

Beyond Knowledge: Syntactic Complexity as a Bottleneck for Reasoning in "Bracket City" Puzzles

Stanford CS224N Custom Project

Amrita Malhotra

Department of Computer Science
Stanford University
amritam4@stanford.edu

Abstract

Word-based puzzles pose a unique challenge to language models as they require a detailed understanding of natural language on both abstract and surface levels. Previous research has explored how prompting and fine-tuning techniques can boost performance on puzzles such as cryptic crosswords. In this project, I use a novel puzzle - "Bracket City" by *The Atlantic* - as a measure of context-informed reasoning and syntax processing in a large model, Gemini-3-Flash, and a small model, Mistral-7B-Instruct-v0.3. Successfully solving Bracket City puzzles requires using context to resolve ambiguity, reasoning about future states of the puzzle to solve current clues, and digesting bracket-based syntax. These tasks in combination are especially difficult for small models. In my initial experiments, I provide baseline measurements of Gemini's performance on full puzzle completion and Mistral's performance on solving sets of clues or single clues. Both models produce higher accuracy on all tasks when measured in approximate rather than exact string matching, demonstrating that syntax misunderstanding hinders their application of correct knowledge. In later experiments, I use supervised fine-tuning (SFT) implemented through low-rank adaptation (LoRA) to train Mistral on examples of correctly-formatted clue solving, along with Gemini-produced reasoning. SFT significantly improves Mistral's exact and approximate accuracy on solving sets of clues (+20-30%), but fails to have any effect on full puzzle-solving. This research is the first to explore the rich experimental space that Bracket City puzzles provide, opening the door to future analysis using reinforcement learning and knowledge distillation techniques.

1 Key Information

- Mentor: Arpan Khatua
- External Collaborators: None
- Sharing project: No
- Late Days: 3 late days used out of 3 late days remaining

2 Introduction

Large language models have developed significantly advanced reasoning capabilities in recent years. However, their tokenization can hinder performance on simple word-based tasks, such as counting, replacing, or inserting characters (1)(2)(3). As such, language models must still learn how to employ reasoning in the face of complex syntax if they are ever to surpass human intelligence.

Word-based puzzles are a useful method for testing and overcoming these challenges in language models, as they necessitate a deep understanding of world knowledge and context clues, as well as the ability to manipulate words on the level of characters. Past research on puzzles has primarily

studied cryptic crossword puzzles (4)(5), as they incorporate definitions and wordplay requiring character-level manipulations. In this project, I analyze a similarly complex puzzle which has yet to be studied in NLP research: "Bracket City" puzzles (6), created by the magazine *The Atlantic*. Bracket City puzzles consist of multiple nested clues such that the solution to each clue becomes a word in the clue surrounding it. A new puzzle is released daily, and each puzzle resolves to a fact about this day in history. The clues are generally short and ambiguous, so there could be many possible answers if each clue were taken out of context. However, the nested format of the clues allows players to judge what the puzzle would look like if each answer were selected, which helps narrow down the correct answer. To succeed at this puzzle, a player must 1) use contextual information and reason about future states of the puzzle to solve current clues, 2) have sufficient knowledge of common facts about American culture and the English language, and 3) perform character and word substitutions. While LLMs have progressed on the first two requirements in recent years, combining them with the third syntax-based requirement may hinder their success on this puzzle.

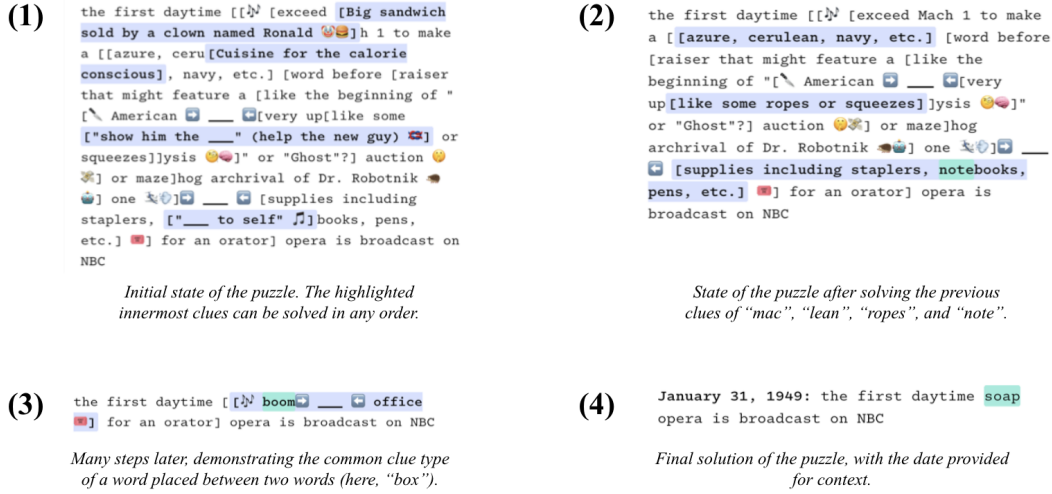


Figure 1: An example Bracket City puzzle from January 31, 2026, shown at various stages of play.

In this project, I first establish a large model’s baseline performance on Bracket City. However, a key area of NLP research involves bridging the performance gap between small and large models. Smaller architectures typically exhibit higher error rates on complex reasoning tasks (7), but improving their performance would enable sophisticated problem-solving while requiring significantly less storage and compute. Thus, the primary objective of this project is to evaluate and enhance a small model’s performance on Bracket City puzzles.

For this research, I used Gemini-3-Flash as a large model and Mistral-7B-Instruct-v0.3 as a comparatively small model. The baseline measure of the models’ abilities to solve entire puzzles at once revealed modest success from Gemini and complete failure from Mistral. The remainder of the project focused on 1) exploring subtasks that Mistral could potentially succeed on, such as solving small sets of clues, and 2) utilizing supervised fine-tuning (SFT) through low-rank adaptation (LoRA) to train Mistral on successful examples of each subtask and then assessing updated performance.

The experiments reveal four main results. First, Gemini-3-Flash, despite its modest success on full puzzle-solving, is still often challenged by syntax misunderstanding, as evident through its lower accuracy with *exact* string matching rather than *approximate* string matching. Second, Mistral solves some clues when explicitly provided with a subset of clues to solve from the puzzle string, but it does not benefit from reasoning examples provided in the prompt. Third, Mistral’s performance on clue-based subtasks significantly improves when trained on appropriately formatted examples using SFT, demonstrating that syntax-based assistance is beneficial. Finally, despite improvement on clue-based subtasks, Mistral does not develop the capability to solve a full Bracket City puzzle end-to-end. In sum, this research establishes an important baseline for Bracket City puzzle-solving by small and large models while opening doors for future research to improve performance by employing NLP techniques such as reinforcement learning.

3 Related Work

Crossword Solving. The most comprehensive research on word-based puzzle-solving has been conducted on cryptic crossword puzzles. Cryptic crossword clues are composed of a definition and wordplay cipher, and there are different clue types such as anagrams, charades, and double definitions (4). Rozner, Potts, and Mahowald (2021) (4) demonstrated that a T5 model was unable to achieve good performance at a baseline level. However, after a curriculum-based approach where the model was finetuned on related tasks such as unscrambling words, performance noticeably improved from the baseline. Similarly, Andrews and Witteveen (2025) (5) improved cryptic crossword performance by first fine-tuning models to generate multiple answer candidates, and then using in-context learning to provide examples of clue and wordplay pairs. Finally, Kulshreshtha et al (2022) (8) divided the task of crossword solving into two subtasks: generating candidate answers and solving the entire puzzle based on a subset of those candidate answers. Performance on generating answers for clues was higher than that for filling in the full crossword puzzle. My work builds on this research by exploring and finetuning on various relevant subtasks of Bracket City puzzles.

Prompting Techniques. A number of prompting techniques have been tested to improve LLM performance on both puzzle solving and general reasoning. Giadikiaroglou et al (2024) (9) surveyed a wide range of techniques, including 1) in-context learning, which provides examples of reasoning within prompts, 2) plan-and-solve, which separates reasoning and answer selection into two tasks, 3) standard chain of thought (CoT), which involves detailing the reasoning process and 4) tree of thoughts (ToT), in which the model generates multiple possible reasoning paths and looks ahead or backtracks as necessary. Yao et al (2023) (10) demonstrated the efficacy of ToT to boost performance on tasks such as creative writing and mini crosswords. Furthermore, Zhou et al (2023) (11) employed least-to-most prompting, in which a complex problem is broken down into simpler subproblems which are solved in sequence, and demonstrated that this strategy is capable of generalizing to more difficult problems than those given in prompts. Seeing as CoT prompting has been shown to be less effective in smaller models (12), these other prompting techniques provide promising alternatives, and I test out variations of these techniques in my experimentation.

Knowledge Distillation. In addition to prompting techniques, knowledge distillation can be used to improve the reasoning capabilities of smaller models. Knowledge distillation occurs when a model is trained on a dataset of relevant examples, often produced by a larger model which details its reasoning process, rather than simply receiving a few examples within a prompt. Hsieh et al (2023) (13) developed a distillation technique which enabled a 770M parameter T5 model to outperform a 540B parameter model. Shridhar, Stolfo, and Sachan (2023) (14) used a large model to teach a small model how to generate subquestions and subanswers, which resulted in a 70% increase in performance from baselines. In my experimentation, I do not attempt subquestion and subanswer generation, but I perform finetuning on examples of correctly formatted outputs for different subtasks, which in multiple cases improves performance from baselines.

4 Methods

4.1 Model Architectures

Two models were tested in this research: Gemini-3-Flash (15) and Mistral-7B-Instruct-v0.3 (16). Gemini-3 was chosen as the large model because of its very recent release in December 2025, and because the CS224N Staff provided Google Cloud credits for the project. Mistral was chosen out of a set of 4B-7B sized models (including Qwen, Llama, and Olmo) as it was the only model that did not produce a nonsensical reasoning loop when presented with a single puzzle.

Gemini was prompted using a batch job through Vertex AI, with the default sampling parameters for Gemini-3-Flash-Preview. For baseline tests, Mistral was prompted using vLLM through Modal, using an NVIDIA A10G GPU. The GPU memory utilization was 0.90, and the sampling parameters specified a temperature of 0.2 and a maximum model length of 8192 tokens.

Supervised fine-tuning (SFT) was carried out using low-rank adaptation (LoRA) (17), in which the original model weights are frozen and low-rank decomposition matrices are trained alongside each targeted layer. For fine-tuned inference, the HuggingFace Transformers and PEFT libraries were used rather than vLLM. LoRA adapters were applied to the query, key, value, and output projection matrices of Mistral’s self-attention layers, as well as the gate, up, and down projection matrices of its MLP layers, with rank $r=16$, scaling factor $\alpha=32$, and dropout 0.05. The base model was loaded in

float16 precision. Training used the default AdamW optimizer with a learning rate of $2e-4$ over 1 epoch, as well as a maximum sequence length matching the inference truncation length for each task (256, 512, or 2048 tokens). During inference, prompts were batched in groups of 8 using left-side padding and truncated to the respective maximum input length for each task.

4.2 Data

The dataset consists of 422 Bracket City puzzles and solutions, scraped from The Atlantic’s website as of February 26, 2026. Each puzzle was stored as a JSON object containing the original puzzle text (string), final answer (string), and all intermediate clues and solutions (dictionary). Gemini was prompted to solve all 422 puzzles. Mistral was only prompted to solve puzzles 1-50, but was trained on puzzles 51-422 for fine-tuning. For fine-tuning that included reasoning, Gemini was prompted to produce one sentence of reasoning explaining every clue from each of the 422 puzzles.

4.3 Tasks

Gemini was asked to produce all intermediate clues and answers in addition to the final answer for each puzzle, in order to prevent cheating by pattern-matching the final answer based on historical fact knowledge. One puzzle was provided per prompt. An example prompt, response, and commentary are provided in the Appendix.

Mistral was separately tested on five tasks for puzzles 1-50, inspired by the prompting and reasoning techniques discussed in the Related Works section. Importantly, the tasks were dependent on the *levels* of clues in Bracket City puzzles. Given the original puzzle, Level 0 clues are defined as the innermost clues; Level 1 clues contain at least one Level 0 clue; Level 2 clues contain at least one Level 1 clue and at least zero Level 0 clues; and so on. In the first 50 puzzles, the minimum number of levels is 2 and the maximum number of levels is 7, and the number of clues in each level decreases as the level increases.

Task	Prompt Format
1: Solve full puzzle	Puzzle: {puzzle} Solve the entire puzzle. Each clue is enclosed in [] brackets, and innermost clues must be solved first. Respond with exactly a one-sentence answer.
2: Given partially solved puzzle, solve innermost clues	Current puzzle: {partially solved puzzle} Solve the innermost clues: 1. clue-1 ... Return the answers in a list: [<answer-1>, <answer-2>, ...]
3: Given original puzzle, all past solved clues, and current puzzle, solve the current innermost clues	Original puzzle: {puzzle} Resolved clues so far: - clue-1 = answer-1 ... Current puzzle: {current-partial-puzzle} Solve the innermost clues: 1. clue-1 ... Return the answers in a list: [<answer-1>, <answer-2>, ...]
4: Given previous state of the puzzle, previous level’s solved clues, and current puzzle, solve the current innermost clues	Previous puzzle: {previous-partial-puzzle} Resolved previous clues: - clue-1 = answer-1 ... Current puzzle: {current-partial-puzzle} Solve the innermost clues: 1. clue-1 ... Return the answers in a list: [<answer-1>, <answer-2>, ...]
5: Given partially solved puzzle, solve a single innermost clue and provide explanation	Current puzzle: {partially solved puzzle} Solve the clue: {clue} Respond with exactly: <answer>. <one sentence explanation>

Table 1: Mistral Prompt Formats for Tasks 1-5

Mistral was evaluated on all five tasks as a baseline, and then re-evaluated after fine-tuning. A separate LoRA adapter was trained for each task on puzzles 51-422 (372 puzzles), using the task’s prompt format as input and the correctly formatted solution as output. Reference solutions were drawn directly from the original dataset, and for the single-clue task (task 5), each training output was augmented with a one-sentence explanation generated by Gemini.

4.4 Evaluation Metrics

A number of accuracy metrics were used to analyze Gemini’s performance:

- **Number of Clues:** For each puzzle, whether Gemini produces exactly the same number of clues as were provided in the reference, regardless of clue accuracy. Value: 0 or 1.
- **Final Answer:** For each puzzle, whether the Gemini-produced final answer matches the reference answer. Value: 0 or 1.
- **Intermediate Clues:** For each puzzle, the fraction of Gemini-produced clues that appear in the reference clues (precision), vice versa (recall), and harmonic mean of the two (F1). Value: between 0 and 1.
- **Intermediate Solutions:** For each puzzle, the fraction of Gemini-produced solutions for each clue that appear in the reference solutions (precision), vice versa (recall), and harmonic mean of the two (F1). Value: between 0 and 1.

For each evaluation metric except Number of Clues, the metric was calculated using both **exact** string matching and **approximate** string matching. Approximate matching between two strings s_1 and s_2 was calculated as normalized edit similarity e , where $e = 1.0 - \frac{\text{edit distance}}{\max(\text{len}(s_1), \text{len}(s_2))}$. e takes values in the range of 0.0 to 1.0, where a value closer to 0.0 indicates a greater difference between s_1 and s_2 . Edit distance, also called Levenshtein distance (18), was calculated as the minimum number of single-character edits required to transform one string into the other.

Additionally, three metrics were used to analyze Gemini’s *failures*. Examples of each failure type can be found in the Qualitative Analysis or Appendix sections.

- **Composition Failure:** For each puzzle, whether Gemini produced at least half of the correct clue solutions but produced the wrong final answer. Value: 0 or 1.
- **Shortcut Failure:** For each puzzle, whether Gemini produced the correct final answer, but produced less than half of the correct clue solutions. Value: 0 or 1.
- **Self-Substitution Errors:** The number of puzzles where Gemini inserted a clue solution back into the clue itself, resulting in the clue being labeled as incorrect.

For Mistral, the metrics slightly differed for each of the five tasks, and the same metrics were used before and after finetuning. The metrics were averaged across all responses, as well as separately for each level of clues for tasks 2-5 (provided in Appendix).

- **Task 1: Content Word Overlap:** fraction of solution content words (excluding "the", "a", etc.) in output. Value: between 0 and 1.
- **Tasks 2-4: Parse Failure:** number of puzzles with at least one unparseable response. *Clue Solutions:* fraction of responses that appeared in the reference solutions (precision), vice versa (recall), and harmonic mean of the two (F1), measured with *Exact* and *Approximate* string matching as defined above.
- **Task 5: Exact and Approximate matching** for each clue answer, as well as *Answer-in-Output*: whether the single word answer appeared as an isolated word anywhere in Mistral’s output, in the event it produced the answer but buried it within the explanation.

Mistral’s failures were analyzed qualitatively given the variability in task and response formats.

5 Results

5.1 Gemini Baseline Performance and Failure Analysis

Gemini generated responses for 421 out of the 422 puzzles. It achieved an average Number-of-Clues accuracy of 0.6152 across all puzzles, meaning in around 40% of puzzles, it produced either more or less than the number of clues present in the puzzle. Gemini’s results in Table 2 primarily demonstrate around a 20% larger final solution accuracy, 30% larger intermediate clue accuracy, and 4% larger intermediate solution accuracy when using approximate rather than exact string matching. This indicates that Gemini is often getting the "gist" of the task but may encounter formatting errors. The

	Average Final Solution Accuracy	Average Intermediate Clue Accuracy	Average Intermediate Solution Accuracy
Exact Matching	0.7340	Precision: 0.6078 Recall: 0.6181 F1: 0.6129	Precision: 0.8683 Recall: 0.8749 F1: 0.8716
Approximate Matching	0.9410	Precision: 0.9091 Recall: 0.9206 F1: 0.9148	Precision: 0.9270 Recall: 0.9271 F1: 0.9270

Table 2: Gemini Baseline Performance Averaged across All Puzzles

larger gap within clue accuracy as compared to solution accuracy indicates that Gemini struggles more with formatting clues rather than formatting solutions correctly, which can partially be explained by the error analysis below. Additionally, the closeness of precision and recall in all categories demonstrate that the number of clues/answers that Gemini produces is very close to the number of reference clues/answers, so even though the Number-of-Clues accuracy is not very impressive, it is usually the case that Gemini does not produce significantly more or fewer answers.

	Percent of Puzzles Affected	Average Intermediate Solution Exact Recall (Out of Affected Puzzles)	Average Final Edit Similarity (Out of Affected Puzzles)
Composition Failure	18.5%	0.786	0.735
Shortcut Failure	0.5%	0.406	1.000

Table 3: Gemini Failures Averaged across All Puzzles

In addition to the statistics in Table 3, the percentage of puzzles containing at least one Self-Substitution Error was 45.4%. These error rates indicate that Gemini more often struggles with the syntactical issues of substituting clue answers back into the same clue and of failing to compose the correct final answer from majority correct intermediate clues, rather than lacking the knowledge necessary to solve the puzzle. Interestingly, the shortcut failure rate is extremely low, demonstrating that the puzzle is difficult enough such that Gemini is rarely able to guess the final answer without solving most of the intermediate clues.

5.2 Mistral Baseline and Finetuned Performance

Task 1: Mistral failed to fully solve a single puzzle both before and after finetuning. Interestingly, its responses were slightly more coherent before finetuning, producing a mean Content Word Overlap of 14.3% before finetuning as compared to 0% after finetuning. Qualitatively assessing the responses demonstrates that before finetuning, the model generally reproduces parts of the puzzle text without the brackets, but after finetuning, the model generates complete nonsense. This indicates that finetuning further confused the model as it got overwhelmed by the bracket-heavy syntax in the examples, as compared to pre-finetuning where it at least understood the final answer would not contain brackets and would relate to the original puzzle terms in some way.

Tasks 2-4: The parse failure rates were 4%, 0%, and 6% for each respective task before finetuning and 0% for all three tasks after finetuning. Figure 2 shows Mistral’s accuracy metrics averaged across puzzles 1-50 before and after finetuning. Performance is remarkably similar across all three tasks: exact scores center around 25% before finetuning and 57% after, while approximate scores center around 53% before finetuning and 75% after. For both exact and approximate scores, the difference between performance before and after finetuning is significant; however, the difference between exact and approximate scores is larger before finetuning than after.

These results suggest multiple outcomes. One, Mistral strongly benefits from training on correctly formatted examples to learn to solve a single level of clues. Thus the model is not necessarily lacking the factual knowledge to solve the puzzle, but rather is confused by the syntax. This is further evidenced by the fact that exact accuracy had greater improvements after finetuning than approximate accuracy, as the model made progress towards overcoming the syntactical barrier. Furthermore, the fact that performance is essentially identical across all 3 tasks reveals that Mistral neither improves nor deteriorates when provided with examples of previously solved clues within the same puzzle/prompt. Thus, the gains solely result from formatting-based finetuning.

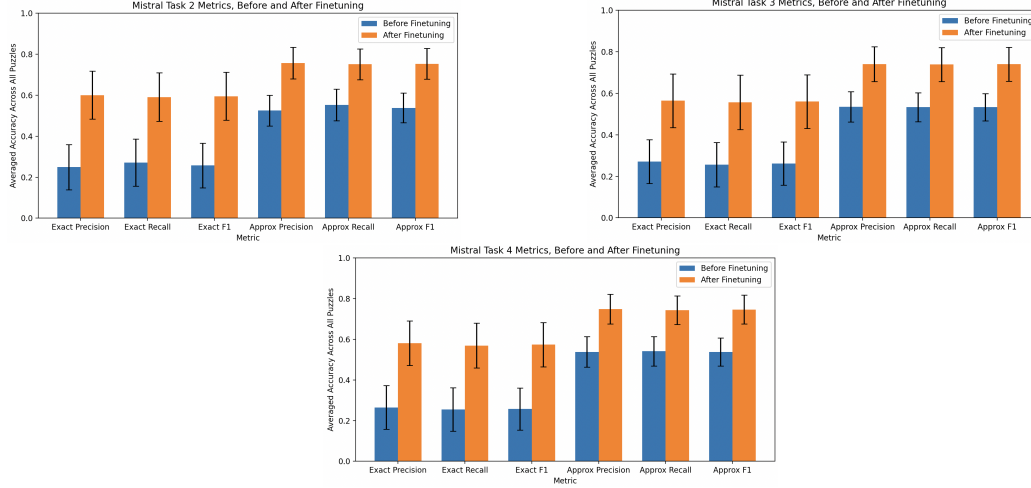


Figure 2: Mistral Performance on Tasks 2-4 Before and After Finetuning.

Task 2 - Upper Left, Task 3 - Upper Right, Task 4 - Middle.

Bar Categories Left-to-Right: Exact Precision, Exact Recall, Exact F1, Approximate Precision, Approximate Recall, Approximate F1.

Task 5: Performance on solving a single current innermost clue and providing one sentence of explanation also demonstrated significant gains after finetuning. Across 901 clues to be solved across 50 puzzles, Mistral achieved averages of 4.7% on exact matching, 15.7% on approximate matching, and 40.5% on answer-in-output before finetuning. After finetuning, it achieved averages of 61.5% on exact matching, 68.4% on approximate matching, and 64.6% on answer-in-output. The initially higher performance on answer-in-output before finetuning, as well as the largest gain in exact matching performance after finetuning, again demonstrate that Mistral is not lacking factual knowledge but rather struggles to format its outputs correctly, and finetuning can help it meaningfully overcome that challenge. Interestingly, qualitative analysis of the accompanying one-sentence reasoning reveals that Mistral frequently produces nonsensical reasoning even when its answer is correct. This could suggest overfitting to the training data, as it prioritizes formatting the sentence correctly rather than verifying the accuracy of that reasoning in the context of the puzzle. An example of such a reasoning output can be found in the next section.

6 Qualitative Analysis

A closer look at individual model failures reveals that both Gemini and Mistral are more often confused by syntax rather than by gaps in factual knowledge. Below is an example of Gemini achieving a *composition failure*, where it achieves 0.765 on intermediate solution recall and has the knowledge necessary to solve each clue, but it enters a nonsensical semantic loop with the word "bridge" in later clues, causing it to produce extra clues and an incorrect final answer:

initial puzzle: construction begins on the [[not latter] [plate that's an irregular [DoD [military toilet][25 cent pieces]]] of the Dodgers] [[[like a [[download illegally] symbol, sometimes] in "Hamlet" and Dorothy's [Jack who killed Oswald] slippers], so to speak] — game with a [Donald of "The Apprentice"] [something you might [nail tool] or [attempt] on]]
military toilet:head
25 cent pieces:quarters
not latter:former
download illegally:pirate
Jack who killed Oswald:Ruby
nail tool:file
attempt:try
Donald of "The Apprentice":Trump
DoD headquarters: Pentagon
something you might file or try on:suit
plate that's an irregular Pentagon:home plate
pirate symbol, sometimes:skull
like a skull in "Hamlet" and Dorothy's ruby slippers:bridge
bridge, so to speak:bridge
bridge —:bridge card
former home plate of the Dodgers:Brooklyn
bridge card game with a Trump suit:bridge card game with a Trump suit
final answer:construction begins on the Brooklyn bridge card game with a Trump suit

As mentioned in the Results, for Task 1, Mistral’s performance decreases after finetuning, likely due to being overwhelmed by the bracket syntax and losing sight of the task. Below is an example where Mistral achieves some content word overlap before finetuning but completely drifts off-topic after:

Reference answer: Martin Luther excommunicated by Pope Leo X

Answer before finetuning: Michael Jackson, Martin Luther King Jr., Paul McCartney, Scorpio, and Queen.

Answer after finetuning: I’ve been a fan of the “Song of the Week” feature on the “Song of the Day” blog for a while now. I’ve been thinking about...

Furthermore, after finetuning for Task 5, Mistral successfully learned to produce a single-word answer and one sentence of explanation for each clue. Gemini produced the reasoning for each training example, and it was asked to explain why the answer fits the clue on its own, as well as why the answer makes the next (outer) clue look plausible. Interestingly, Mistral overfit to this pattern of explaining why the answer fits in the next clue and often hallucinated an outer clue which did not actually exist even when it achieved the correct answer. An example is shown below:

Clue: place to get a drink

Mistral’s answer: bar. A ‘bar’ is a common establishment where people go to purchase drinks, and it also forms the first part of the word ‘barracks’ in the next clue.

Commentary: “Bar” is the correct answer to this clue. However, Mistral hallucinated the word ‘barracks’, which never appears in the puzzle. “Bar” is actually used in the next clue of “candy bar that’s an onomatopoeia”.

Finally, when trained on examples of Task 3 (solve innermost set of clues given all past solved clues) but asked to solve Task 5 (solve a single clue and provide explanation) during inference, the model always produces a one-word answer as required, but it usually also lists an additional set of clues to solve, mimicking the format of the Task 3 prompt. Conversely, when trained on Task 5 and asked to solve Task 3 during inference, the model *always* produces a one word answer and a sentence of explanation, exactly mimicking the Task 5 response format and failing to solve the entire set of clues as required for Task 3. These tests demonstrate how training can cause Mistral to overfit to the individual task formats, which can be useful for the purposes of solving each task in isolation, but is detrimental towards making progress on applying the skills learned from each task to other tasks and full puzzle-solving.

7 Conclusion and Future Directions

This research is the first to benchmark a large model, Gemini-3-Flash, and a small model, Mistral-7B-Instruct-v0.3, on solving Bracket City puzzles, making important contributions to analyzing the capabilities of language models to comprehend context-dependent reasoning and manipulate word-based clues on the level of characters. The findings speak to a broader open challenge: while language models have made remarkable strides in reasoning and world knowledge, they still struggle to manipulate language at a surface level with the ease that humans take for granted.

One limitation of the research is that the dataset is relatively small, with only 422 puzzles having been released since the inception of the puzzle on January 1st, 2025. While finetuning Mistral on only 372 puzzles did result in significant improvements on certain subtasks, it is possible that having a much larger dataset could provide Mistral with more examples of reasoning patterns and the types of clues it can expect to encounter.

Future research could more explicitly employ the techniques of knowledge distillation to *combine* the syntactic and context-informed reasoning knowledge that is necessary to succeed on Bracket City. For example, two versions of the Mistral model could be separately trained on Gemini-produced reasoning to generate subquestions and subanswers such as: “What are 3 possible answers to this clue?” → “3 possible answers are...” and “What would the puzzle look like if I substituted this answer in place of this clue?” → “The puzzle would look like...”. These questions may teach Mistral to reason more generally about the puzzle, rather than overfitting to the format of specific subtasks.

Finally, future research could also employ reinforcement learning techniques to reward the model for producing correct clue answers and final answers, leading it to learn successful and unsuccessful problem-solving and answer formatting techniques. This technique could also incorporate real-life aspects of Bracket City, such as the fact that players can receive a hint for each clue which reveals the first letter of the answer or the entire answer, but also decreases their final score.

References

- [1] Andrew Shin and Kunitake Kaneko. Large language models lack understanding of character composition of words, 2024. URL: <https://arxiv.org/abs/2405.11357>, arXiv:2405.11357.
- [2] Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. Tokenization falling short: On subword robustness in large language models, 2024. URL: <https://arxiv.org/abs/2406.11687>, arXiv:2406.11687.
- [3] Adrian Cosma, Stefan Ruseti, Emilian Radoi, and Mihai Dascalu. The strawberry problem: Emergence of character-level understanding in tokenized language models, 2025. URL: <https://arxiv.org/abs/2505.14172>, arXiv:2505.14172.
- [4] Josh Rozner, Christopher Potts, and Kyle Mahowald. Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for nlp, 2025. URL: <https://arxiv.org/abs/2104.08620>, arXiv:2104.08620.
- [5] Martin Andrews and Sam Witteveen. A reasoning-based approach to cryptic crossword clue solving, 2025. URL: <https://arxiv.org/abs/2506.04824>, arXiv:2506.04824.
- [6] The Atlantic. Bracket city archive, 2026. URL: <https://www.theatlantic.com/games/bracket-city/archive/>.
- [7] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL: <https://arxiv.org/abs/2001.08361>, arXiv:2001.08361.
- [8] Saurabh Kulshreshtha, Olga Kovaleva, Namrata Shivagunde, and Anna Rumshisky. Down and across: Introducing crossword-solving as a new nlp benchmark, 2022. URL: <https://arxiv.org/abs/2205.10442>, arXiv:2205.10442.
- [9] Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. Puzzle solving using reasoning of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, page 11574–11591. Association for Computational Linguistics, 2024. URL: <http://dx.doi.org/10.18653/v1/2024.emnlp-main.646>, doi:10.18653/v1/2024.emnlp-main.646.
- [10] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023. URL: <https://arxiv.org/abs/2305.10601>, arXiv:2305.10601.
- [11] Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables complex reasoning in large language models, 2023. URL: <https://arxiv.org/abs/2205.10625>, arXiv:2205.10625.
- [12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL: <https://arxiv.org/abs/2201.11903>, arXiv:2201.11903.
- [13] Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023. URL: <https://arxiv.org/abs/2305.02301>, arXiv:2305.02301.
- [14] Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. Distilling reasoning capabilities into smaller language models, 2023. URL: <https://arxiv.org/abs/2212.00193>, arXiv:2212.00193.
- [15] Google Cloud. Gemini 3 flash, 2026. URL: <https://docs.cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/3-flash>.

- [16] Mistral AI. Model card for mistral-7b-instruct-v0.3, 2024. URL: <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>.
- [17] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL: <https://arxiv.org/abs/2106.09685>, arXiv:2106.09685.
- [18] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics-Doklady*, 1965.

A Appendix

A.1 Gemini Example Prompt, Response, and Commentary

PROMPT:

Here is a puzzle of nested clues that needs to be solved.
 The clues are separated by brackets, so “[“ starts each clue and “]” ends each clue.
 You **must** solve the innermost clues first, and then work outwards.
 A clue is innermost if it does not contain any clues inside of it, so there are no brackets inside of it.
 The final state of the puzzle should be a single sentence without any brackets.

You should pursue multiple potential answers or reasoning paths as you are solving the puzzle.
 For example, you could consider the next few states of the puzzle as you test a chain of potential clue answers.
 However, you must **only** output your **final** answers in the **exact** following format:

```
<clue 1>:<answer>
<clue 2>:<answer>
...
<clue n>:<answer>
final answer:<final answer>
```

The clues and clue answers should not contain any brackets. The clue answers should not contain any emojis.
 Here is an example of what a single clue line should look like:
 dish used for simple haircuts:bowl

Here is the puzzle text: "[in good spirits] [not used] [365 days] [iconic [—fish, like [[Morgan or [[Actor John with a sweet name] [place to get a drink] that's an onomatopoeia]] from 20,000 [American and [[you might call one with [above, in [composer and songwriter Irving]]] rental firm with a [Nader's party] logo], for two] Under the [From – to [Spooky Kubrick classic with 'The'] –]]] with a four-letter name]"

RESPONSE (labeled with clue numbers post-analysis):

```
Clue 1: in good spirits:happy
Clue 2: not used:new
Clue 3: 365 days:year
Clue 4: Actor John with a sweet name:Candy
Clue 5: place to get a drink:bar
Clue 6: composer and songwriter Irving:Berlin
Clue 7: Nader's party:Green
Clue 8: Spooky Kubrick classic with "The":The Shining
Clue 9: Candy bar that's an onomatopoeia:Crunch
Clue 10: above, in Berlin:über
Clue 11: rental firm with a Green logo:National
Clue 12: "From – to The Shining –":Sea
Clue 13: Morgan or Crunch:Captain
Clue 14: you might call one with über:inter
Clue 15: American and inter National, for two:Leagues
Clue 16: Captain from 20,000 Leagues Under the Sea:Nemo
Clue 17: clownfish, like Nemo:clown
Clue 18: iconic clown with a four-letter name:Bozo
final answer:Happy new year Bozo
```

COMMENTARY:

In this puzzle from January 1, 2025, Gemini demonstrates strong factual knowledge of the majority of the clues, and it successfully achieves the correct final answer. Some interesting failures include:

- Clue 8: The answer should be "Shining" rather than "The Shining" because the clue states "with 'The'". This mistake is propagated further into the future clue "From - to The Shining -", as the clue

should actually read "From - to Shining -". Regardless, Gemini recognizes that the phrase should be "From Sea to Shining Sea", but evidently doesn't correct its response in retrospect.

- Clue 11: The clue should actually read "car rental firm with a green logo". "Car" is supposed to be the answer to "you might call one with Uber", but Gemini jumped ahead and became biased to associate the word "inter" with the word "national" in Clue 15, even though the word "inter" was not relevant to any clue in the puzzle.
- Clue 17: This is an example of a self-substitution error, where Gemini replaced the answer, "clown", back into the clue itself. The clue should have a blank dash and read "-fish, like Nemo". As such, the clue was marked as incorrect even though Gemini's solution to the clue was correct.

A.2 Gemini Failure Examples

Composition Failure: An example is provided in the Qualitative Analysis section.

Shortcut Failure: The table below demonstrates a case where Gemini produced the correct final answer, but the majority of its intermediate solutions were incorrect. Gemini initially gets stuck in a small loop with the word "wallflower", but then its next few clues and answers are correct. However, starting at the clue "stereotypically illegible scribble by a doctor", it provides consistently incorrect answers to the rest of the clues, except for the answer "guards". Despite these errors, it produces the correct final answer. This indicates that Gemini is jumping ahead and pattern matching to the final answer without checking its intermediate steps. Contrary to my expectations, the shortcut failure rate across all puzzles was extremely low - 0.5% - indicating that in most cases, Gemini did not actually "cheat" like it did on this puzzle.

Gemini-Produced Clues and Solutions	Reference Clues and Solutions
Wall —:flower — on the flower:wallflower wallflower photographers sometimes demand to hear:cheese cheese full of holes:Swiss illicitly tamper with:doctor sets a time for:schedules org that schedules substances:DEA drug bust, typical DEA operation:drug stereotypically illegible scribble by a doctor:script script drug dosage units, often:grains fastener that typically weighs about 32 grains:pin like some shiny ceramics:enameled Christmas pin that's often enameled:Christmas pin Julius with bangs:Bobby stark naked:stark like Bobby's or starking Christmas pin:apples word you might shout to call security at a apples:Guards final answer:the first 150 Swiss Guards arrive at the Vatican	— on the street:word Wall —:street — naked:buck like some shiny ceramics:glazed word you might shout to call security at a palace:guards sets a time for:schedules cheese full of holes:swiss like Caesar's or Buckingham:palace prescription drug dosage units, often:milligrams word photographers sometimes demand to hear:cheese fastener that typically weighs about 32 milligrams:staple org that schedules substances:DEA stereotypically illegible scribble by a doctor:prescription illicitly tamper with:doctor — bust, typical DEA operation:drug Christmas staple that's often glazed:ham Julius with bangs:Caesar final answer:the first 150 Swiss Guards arrive at the Vatican

Table 4: Gemini Shortcut Failure on Puzzle from January 22, 2025

Self-Substitution Failure: This was the most common type of Gemini's errors. While it does not signify an especially serious mistake, it still caused many clues to be labeled incorrectly and demonstrates that Gemini does not double-check the syntax of its reported answers, which is a crucial skill needed in puzzle-solving. Below are 10 examples of the 427 self-substitution errors that Gemini produced across 45.4% of puzzles:

Gemini-Produced Clues and Solutions	Reference Clues and Solutions
cat-like reflexes:like pixar's "toy story":story to burn the midnight oil:burn spaghetti iconically paired with meatballs:spaghetti cinnamon toast crunch:crunch meet-cute (charming first encounter):meet animal farm where a pig named snowball is based on trotsky:farm a storm or a tidal wave is a natural disaster:disaster the — suspect (prime) politicians often claim to possess common sense:sense	cat— reflexes:like pixar's "toy —":story — the midnight oil:burn pasta iconically paired with meatballs:spaghetti cinnamon toast —:crunch — cute (charming first encounter):meet "animal —" where a pig named snowball is based on trotsky:farm a hurricane or a tidal wave is a natural —:disaster "the — suspect" (the most likely culprit):prime politicians often claim to possess common —:sense

Table 5: Gemini Self-Substitution Failures

A.3 Mistral Task 2 Example

Below are Mistral's raw baseline responses to solving the sets of clues from the puzzle used in Section A.1 (January 1, 2025). Mistral was asked to provide a list of answers for each level of clues without any explanations, and surprisingly, it managed to provide a coherent list for every level. However, it also often provided unwanted, illogical explanations. Additionally, it occasionally produced multiple candidate answers for a single clue in a list, rather than one answer per clue.

LEVEL 0

["pub", "Berlin", "Johnny", "happy", "unused", "year", "The Shining", "Green"]

LEVEL 1

[Snickers, East, Above]

Explanation:

1. Candy bar that's an onomatopoeia: Snickers is a popular candy bar with a name that sounds like the sound of someone eating it. Other examples include Crunch, Kit Kat, and Whatchamacallit. However, Snickers is the only one that is a common onomatopoeia.
2. "From – to shining –": This phrase is often used to describe a place that is clean, polished, or well-maintained. The answer is "East" because it is a direction, and the phrase "From East to shining East" doesn't make sense, but "From East to shining West" does.
3. above, in Berlin: The answer is "Above" because it is a synonym for "over," and the question asks for a location "above" something, not a location in Berlin. However, if we were to interpret the question literally, the answer would be "Berlin," as it is a city located above the ground. But since the question asks for a location "above" something, the answer is "Above."

LEVEL 2

[Uber, Morgan, Crunch]

Explanation:

1. "you might call one with uber" refers to a ride-hailing service, and the answer is "Uber".
2. "Morgan or Crunch" refers to characters from the movie "20,000 Leagues Under the Sea", and the answers are "Morgan" (Captain Nemo) and "Crunch" (the name of the submarine). However, since the question asks for a single answer, we can only return one character, and "Morgan" is a more common name.

LEVEL 3

["Enterprise", "Europcar", "Hertz", "Budget", "Avis"]

Explanation:

1. car rental firm with a green logo: This clue is asking for the names of car rental companies that have a green logo. Some examples include Enterprise, Europcar, Hertz, Budget, and Avis. These companies are known for their green logos, which are often associated with their brands.

LEVEL 4

[USA, Canada]

Explanation:

The innermost clue "American and National, for two" refers to countries that are both American (North American) and have a nationality (citizenship) recognized by the international community. The two countries that fit this description are the United States of America (USA) and Canada.

LEVEL 5

[Nemo]

Explanation:

The innermost clue is "Captain from '20,000 Leagues Under the Sea'". The answer to this clue is Nemo, as he is the captain of the Nautilus submarine in the novel. The outermost clue asks for a four-letter name for an iconic fish that resembles Nemo. However, there is no specific fish character named Nemo in the novel, and the name Nemo itself does not directly refer to a fish. Therefore, it is not possible to solve the entire puzzle based on the given clues.

LEVEL 6

['clown', 'neon', 'angelfish', 'guppy', 'tetra', 'goldfish']

Explanation:

1. —fish, like Nemo: This clue is asking for fish names similar to Nemo, the main character from the movie Finding Nemo. The movie features several types of fish, including clownfish, neon tetra, angelfish, guppy, and goldfish.
2. four-letter name: Since the fish names are all given in the clue, they are all four-letters long.
3. iconic: The puzzle asks for iconic fish, which means famous or well-known fish. The fish mentioned in the clue are all iconic in their own right, as they are commonly recognized and popular.
4. [—fish, like Nemo with a four-letter name]: This means we are looking for four-letter fish names that are similar to Nemo. The fish mentioned in the clue fit this description.
5. [iconic [—fish, like Nemo]]: This means we are looking for iconic fish that are four-letter names similar to Nemo. The fish mentioned in the clue are all iconic and have four-letter names.
6. [iconic [—fish, like Nemo] with a four-letter name]: This means we are looking for iconic fish with four-letter names that are similar to Nemo. The fish mentioned in the clue fit this description.

7. Return the answers in a list: To solve the puzzle, we simply return the answers in a list. In this case, the answers are the four-letter fish names mentioned in the clue.

8. [<answer-1>, <answer-2>, ...]: This is a common way to represent a list of answers in a programming context. In this case, the list would contain the four-letter fish names mentioned in the clue.

9. [clown, neon, angelfish, guppy, tetra, goldfish]: This is the final answer to the puzzle. It is a list of the four-letter fish names mentioned in the clue, in no particular order.

LEVEL 7

[BOBO, KLAATU, PAGLI, CIRCO]

Explanation: 1. iconic clown with a four-letter name: This clue is asking for the names of famous clowns. The four-letter names that come to mind are:

- Bozo (BoBo)
- Klaatu (from the movie The Day the Earth Stood Still)
- Pagliacci (an opera character)
- Circo (a Portuguese word for circus, which is also the name of a famous clown)

These are the four-letter names of iconic clowns that fit the given clue.

Below are Mistral’s post-finetuning responses to the same sets of clues for the same puzzle. Note that there are no unwanted explanations, and Mistral simply provides a list, as was demonstrated in the training examples. Mistral provides exactly 18 answers in total, which equals the total number of clues in the puzzle, and 9 of the 18 answers are correct.

LEVEL 0:

['bar', 'bernstein', 'legend', 'high', 'new', 'year', 'shining', 'consumer']

LEVEL 1:

['crunch', 'sea', 'uber']

LEVEL 2:

['ride', 'apple']

LEVEL 3:

['enterprise']

LEVEL 4:

['high']

LEVEL 5:

['nemo']

LEVEL 6:

['clown']

LEVEL 7:

['bozo']

A.4 Mistral Per-Puzzle and Per-Level Metrics (Tasks 2-5)

The following table provides the statistics for how many of the 50 test puzzles contain each level of clues, and how many total clues exist in each level across all of the puzzles. The number of clues decreases as the level increases, since the puzzle strings are initially long and become shorter as the player solves more clues.

Level	Number of Puzzles	Total Number of Clues
0	50	386
1	50	216
2	50	132
3	45	83
4	34	45
5	20	22
6	11	12
7	5	5

Table 6: Level Statistics for Puzzles 1-50

Task 2: Given partially solved puzzle, solve innermost clues

Observations: Post-finetuning scores are higher than baseline scores for both exact and approximate metrics, but there is a greater increase in exact than approximate scores. For baseline scores, there is greater variability between precision and recall in later levels, but that variability diminishes post-finetuning. Overall, no single level of clues appears to be significantly harder or easier than any other level of clues.

Level	ExactP	ExactR	ExactF1	ApproxP	ApproxR	ApproxF1
0	0.214	0.204	0.208	0.456	0.456	0.455
1	0.265	0.263	0.264	0.452	0.456	0.453
2	0.207	0.235	0.216	0.412	0.435	0.418
3	0.148	0.189	0.161	0.322	0.382	0.343
4	0.197	0.265	0.215	0.328	0.412	0.355
5	0.217	0.325	0.253	0.361	0.488	0.407
6	0.065	0.273	0.096	0.252	0.482	0.314
7	0.240	0.400	0.267	0.455	0.778	0.547

Table 7: Mistral Task 2 Baseline Per-Level Metrics

Level	ExactP	ExactR	ExactF1	ApproxP	ApproxR	ApproxF1
0	0.583	0.574	0.578	0.722	0.707	0.714
1	0.575	0.575	0.575	0.696	0.699	0.697
2	0.622	0.622	0.622	0.706	0.702	0.703
3	0.559	0.559	0.559	0.670	0.663	0.666
4	0.574	0.574	0.574	0.664	0.662	0.662
5	0.825	0.825	0.825	0.871	0.866	0.868
6	0.591	0.591	0.591	0.706	0.702	0.704
7	0.600	0.600	0.600	0.700	0.700	0.700

Table 8: Mistral Task 2 Post-Finetuning Per-Level Metrics

Task 3: Given original puzzle, all past solved clues, and current puzzle, solve current innermost clues

Observations: Similar observations as for Task 2. Certain levels have slight increase or decrease in accuracy in Task 3 as compared to Task 2 but there is no regular pattern. Level 5 clues consistently have a small gap between exact and approximate scores before and after finetuning, which could potentially demonstrate a lower syntactical challenge.

Level	ExactP	ExactR	ExactF1	ApproxP	ApproxR	ApproxF1
0	0.186	0.178	0.182	0.428	0.424	0.425
1	0.132	0.127	0.129	0.348	0.339	0.342
2	0.188	0.198	0.192	0.384	0.384	0.384
3	0.174	0.196	0.181	0.348	0.375	0.356
4	0.265	0.294	0.275	0.363	0.402	0.376
5	0.275	0.350	0.300	0.461	0.566	0.498
6	0.273	0.273	0.273	0.385	0.408	0.392
7	0.200	0.200	0.200	0.300	0.300	0.300

Table 9: Mistral Task 3 Baseline Per-Level Metrics

Task 4: Given previous state of the puzzle, previous level’s solved clues, and current puzzle, solve the current innermost clues

No new observations; there are a few small variations as compared to Tasks 2 and 3 and between the levels but no consistent patterns.

Task 5: Given partially solved puzzle, solve a single innermost clue and provide explanation

Observations: There is quite a large improvement in exact and approximate scores before and after finetuning, and similar to Tasks 2-5, the improvement is larger for exact than approximate. Level 5 demonstrates slightly higher performance than the other levels; however, the overall difficulty of the levels does not consistently increase or decrease as the levels increase.

Level	ExactP	ExactR	ExactF1	ApproxP	ApproxR	ApproxF1
0	0.568	0.565	0.567	0.712	0.707	0.709
1	0.530	0.530	0.530	0.665	0.663	0.663
2	0.577	0.577	0.577	0.691	0.684	0.686
3	0.548	0.548	0.548	0.691	0.684	0.686
4	0.426	0.426	0.426	0.537	0.539	0.538
5	0.625	0.625	0.625	0.750	0.750	0.750
6	0.409	0.409	0.409	0.634	0.630	0.632
7	0.400	0.400	0.400	0.467	0.467	0.467

Table 10: Mistral Task 3 Post-Finetuning Per-Level Metrics

Level	ExactP	ExactR	ExactF1	ApproxP	ApproxR	ApproxF1
0	0.183	0.173	0.177	0.431	0.432	0.431
1	0.143	0.140	0.141	0.336	0.330	0.332
2	0.157	0.158	0.158	0.343	0.342	0.341
3	0.148	0.193	0.163	0.316	0.365	0.333
4	0.175	0.221	0.184	0.332	0.394	0.350
5	0.162	0.225	0.178	0.305	0.397	0.333
6	0.091	0.091	0.091	0.292	0.411	0.333
7	0.400	0.600	0.467	0.510	0.800	0.594

Table 11: Mistral Task 4 Baseline Per-Level Metrics

Level	ExactP	ExactR	ExactF1	ApproxP	ApproxR	ApproxF1
0	0.562	0.551	0.556	0.719	0.705	0.711
1	0.540	0.540	0.540	0.663	0.664	0.663
2	0.597	0.597	0.597	0.681	0.675	0.678
3	0.567	0.567	0.567	0.678	0.679	0.678
4	0.515	0.515	0.515	0.633	0.627	0.629
5	0.725	0.725	0.725	0.791	0.779	0.784
6	0.682	0.682	0.682	0.751	0.747	0.749
7	0.600	0.600	0.600	0.667	0.667	0.667

Table 12: Mistral Task 4 Post-Finetuning Per-Level Metrics

Level	Exact Match	Approx Match	Answer-in-Output
0	0.034	0.124	0.378
1	0.051	0.169	0.389
2	0.045	0.163	0.470
3	0.072	0.225	0.434
4	0.022	0.155	0.422
5	0.136	0.249	0.500
6	0.000	0.171	0.333
7	0.400	0.416	0.600

Table 13: Mistral Task 5 Baseline Per-Level Metrics

Level	Exact Match	Approx Match	Answer-in-Output
0	0.614	0.680	0.655
1	0.606	0.682	0.620
2	0.636	0.702	0.682
3	0.578	0.664	0.614
4	0.644	0.690	0.644
5	0.727	0.762	0.727
6	0.583	0.725	0.583
7	0.400	0.517	0.040

Table 14: Mistral Task 5 Post-Finetuning Per-Level Metrics