirates]', 'Ticket Class': 'First', 'Departure Time': '4:15 AM',
'Arrival Time': '1:15 PM', 'Travel Time': '12 hr 30 min', 'Number Of Layovers': '1', 'Carbon
Emission Avg Diff ( % )': '+32', 'Price': '$7379.0', 'Travel Date': '2024-04-17', 'Source
City': 'New Delhi', 'Destination City': 'Amsterdam', 'Layover 1 Location': 'Dubai', 'Layover
1 Time': '1 hr 45 min ', 'Flight 1 Features': '[Individual suite, Wi-Fi for a fee, In-seat
power & USB outlets, On-demand video, Carbon emissions estimate: 840 kg]', 'Flight 2
Features': '[Individual suite, Wi-Fi for a fee, In-seat power & USB outlets, On-demand
video, Carbon emissions estimate: 1,828 kg]'}

Option B:{'Airline': '[Lufthansa]', 'Ticket Class': 'Economy', 'Departure Time': '2:50 AM',
'Arrival Time': '10:35 AM', 'Travel Time': '11 hr 15 min', 'Number Of Layovers': '1',
'Carbon Emission Avg Diff ( % )': '+9', 'Price': '$979.0', 'Travel Date': '2024-04-17',
'Source City': 'New Delhi', 'Destination City': 'Amsterdam', 'Layover 1 Location':
'Frankfurt am Main', 'Layover 1 Time': '1 hr 10 min ', 'Flight 1 Features': '[Often delayed
by 30+ min, Average legroom (31 in), Wi-Fi for a fee, In-seat power & USB outlets, On-demand
video, Carbon emissions estimate: 481 kg]', 'Flight 2 Features': '[Average legroom (30 in),
Wi-Fi for a fee, Carbon emissions estimate: 49 kg]'}

Let's think Step by Step:

Option A has a single layover of 1 hr and 45 minutes which is longer than the users
specified minimum of 1 hour. While the flight is not environmentally friendly and emits 32%
more carbon than the average, the layover is exclusively in Dubai matching the user
alternate condition. The flight has an arrival time of 1:15 PM and does not reach the
destination before noon, but the departure time is 4:15 AM and not between 7:45 pm and 10:45
pm per the user's requirement. The flight doesn't depart between 1:30 pm and 3:30 pm, but
the price is $7379 which is greater than $1800. The ticket price is above $1600 but the
flight emits more carbon than average to satisfy the users alternate preference. Finally,
the arrival time is afternoon but there are no layovers in Frankfurt - with the sole layover
being in Dubai. Option A thus matches all the users criteria.

Option B has a single layover of 1 hr and 10 minutes which is longer than the users
specified minimum of 1 hour. The flight is not environmentally friendly with a carbon
emissions estimate of 9% above the average and also has a layover in Frankfurt and not
exclusively in Atyrau and Dubai - failing the users criteria. The arrival time is 10:35 AM
which is before noon per the user's requirement. However, the flight neither departs between
1:30 pm and 3:30 pm, nor is the price greater than $1800 and fails this condition. The
departure time is 2:50 am and the ticket price is $979. The ticket price is under $1600
satisfying the next user requirement. The flight meets the users final criteria - it arrives
before noon. Option B thus does not match the users criteria due to the combination of
Layover Location and emission preferences, as well as price and departure time.

Answer is Option A

A user has specified certain criteria for booking a flight. Below are five different flight
options labeled as 'A', 'B', 'C', 'D', and 'E'. Review these options and select the one that
best matches the user requirements. Respond with a single option and the phrase 'The answer
is Option ' followed by the correct letter - 'A', 'B', 'C', 'D', or 'E'

Q. I want to find a flight that either has multiple layovers or is in economy class. If my
flight is non-stop, I'd rather not fly first class.

Option A:
.
.
.
```

Figure 14: Evaluation prompt using CoT-partial.