Apples, Oranges, and Tennis Balls: Flexible Retrieval Strategies through the Lens of Connections

Anonymous ACL submission

Abstract

Humans exhibit flexible retrieval strategies that allow them to adaptively access different types of semantic knowledge depending on context and goals. In contrast, LLMs struggle with tasks requiring this kind of controlled, adaptive memory access. In this work, we use the NYT Connections puzzle as a testbed for studying flexible retrieval under constraints.

We propose a neuro-symbolic approach to explicitly encourage the LLM to examine words from multiple angles. Our approach significantly outperforms LLM-only baselines, improving average scores by 2-7x across models, enabling models that previously solved 0-5 puzzles to perfectly solve up to 32. We also show that combining smaller, open-source LLMs with symbolic reasoning can outperform larger proprietary models. We make our code and data publicly available.[1]

1 Introduction

The *New York Times* (NYT) Connections puzzle presents a deceptively simple challenge: organize 16 words into four disjoint groups of four, where each group reflects a meaningful concept (e.g., *Types of Fish* or *Words with Silent Letters*) (Liu, 2023a). Solving it requires extensive use of **semantic memory** (Muraki and Pexman, 2024): players need to draw on word meanings, grammar, spelling, pronunciation and world knowledge (Samdarshi et al., 2024). Many puzzles deliberately include plausible "red herring" groupings, forcing players to inhibit the most salient associations (Liu, 2023b).

Figure 1 shows an example puzzle and its solution. Notice that SPAT can also belong to the *Dispute* group, and ZIP – to *Brief Moment*.

Despite its apparent simplicity, Connections has been shown to be challenging for LLMs (Todd et al., 2024; Samdarshi et al., 2024). Recent work found a substantial gap, with the top performing LLM only solving 18% of puzzles, while expert humans were able to solve 60%. In particular, LLMs were shown to perform well on purely semantic relations, but had difficulty with multiword phrases ("to kick the bucket"), morphological reasoning, and encyclopedic categories.

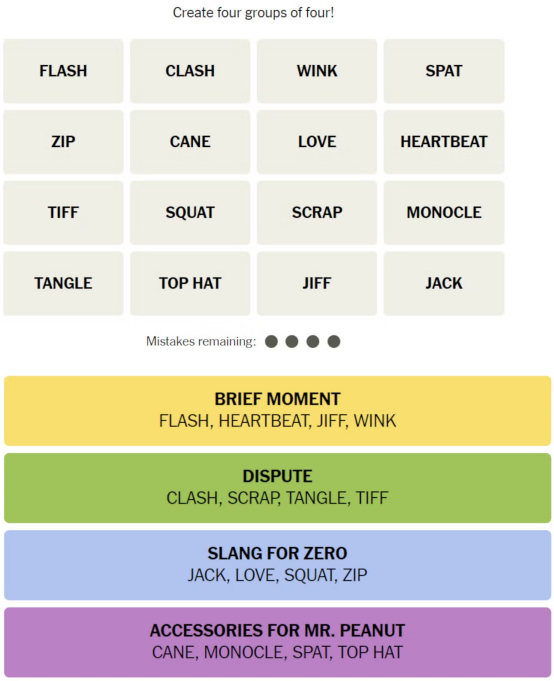

Figure 1: Example NYT Connections puzzle (top) and solution (bottom), reproduced from Smith (2024). Solvers must partition 16 words into four groups of four, where each group shares a common concept. Note that SPAT and ZIP are distractors that could belong to more than one concept, making the puzzle harder.

Connections is a particularly interesting computational challenge because of two properties. First, solving the puzzle requires viewing each word through multiple lenses simultaneously. A solver must consider not just the dominant association of each word, but suppress it when a more task-relevant grouping exists. This mirrors what cog-

[1] URL redacted for anonymity.

nitive psychologists term **flexible retrieval strategies** (Badre and Wagner, 2007; Hoffman et al., 2018): the human ability to dynamically guide memory search depending on context and goals, governed by the brain's *semantic control system* (Jefferies, 2013). LLMs, which rely on learned statistical associations, excel at retrieving frequent associations but fail at the contextual suppression of salient distractors and the creative recombination required to find the correct grouping (Zhang, 2025; Lee et al., 2026; Mizrahi et al., 2025).

The second property that makes Connections interesting is that the solution has a rigid, easily verifiable structure: exactly four disjoint groups of four words each. In other words, the global solution space is logically constrained.

We propose `CoNStruct`, a neuro-symbolic method that leverages both LLM knowledge and external sources to imitate flexible retrieval strategies, and a symbolic constraint satisfaction algorithm to form a valid solution. Our contributions are:

- We propose a neuro-symbolic approach to solve Connections, inspired by cognitive psychology. The Connections game provides a natural testbed due to its flexible retrieval demands and easily verifiable constraint structure.
- Our algorithm, `CoNStruct`, significantly outperforms LLM-only baselines, improving average scores by 2-7x across models, enabling models that previously solved 0-5 puzzles to perfectly solve up to 32 puzzles.
- `CoNStruct` outperforms larger proprietary models like GPT-4o (Hurst et al., 2024) and Claude 3.5 Sonnet (Anthropic, 2024) using much smaller, open-source LLMs such as LLaMA-3.1 8B (Grattafiori et al., 2024).
- We conduct ablation studies demonstrating the individual contribution of each pipeline component, showing that removing any single component consistently reduces performance across all models and metrics.
- We make our code and data available.[1]

We note that `CoNStruct`'s retrieval strategies and constraint-satisfaction procedure are domain-specific, designed around the distinctive properties of Connections. Nevertheless, we hope they can inspire analogous approaches in other domains where similar properties arise.

2 Problem Definition

We now formally define the task. Let $W = \{w_1, w_2, \ldots, w_{16}\}$ be the set of input words. The goal is to produce a set of four disjoint groups $G = \{g_1, g_2, g_3, g_4\}$ such that Each $g_i \subset W$, with $|g_i| = 4$ and $\bigcup_{i=1}^{4} g_i = W$, and each group g_i corresponds to a meaningful, real-world concept.

Unlike the original NYT game, which allows iterative guessing of one group at a time with immediate feedback, our problem formulation assumes the system only has a *single attempt* to solve the game. Naturally, this increases the task's difficulty.

3 Methodology

Overview. Figure 2 provides a high-level overview of `CoNStruct`, our neuro-symbolic system for solving Connections puzzles. The pipeline consists of four main stages: (I) **Concept Generation**, which proposes candidate groupings using both LLMs and external knowledge sources; (II) **Refinement**, which sharpens the concepts; (III) **Constraint Satisfaction**, which selects the best valid solution; and (IV) **Leftover Words**, which handles any leftover words that remain ungrouped. Next we review the pipeline in detail. See Appendix J for prompts.

3.1 Concept Generation

Rather than committing to a single interpretation of the words, the system embeds them in multiple semantic contexts simultaneously. We explicitly targeted retrieval strategies shown to cover $\sim 85\%$ of the Connections puzzle solutions: Semantic Relations, Encyclopedic Knowledge and Multiword Expressions (Samdarshi et al., 2024).

Common Semantic Relations. Semantic Relations are the most popular type of grouping in the Connections puzzles (Samdarshi et al., 2024). The most common examples are words that are synonyms or all fall under a shared hypernym. Because this is an area where LLMs perform relatively well, we simply prompted the model to try and find such groups of 3-5 words.

Encyclopedic Knowledge. Concepts based on encyclopedic knowledge are difficult for LLMs. For example, the LLMs we experimented with often missed the connection between HEADBAND, MULLET, NEON and SPANDEX (*80's fashion trends*). To overcome this issue, we query the Wikipedia API (Wikipedia contributors, 2025a), looking for Wikipedia pages where three words

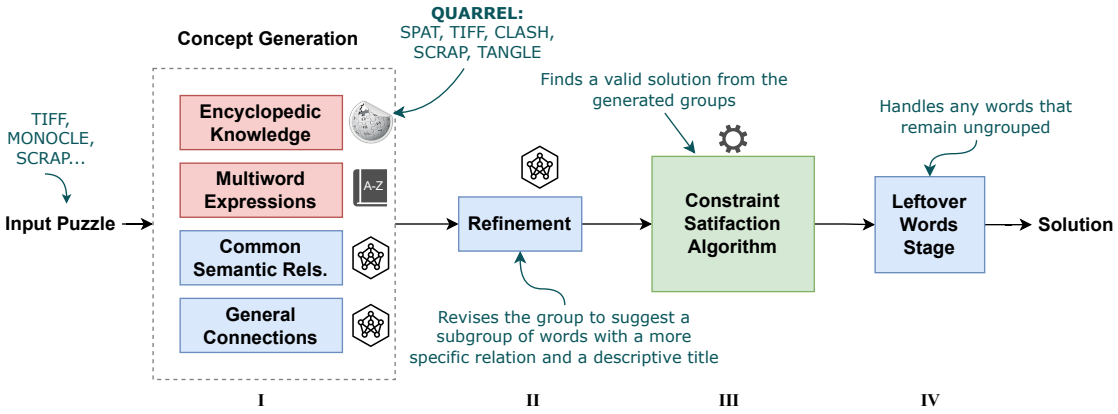

Figure 2: The CoNStruct neuro-symbolic pipeline takes as input 16 words (the puzzle). Candidate groups are formed in the Concept Generation stage using four modules targeting different knowledge types (§3.1). Next, the Refinement stage (§3.2) improves specificity and coherence of groups. The symbolic Constraint Satisfaction Algorithm (§3.3) then identifies a valid, disjoint set of groups. Finally, the Leftover Words stage (§3.4) handles words that were not assigned to a group. The final output consists of four disjoint groups of four words each.

from the puzzle appear. When such a page is found, we query the LLM again, asking which words from the puzzle belong to the concept represented by the title of the page. For example, the words above all appear in the page "1980s in fashion". We note that we have also tried using the Wikipedia *category structure*, but this approach turned out to have significantly less coverage, and thus was left for future work.

Multiword Expressions (MWEs). Multiword expressions are one of the weakest areas for LLMs. In those groups, the words all form a multiword expression when combined with a hidden word (e.g., the words WILD, WALL, SUN, and MAY can all be combined with the word "Flower"). Prior work showed that the best performing LLM was only able to identify 14% of these groups, and the next best one only 4% (Samdarshi et al., 2024).

To address this, we scrape Wiktionary (Wikipedia contributors, 2025b) for MWEs that include the puzzle words, focusing on before/after words. For robustness, we verify the candidate expressions against the Cambridge Dictionary (Cambridge University Press, 2025). The expressions found are grouped with labels such as *"Words that come before the word Flower"*.

General Connections. To capture groups not covered in previous steps, we prompted the LLM to propose as many 3–5 word groupings as possible based on any clear relation it could identify.

Additional Expansion Step. For each of the groups identified so far (except for multi-word expressions), we query the LLM about whether additional words from the puzzle belong in the group (see Figure 3). The idea is to mimic the human strategy of finding a seed set of 2-3 words that share some connection and see if it applies to more words (Aronow and Levine, 2023; Skwarecki, 2024). For self-consistency, we ask the LLM five times and use majority voting (Wang et al., 2023). To prevent hallucinations, we remove all words that were not in the original puzzle. We discard groups that have only 1-2 words after the expansion step. This is an *inhibitory process*, where concepts that fail to expand coherently are discarded.

3.2 Refinement

To combat the LLM's tendency to offer vague, overgeneral concepts, we refine concepts with 4 or more words by prompting the LLM to identify whether they contain a subgroup of ≥ 3 words with a more specific label. For example, in Figure 3, right, the label SLANG TERMS is refined into SLANG TERMS FOR ZERO. These refined groups are then re-expanded (using a self-consistency threshold of 30% to increase coverage), to check whether additional words fit the revised label. We apply refinement to all groups except for MWE groups.

3.3 Constraint Satisfaction

As puzzles regularly involve distractors that can fit multiple groups, greedy approaches, with their early, irreversible commitment, often lead to incorrect solutions. To overcome this, we frame the puzzle as a constraint satisfaction problem, looking

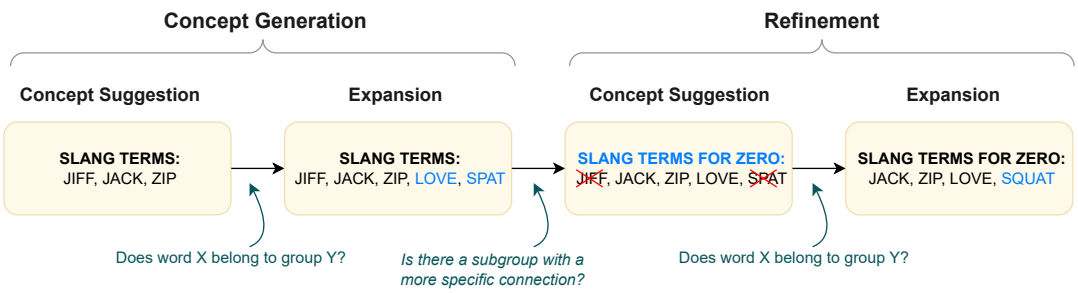

Figure 3: Example of the expansion/refinement steps. Initial *Slang Terms* group is expanded, identifying additional words that may fit. The category is then refined to the more-specific *Slang Terms for Zero*, and words that no longer fit (JIFF and SPAT, crossed out) are removed. A second expansion for the more-specific concept adds SQUAT.

for the largest set of disjoint groups of exactly four words each. See illustration in Appendix C.

We solve this CSP with a recursive backtracking algorithm. The algorithm performs a depth-first search, maintaining a partial solution, expanding via 4-word subsets of candidate concepts that are disjoint with the partial solution (see Appendix B).

We note that the topology of the constraints graph directly affects the size of the search space, and thus this step might be more expensive in domains with looser constraints than Connections.

3.4 Leftover Words Stage

After the constraint-satisfaction stage, some words may remain ungrouped (if the algorithm did not manage to find four groups). If exactly four words are left, we return them as the fourth group. Otherwise, we prompt the LLM to find groups for the leftover words. We then run the constraint satisfaction algorithm again, incorporating the new candidate groups with the previously selected ones.

4 Experimental Setup

We explore the following research questions:

RQ1: Does our neuro-symbolic approach improve LLMs' performance on the NYT Connections puzzle?

RQ2: Can neuro-symbolic systems using smaller open-source LLMs outperform much larger proprietary models on Connections?

RQ3: What is the individual contribution of each component in our system?

Dataset. We evaluate `CoNStruct` using the dataset created by Samdarshi et al. (2024), containing 441 Connections puzzles collected between June 2023-August 2024. Each puzzle includes a ground-truth solution, where each group is also annotated with the type of knowledge required to identify it; annotations were created by linguists.

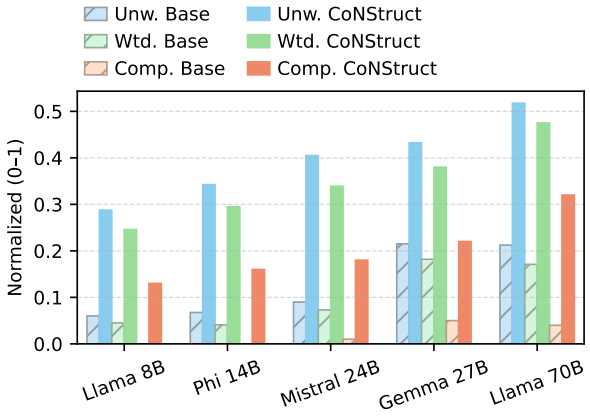

Figure 4: Normalized performance gains from applying `CoNStruct` across 5 language models, showing unweighted score, weighted score, and number of completely solved puzzles. Scores are normalized (1=maximum possible: 4, 10, 100, respectively). `CoNStruct` (solid bars) consistently improves performance across all models and metrics relative to the baseline (hatched bars). See Appendix D for separate visualizations.

For the evaluation of `CoNStruct`, we use 100 puzzles from this dataset (ids 300-400), spanning from April 7, 2024 to July 15, 2024.

Baseline. As a baseline, we take the prompt of Samdarshi et al. (2024), containing instructions as presented to players on the NYT website and three examples (see Appendix J.1).

Models. To ensure our results are consistent across models we use five LLMs of varying sizes: LLaMA-3.1 8B (Grattafiori et al., 2024), Phi-4 14B (Abdin et al., 2024), Mistral-Small-3.1 24B (Mistral AI, 2025), Gemma-3 27B (Team et al., 2025), and LLaMA-3.1 70B (Grattafiori et al., 2024). When evaluating the baseline, the model parameters (e.g., temperature, top-p) are set to the default,

consistent with Samdarshi et al. (2024). When evaluating CoNStruct, the temperature value is set to 0.2 and the maximum number of output tokens is set to 2000. Other parameters are set to the default.

Evaluation Metrics. We evaluate performance using three metrics, reflecting different granularities.

- **Unweighted Score.** This metric measures how many of the model's predicted groups exactly match the ground-truth – i.e., all four words match all four words of one of the ground-truth concepts. Each ground-truth group can only count once and the maximum possible score per puzzle is 4. We report the average across all puzzles.
- **Weighted Score.** This score accounts for the relative difficulty of each group, based on the NYT's color-coded system. Each group has an associated weight: Yellow = 1, Green = 2, Blue = 3, Purple = 4. The score is the sum of weights of the concepts that were exactly matched. The maximum possible weighted score is 10 (when all four groups are correct). We report the average across all puzzles.
- **Completely Solved Puzzles.** The number of puzzles where all four predicted concepts are correct, i.e., the unweighted score is 4.

See further experimental details in Appendix A.

5 Results

We experimented with LLaMA-3.1 8B, Phi-4 14B, Mistral-Small-3.1 24B, Gemma-3 27B, and LLaMA-3.1 70B. For each model, we tested both (1) the baseline prompt (Section 4) and (2) our pipeline using the model.

5.1 RQ1: Performance Improvement

Figure 4 shows the combined results for all three metrics, normalized by the maximum possible score of each metric. See Appendix D for separate (unnormalized) figures for each metric.

Unweighted Score. Augmenting the LLM with our pipeline resulted in substantial improvements in the average number of correct groups per puzzle across all models. The largest model, LLaMA-3.1 70B, increased its score from 0.85 to 2.07 (max possible score is 4.0). Smaller models such as Phi-4 14B and Mistral-3.1 24B exhibited the largest relative gains, improving by +1.1 and +1.26 groups, respectively, an increase of 4-5x.

Weighted Score. All models also demonstrated substantial improvements in the average *weighted* score when using CoNStruct. Even the strongest model, LLaMA-3.1 70B, nearly triples its score. The largest relative gains again occur with smaller models (4-7x). The results suggest that CoNStruct not only improves overall performance, but also performance on the harder concepts.

Model	Unweighted	Weigh.	Comp.
Gemini 1.5 Pro	0.21	0.17	0.04
Claude 3.5 Sonn.	0.32	0.27	0.11
GPT-4o	0.29	0.23	0.06
LLaMA 3.1 405B	0.24	0.18	0.03
Mistral Large 2	0.21	0.16	0.04

Table 1: Performance of SOTA LLMs, reproduced from Samdarshi et al. (2024). Scores are normalized by the maximum possible value for each metric (average unweighted score/4, average weighted score/10, and completely solved puzzles/100). CoNStruct surpasses these results using much smaller open-source LLMs.

Completely Solved Puzzles. The most pronounced improvement was in the number of fully solved puzzles. Under the baseline, models solved completely only between 0-5 puzzles. CoNStruct increased this count to 13–22 puzzles for the smaller models, and 32 for LLaMA-3.1 70B.

To summarize, our approach resulted in **significant improvements across the board**.

5.2 RQ2: Comparison to Larger Models

To answer RQ2, we compared our results with the performance of the models evaluated by Samdarshi et al. (2024) (the LLM-only approach used as our baseline). Their evaluation included larger models than the ones we used: Claude 3.5 Sonnet (Anthropic, 2024), GPT-4o (Hurst et al., 2024), Gemini 1.5 Pro (Team et al., 2024), LLaMA-3.1 405B (Grattafiori et al., 2024) and Mistral Large 2 (Mistral AI team, 2024). We used the results of their study reported on **the same** 100 puzzles we experimented on. The results are shown in Table 1. CoNStruct, using much smaller models such as Phi-4 and Mistral-3.1-small, **outperforms all of these models in every metric**. Notably, even our smallest model, LLaMA-3.1 8B, surpasses all models by fully solving 13 puzzles.

5.3 RQ3: Ablations

We performed ablation studies to measure the contribution of different parts of the pipeline. We present results for the pipeline using LLaMA-3.1 70B in Table 2. Results for other models are consistent, and are provided in Appendix F.

Metric	Full Pipeline	No Leftover Words	No Constraint Sat.†	No Refinement†
Unweighted Score	2.07	1.79	1.52	1.21
Weighted Score	4.75	3.77	3.10	2.55
Completely Solved	32	7 (+25)*	3 (+17)*	10 (+5)*

Table 2: Ablation results on LLaMA-3.1 70B showing the impact of removing Leftover Words handling, the Constraint Satisfaction Algorithm, or Refinement. Removing any component reduces performance across all metrics. *Numbers in parentheses indicate additional puzzles effectively solved with 3 out of 4 correct groups, see Note. †The Leftover Words stage is also omitted when Constraint Satisfaction or Refinement is ablated, to better isolate the effect of those components without the effect of adding new candidates.

> **Note.** Sometimes, after the Constraint Satisfaction stage (Section 3.3), `CoNStruct` identifies three out of four correct groups. These puzzles are effectively solved, as the remaining four words can be trivially grouped without LLM inference (and indeed, this is what happens in the Leftover Words phase, Section 3.4). In ablations, where we omit the Leftover Words phase, we report both the number of completely solved puzzles (four groups correctly identified) and the number of *effectively* completely solved puzzles (three groups correctly identified).

Leftover Words Stage. This stage is designed to complete partial solutions. As shown in Table 2, removing it leads to a drop in performance.

Constraint Satisfaction. To isolate the contribution of the constraint satisfaction component, we replace it with a simpler greedy selection algorithm that selects up to four non-overlapping groups of exactly four words each from the list of candidate groups. Removing the constraint satisfaction algorithm significantly reduces performance. The average number of completely solved puzzles drops from 7 (plus 25 effectively solved puzzles with 3 correct groups) to just 3 (plus 17 effectively solved puzzles), with corresponding drops in both unweighted and weighted scores.

Refinement. To assess the impact of the refinement stage in our pipeline, we conduct an ablation where this stage is entirely removed, and we directly pass the unrefined groupings to the constraint satisfaction algorithm. We remove all groups with more than 10 words due to computational complexity. As shown in Table 2, this has a substantial impact on all metrics, which drop notably.

Concept Generation. To evaluate the contribution of each concept generation module in `CoNStruct`, we traced back each correct group in the final prediction to its source.

For each such correct group, we search the initial suggested groups from each module for any that contain 3-5 words and overlap with the final group on at least 3 out of its 4 words. We define

- **Partial Credit:** If a final group matches n sources, each receives partial ($\frac{1}{n}$) credit.
- **Unique Credit:** A source receives credit if it is the only one to match a correct group.

As shown in Table 3, each source contributes meaningfully to the final groups. The semantic and general modules have the highest number of matches, while the multiword expression module, though contributing fewer matches, is crucial for capturing specific group types that other sources miss. A closer look into source overlap patterns (proportion of matches from source A also covered by source B, see Appendix F.1) corroborates this observation.

Taken together, these results demonstrate that each source contributes uniquely to the overall performance of `CoNStruct`. External resources (wikipedia and dictionary) are not the primary drivers of `CoNStruct`'s performance, although they still improve coverage and accuracy; these findings validate our choice to include all four strategies.

Metric	Partial Credit	Unique Credit
MWE	9.83	8
Encyclopedic	46.83	8
Semantic	57.17	8
General	62.17	10

Table 3: Attribution of correct final groups to each concept generation module. The Semantic and General modules had the broadest overall impact, while the MWE module, despite contributing to fewer groups overall, was uniquely responsible for 8 groups.

6 Further Analysis

Analysis in this section is performed on the same 100 puzzles used in our main evaluation.

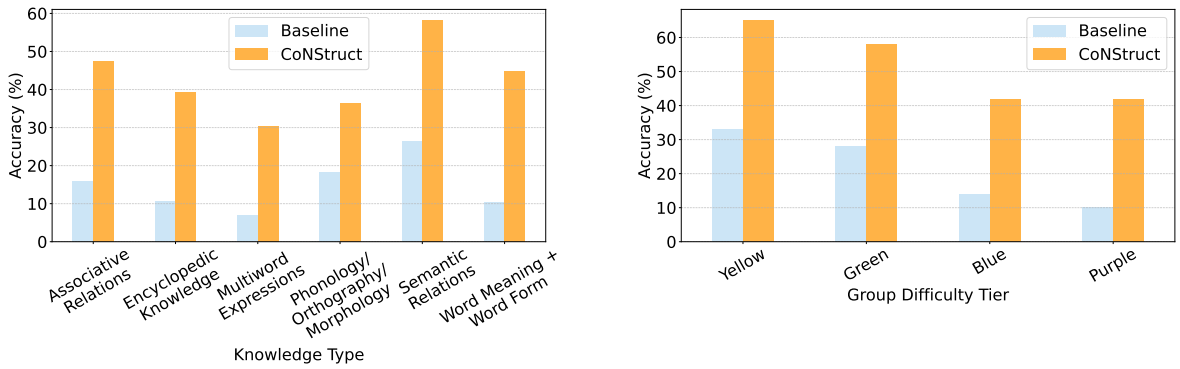

Figure 5: Comparing % correctly identified groups, baseline vs. CoNStruct, using LLaMA-3.1 70B. Left: Accuracy across different knowledge types. Results show consistent improvements across all knowledge types. Right: Accuracy across different difficulty tiers, from Yellow (easiest) through Purple (hardest). Results show consistent improvements across all difficulty levels, with particularly large gains in the more challenging Blue and Purple tiers.

Different Knowledge Types. Samdarshi et al. (2024) categorized the knowledge types required for the Connections puzzles (see taxonomy in Appendix I). Each group in the dataset is annotated with a knowledge type, allowing us to evaluate accuracy across different types. Accuracy is defined as the number of correctly predicted groups divided by the number of ground-truth groups of that type.

Figure 5 (left) shows results for the LLaMA-3.1 70B model; see results for other models in Appendix E. As shown, CoNStruct **outperforms the baseline on every knowledge type**.

Difficulty Tiers. We analyze CoNStruct's accuracy across the four difficulty tiers used by the NYT: Yellow (easiest), Green, Blue, and Purple (hardest). Each puzzle contains exactly one group from each tier. We measure accuracy for each difficulty tier for both the baseline and CoNStruct.

Figure 5 (right) shows results for LLaMA-3.1 70B (see other models in Appendix E). CoNStruct **significantly improves performance at every tier**. Accuracy rises from 33% to 65% on the easiest tier and from 10% to 42% on the hardest. Notably, the largest relative gains are in the harder tiers.

Comparison to Human Mistakes. To better understand where CoNStruct fails we analyze 30 puzzles and identify common error sources: failure to identify the correct word **sense** (52.9%), falling for a **distractor** (17.1%), correct concept identified but lost due to incorrect **refinement** (14.3%) or in the **constraint satisfaction** because of a wrong solution with the same number of disjoint 4-word groups (10%) or even larger (5.7%).

We next test whether CoNStruct success aligns with human difficulty. We obtain human performance statistics for these puzzles from the New York Times *Connections Bot* (Giratikanon, 2024), an official analysis tool that reports aggregate player outcomes and common mistakes.

We observe a weak positive association between human solve rate and CoNStruct unweighted score (Pearson $r = 0.28$, Spearman $\rho = 0.283$). Categories found by CoNStruct have a higher human success rate (mean 83.64% compared to 72.63%). This difference is statistically significant (Welch's two-sided t-test yields $t = 3.9$, $p =$ 1.5e-4), indicating that categories that CoNStruct found tended to be categories that humans solve more frequently.

Interestingly, ∼33% of the top-3 human mistakes were due to distractors. While there is a substantial overlap between mistakes of CoNStruct and humans, CoNStruct's errors are often not the most frequent one humans make (see Appendix G). Results might be affected by the iterative gameplay of humans, allowing intermediate guesses.

Stability Analysis. Due to budget constraints, instead of multiple runs we chose to test our approach across 5 models of varying sizes and architectures. As a quick assessment of the robustness of our results, we performed five more independent runs on the best-performing LLaMA-3.1 70B with identical settings to the original run. Normalizing by the maximum possible scores yields 0.515 ± 0.075 (unweighted metric) and 0.465 ± 0.088 (weighted metric), where $\pm$ denotes the between-run std, corroborating the stability of the results.

Label Correspondence with NYT Labels. While CoNStruct is evaluated solely on whether it predicts the correct words in each group, and not whether its rationale matches the NYT's, we exa-

mind how semantically aligned CoNStruct's group labels are with the official NYT labels. See Appendix H for details; we find that CoNStruct's labels are well above a random baseline and reach approximately 64% of the upper bound, suggesting that its rationales are generally meaningful.

7 Related Work

LLMs and the NYT Connections Puzzle. Recent work by Todd et al. (2024) and Samdarshi et al. (2024) shows that even advanced LLMs struggle with the Connections puzzle. Both studies highlight consistent weaknesses in handling multi-word expressions, form-based categories (phonology, morphology, etc.), and categories requiring abstract reasoning. They also find that red herrings often mislead LLMs. Our work builds directly on these findings by targeting challenging knowledge types and addressing the red herring problem through symbolic reasoning.

LLMs and Concepts. Research on how LLMs represent concepts has revealed key differences from human cognition. Shani et al. (2023) found that LLMs often violate core organizational principles such as asymmetry, transitivity, and property inheritance. In a follow-up work, Shani et al. (2025) showed that LLMs also struggle with fine-grained semantic distinctions and item typicality. In this paper, we use the Connections puzzle as a testbed for examining how LLMs can approximate human-like concept formation.

Neuro-symbolic Approaches. Neuro-symbolic approaches combine the flexible learning capabilities of neural networks with the structured reasoning and interpretability of symbolic representations, with the goal of building AI systems that are semantically grounded, explainable, and reliable (Garcez and Lamb, 2023). These approaches have been successfully applied to tasks such as first-order logical reasoning (Mittal et al., 2024), long-term planning (Wu and Mitra, 2024), abstraction (Bober-Irizar and Banerjee, 2024; Butt et al., 2024) and constrained text generation (Régin et al., 2024). To the best of our knowledge, no prior work has applied a neuro-symbolic framework to Connections.

Cognitive Inspiration. In designing our method, we were explicitly inspired by cognitive theories. In particular, **Inhibition Theory** (Anderson et al., 1994; Anderson, 2003) posits that flexible retrieval is achieved by actively suppressing currently irrelevant or competing memory representations. **Semantic Contextualization Theory**, on the other hand, posits that flexible retrieval (Tulving and Thomson, 1973; Barsalou, 1982) is achieved by restructuring the semantic context in which retrieval occurs, rather than suppressing competing memories directly.

Our system reflects a particular balance, chosen for practical effectiveness rather than theoretical completeness. An interesting direction for future work is to alter the pipeline to mimic different cognitive theories, exploring their trade-offs.

8 Conclusions and Future Work

LLMs struggle with tasks requiring flexible retrieval strategies that humans possess. In this work, we use the NYT Connections puzzle as a window into how humans flexibly form concepts through controlled, adaptive memory access.

Our neuro-symbolic approach, CoNStruct, integrates LLMs with symbolic reasoning and external knowledge. On a benchmark of 100 puzzles, CoNStruct substantially improves performance over LLM-only approaches across all models and metrics. CoNStruct outperforms large proprietary models using much smaller, open-source LLMs.

In the future, we plan to explore heuristic-based search to optimize the selection strategy, including implementing a mechanism for assessing concept strength, as well as expand to more strategies (e.g., phonology). The framework also holds potential for cognitive science: A particularly interesting direction is to alter the pipeline to explicitly mimic different cognitive theories, exploring their trade-offs. Another direction is *generating* puzzles, to help cognitive scientists create large sets of controlled semantic tasks or vary difficulty systematically, e.g., emphasizing certain types of retrieval for studying human strategies.

Beyond the puzzle of Connections, CoNStruct provides a **blueprint** for building neuro-symbolic systems for other domains where flexible retrieval strategies are useful. We hope our results encourage further research in this direction.

9 Limitations

- Our evaluation focuses solely on accuracy, meaning whether the groupings match the actual solution. It does not take into account the reasoning of the system behind each grouping and whether it matches the ground truth rationale.
- We have only tested our algorithm on English. Different languages might have less resources, and might require adapting the algorithm.
- Our ablation studies for the concept generation modules are based on an approximation of the influence of each module. We did not run the entire pipeline again for each module for more exact ablation results due to cost. We recognize that this is an approximation.

10 Ethical Considerations

Bias in LLMs. This paper focuses on improving the capabilities of LLMs in solving a word game from the NYT – a relatively low-risk domain. However, we acknowledge that LLMs can reflect and amplify biases present in their training data, and any application of such models should be mindful of these potential biases.

Use of AI Assistants. We used GPT-4o to assist with coding, writing, and rephrasing. All AI-generated outputs were reviewed and edited to ensure they aligned with our goals and accurately reflected our original intent.

Use of Scientific Artifacts. All models and datasets used in this work are consistent with their intended use and licensing terms. We use the dataset released by Samdarshi et al. (2024), which is publicly available at: `https://github.com/mustafamariam/LLM-Connections-Solver`. While the repository does not include a formal license, the authors request citation for use of their data and annotations. We comply with this request and cite their work appropriately. We do not redistribute or modify their dataset. We additionally obtain aggregate human-performance statistics (e.g., solve rate and common incorrect groupings) from the New York Times *Connections Bot* (Giratikanon, 2024), an official companion analysis tool for the game. We use only puzzle-level and category-level aggregates reported by the Bot and do not collect, access, or store any individual user data. We also use open-source language models including LLaMA-3.1 (8B and 70B) (Grattafiori et al., 2024), Phi-4 14B (Abdin et al., 2024), Mistral-Small-3.1 24B (Mistral AI, 2025), and Gemma-3 27B (Team et al., 2025), all of which are licensed for research (under LLaMA Community License, Apache 2.0 or Gemma 3 License). We use the Wikipedia and Wiktionary API (both under CC BY-SA). We use the Cambridge Dictionary API to look up expressions in accordance with their Terms of Use. We will release our code and outputs as open-source (redacted for anonymity).

Use of Dataset. The dataset used contains no personal or identifying information; all content consists of generic word lists. We manually reviewed samples to ensure no offensive or sensitive content was introduced. No user data was collected, and no privacy risks were identified.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Michael C Anderson. 2003. Rethinking interference theory: Executive control and the mechanisms of forgetting. *Journal of memory and language*, 49(4):415–445.

Michael C Anderson, Robert A Bjork, and Elizabeth L Bjork. 1994. Remembering can cause forgetting: retrieval dynamics in long-term memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20(5):1063.

Anthropic. 2024. Claude 3.5 sonnet. Accessed July 28, 2025.

Isaac Aronow and Elie Levine. 2023. How to line up a great connections solve. Accessed July 28, 2025.

David Badre and Anthony D Wagner. 2007. Left ventrolateral prefrontal cortex and the cognitive control of memory. *Neuropsychologia*, 45(13):2883–2901.

Lawrence W Barsalou. 1982. Context-independent and context-dependent information in concepts. *Memory & cognition*, 10(1):82–93.

Mikel Bober-Irizar and Soumya Banerjee. 2024. Neural networks for abstraction and reasoning. *Scientific Reports*, 14(1):27823.

Natasha Butt, Blazej Manczak, Auke Wiggers, Corrado Rainone, David W Zhang, Michaël Defferrard, and Taco Cohen. 2024. Codeit: Self-improving language models with prioritized hindsight replay. *arXiv preprint arXiv:2402.04858*.

Cambridge University Press. 2025. Cambridge dictionary. Accessed July 28, 2025.

DeepInfra. 2025. Deepinfra api. Accessed July 28, 2025.

Artur d'Avila Garcez and Luis C Lamb. 2023. Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 56(11):12387–12406.

Tom Giratikanon. 2024. Connections bot f.a.q. Accessed January 4, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Paul Hoffman, James L McClelland, and Matthew A Lambon Ralph. 2018. Concepts, control, and context: A connectionist account of normal and disordered semantic cognition. *Psychological review*, 125(3):293.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Elizabeth Jefferies. 2013. The neural basis of semantic cognition: converging evidence from neuropsychology, neuroimaging and tms. *Cortex*, 49(3):611–625.

Seongyun Lee, Yongrae Jo, Minju Seo, Moontae Lee, and Minjoon Seo. 2026. Lost in the noise: How reasoning models fail with contextual distractors. *arXiv preprint arXiv:2601.07226*.

Wyna Liu. 2023a. Connections - how to play. Accessed July 28, 2025.

Wyna Liu. 2023b. How our new game, connections, is put together. Accessed July 28, 2025.

Mistral AI. 2025. Mistral small 3.1. Accessed July 28, 2025.

Mistral AI team. 2024. Large enough. Accessed July 28, 2025.

Chinmay Mittal, Krishna Kartik, Parag Singla, and 1 others. 2024. Fcorebench: Can large language models solve challenging first-order combinatorial reasoning problems? *arXiv preprint arXiv:2402.02611*.

Moran Mizrahi, Chen Shani, Gabriel Stanovsky, Dan Jurafsky, and Dafna Shahaf. 2025. Cooking up creativity: Enhancing llm creativity through structured recombination. *arXiv preprint arXiv:2504.20643*.

Emiko Muraki and Penny Pexman. 2024. Nyt connections: Tips to improve your game through the science of semantic memory. Accessed July 28, 2025.

Florian Régin, Elisabetta De Maria, and Alexandre Bonlarron. 2024. Combining constraint programming reasoning with large language model predictions. *arXiv preprint arXiv:2407.13490*.

Prisha Samdarshi, Mariam Mustafa, Anushka Kulkarni, Raven Rothkopf, Tuhin Chakrabarty, and Smaranda Muresan. 2024. Connecting the dots: Evaluating abstract reasoning capabilities of llms using the new york times connections word game. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21219–21236.

Chen Shani, Dan Jurafsky, Yann LeCun, and Ravid Shwartz-Ziv. 2025. From tokens to thoughts: How llms and humans trade compression for meaning. *arXiv preprint arXiv:2505.17117*.

Chen Shani, Jilles Vreeken, and Dafna Shahaf. 2023. Towards concept-aware large language models. *arXiv preprint arXiv:2311.01866*.

Beth Skwarecki. 2024. Why you keep losing nyt connections. Accessed July 28, 2025.

Mina Smith. 2024. New york times connections hints and answers for #302 april 8, 2024. Accessed July 28, 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Graham Todd, Tim Merino, Sam Earle, and Julian Togelius. 2024. Missed connections: Lateral thinking puzzles for large language models. In *2024 IEEE Conference on Games (CoG)*, pages 1–8. IEEE.

Endel Tulving and Donald M Thomson. 1973. Encoding specificity and retrieval processes in episodic memory. *Psychological review*, 80(5):352.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Wikipedia contributors. 2025a. Wikipedia, the free encyclopedia. Accessed July 28, 2025.

Wikipedia contributors. 2025b. Wiktionary, free dictionary. Accessed July 28, 2025.

Erik Wu and Sayan Mitra. 2024. Can llms plan paths with extra hints from solvers? *arXiv preprint arXiv:2410.05045*.

Zheng Zhang. 2025. Comprehension without competence: Architectural limits of llms in symbolic computation and reasoning. *arXiv preprint arXiv:2507.10624*.

A Experimental Details

A.1 Hyper-parameter Search Details

For the `CoNStruct` pipeline, we set temperature to 0.2 and maximum output tokens to 2000 for all LLM calls. No hyperparameter search was performed, the temperature was set to 0.2 to reduce randomness, and maximum output tokens was set to 2000 as a precautionary limit to avoid unnecessarily long generations. Due to budget constraints, instead of multiple runs we chose to test our approach across 5 models of varying sizes and architectures. We provide a variance estimate on a subset of puzzles to show stability in Section 6.

We experimented with different thresholds for the self-consistency vote (Wang et al., 2023) after the refinement stage (number of "yes" responses required to add a word during group expansion). On a held-out set using LLaMA-3.1 8B, we tested 3 options - 20% threshold (i.e., 1 vote out of 5), 30% threshold (i.e., 2 votes out of 5), and 50% threshold (i.e., 3 votes out of 5). We found that a 30% threshold yielded the best results and used that across models.

A.2 Computational Budget

We used DeepInfra's API (DeepInfra, 2025) to experiment with the different models. The total cost of our development was 70 dollars. Using LLaMA-3.1 70B, the most expensive model we used, the average cost per puzzle was approximately $0.2.

B Pseudocode for Constraint Satisfaction Algorithm

Algorithm 1 Backtracking Constraint Solver

```
1: function BACKTRACKING(groups, used_words, final_groups, best_groups)
2:     if groups is empty then
3:         if size of final_groups > size of best_groups then
4:             best_groups ← copy of final_groups
5:         end if
6:         return
7:     end if
8:     current_group ← first element of groups
9:     remaining_groups ← rest of groups
10:    for all combination of 4 words in current_group do
11:        combination_set ← set of words in combination
12:        if combination_set ∩ used_words is empty then
13:            new_used_words ← used_words ∪ combination_set
14:            new_final_groups ← final_groups + combination
15:            BACKTRACKING(remaining_groups, new_used_words, new_final_groups, best_groups)
16:        end if
17:    end for
18:    BACKTRACKING(remaining_groups, used_words, final_groups, best_groups)
19: end function
```

The algorithm performs a depth-first search over a decision tree in which, for each candidate group (current_group), it branches on (i) including a 4-word subset from this group that is disjoint from the already used words (line 15), or (ii) skipping this candidate group entirely (line 18). Each recursive call to the Backtracking function takes four arguments: The remaining candidate groups, the set of used words in the current partial solution (selected groups so far), the current partial solution itself, and the best solution found so far.

C Constraint Satisfaction Stage Figure

Figure 6 illustrates the Constraint Satisfaction stage. On the left, we see the initial noisy set of candidate groups, where many words appear in multiple groupings. Groups with less than 4 words are removed. In the middle, the constraint satisfaction algorithm resolves conflicts between groups with overlapping words. On the right, it produces a valid solution composed of disjoint groups of four words. In this case, it managed to find all four of them – a perfect solution.

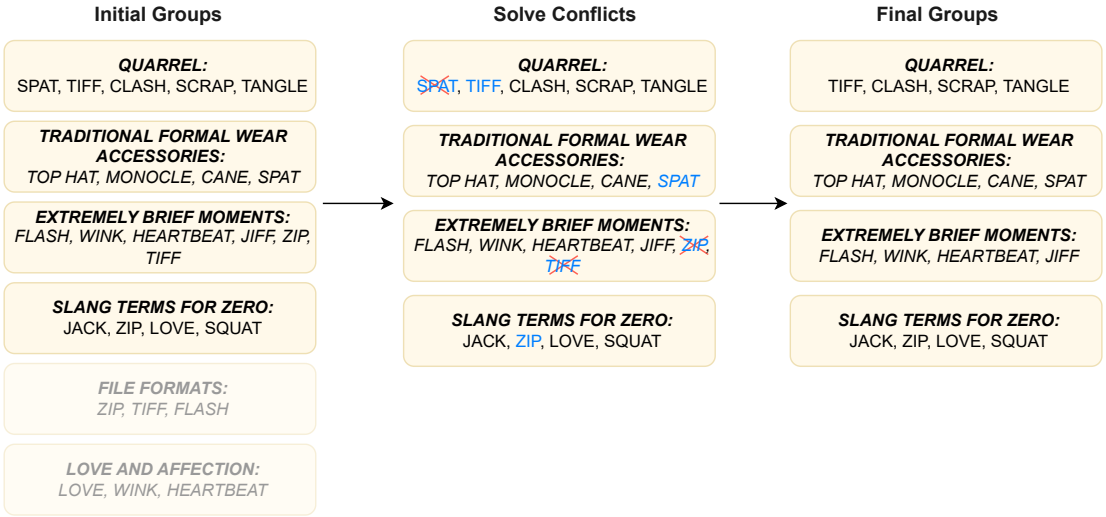

Figure 6: The constraint satisfaction process used to select a valid solution from a noisy pool of candidate groups. **Left:** Initial set of group candidates. Many words appear in multiple groups (e.g., ZIP, TIFF, SPAT). Groups with less than 4 words (e.g., *File Formats*, *Love and Affection*), are removed before this stage. **Middle:** Identifying conflicts (blue words) and finding the largest number of disjoint 4-word groups. **Right:** The final output.

D Results on each Metric Separated

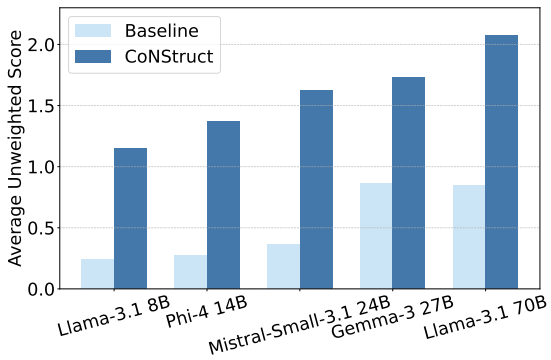

Figure 7: Average unweighted score across five LLMs, comparing the baseline prompting strategy (light bars) with `CoNStruct` (dark bars). All models show substantial improvement using our method. Smaller models like Phi-4 and Mistral-Small-3.1 benefit most, improving from 0.27 to 1.37 and 0.36 to 1.62 groups on average, respectively. Even the largest model, LLaMA-3.1 70B, more than doubles its performance (0.85 to 2.07).

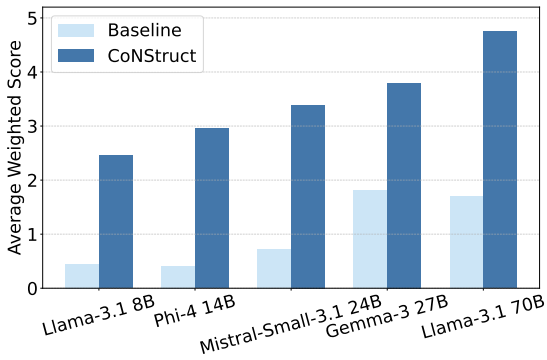

Figure 8: Average weighted score across five LLMs, comparing the baseline prompting strategy (light bars) with `CoNStruct` (dark bars). The weighted score reflects both the number and difficulty of groups solved. All models see significant gains with `CoNStruct`. Smaller models like LLaMA-3.1 8B and Phi-4 improve by 5x and 7x respectively. The strongest model, LLaMA-3.1 70B, increases from 1.71 to 4.75, indicating substantial improvements in solving even the most challenging groups.

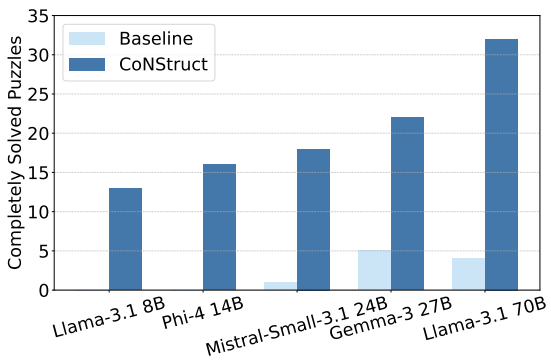

Figure 9: Number of completely solved puzzles (i.e., all four groups correctly identified) across five LLMs, comparing the baseline prompting strategy (light bars) with `CoNStruct` (dark bars). `CoNStruct` significantly increases the number of completely solved puzzles across all models. Notably, LLaMA-3.1 70B solves 32 puzzles compared to 4 in the baseline, and even smaller models like Phi-4 and LLaMA-3.1 8B, which solved 0 puzzles under the baseline, achieve 16 and 13 completely solved puzzles respectively.

E Results of Accuracy by Knowledge Types and Difficulty Tiers for each Model

E.1 LLaMA-3.1 8B Model

Results for the LLaMA-3.1 8B Model can be seen in Figure 10

E.2 Phi-4 14B Model

Results for the Phi-4 14B Model can be seen in Figure 11

E.3 Mistral-3.1-Small 24B Model

Results for the Mistral-3.1-Small 24B Model can be seen in Figure 12

E.4 Gemma-3 27B Model

Results for the Gemma-3 27B Model can be seen in Figure 13

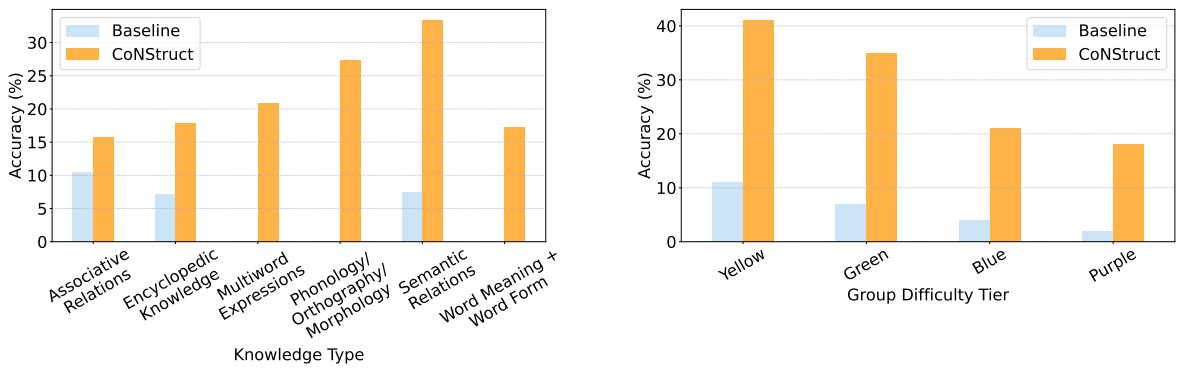

Figure 10: Comparing the baseline (blue) with `CoNStruct` (orange), using the LLaMA-3.1 8B model. Left: Accuracy across different knowledge types. y-axis: percentage of correctly identified groups within a given knowledge type. Results show consistent improvements across all knowledge types. Right: Accuracy across different difficulty tiers. y-axis: percentage of correctly identified groups within each tier: Yellow (easiest) through Purple (hardest). Results show consistent improvements across all difficulty levels.

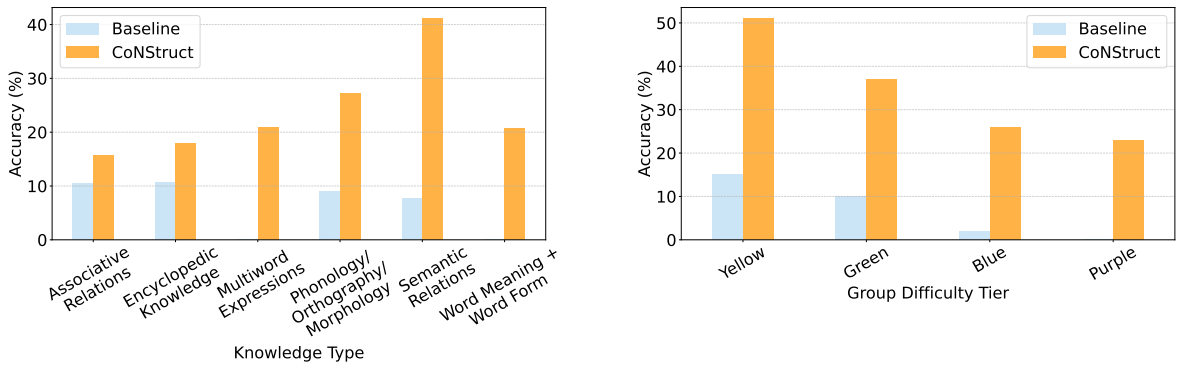

Figure 11: Comparing the baseline (blue) with `CoNStruct` (orange), using the Phi-4 14B model. Left: Accuracy across different knowledge types. y-axis: percentage of correctly identified groups within a given knowledge type. Results show consistent improvements across all knowledge types. Right: Accuracy across different difficulty tiers. y-axis: percentage of correctly identified groups within each tier: Yellow (easiest) through Purple (hardest). Results show consistent improvements across all difficulty levels, with particularly large gains in the more challenging Blue and Purple tiers.

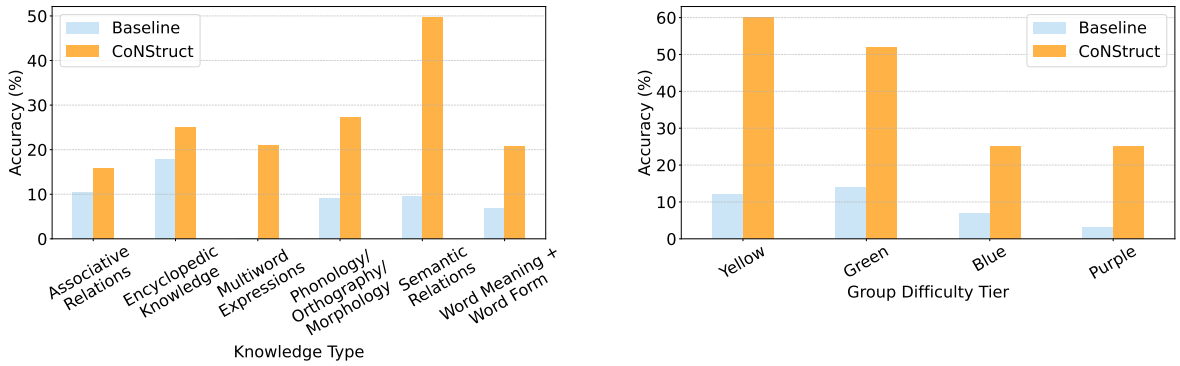

Figure 12: Comparing the baseline (blue) with `CoNStruct` (orange), using the Mistral-3.1-Small 24B model. Left: Accuracy across different knowledge types. y-axis: percentage of correctly identified groups within a given knowledge type. Results show consistent improvements across all knowledge types. Right: Accuracy across different difficulty tiers. y-axis: percentage of correctly identified groups within each tier: Yellow (easiest) through Purple (hardest). Results show consistent improvements across all difficulty levels.

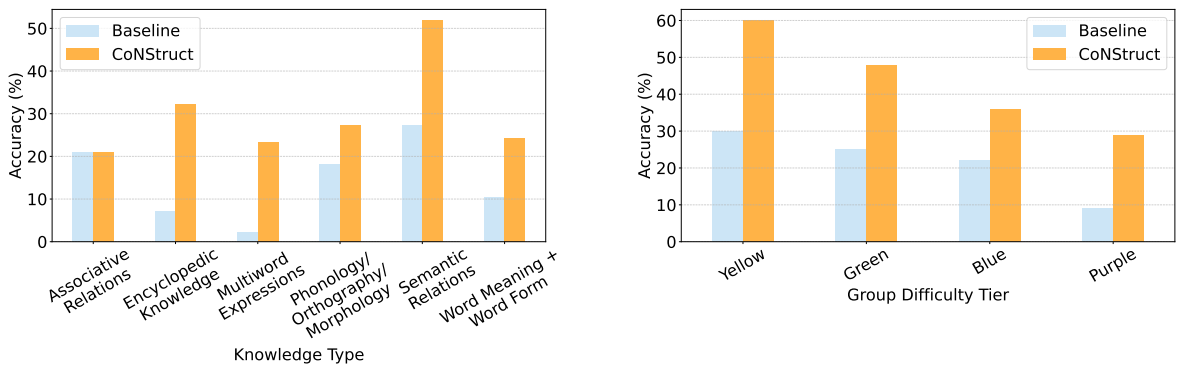

Figure 13: Comparing the baseline (blue) with CoNStruct (orange), using the Gemma-3 27B model. Left: Accuracy across different knowledge types. y-axis: percentage of correctly identified groups within a given knowledge type. Results show consistent improvements across all knowledge types. Right: Accuracy across different difficulty tiers. y-axis: percentage of correctly identified groups within each tier: Yellow (easiest) through Purple (hardest). Results show consistent improvements across all difficulty levels.

F Ablations Results for each Model

F.1 Source Overlap Matrix for LLaMA-3.1 70B

To better understand the relationship between modules, we compute a source overlap matrix (Table 4). For each concept generation module, we again look at the groups initially suggested that have 3-5 words. We extract only the groups that match the ground truth (i.e., have $\geq$3-word overlap with a ground truth group). Then, for each such group from source A, we measure the proportion that also matches a group from source B (again using $\geq$3-word overlap). Diagonal entries represent self-overlap and are therefore 1.0 by definition.

The results show redundancy between the Semantic Relations, Encyclopedic Knowledge and the General Connections modules, which recover many of the same correct groups. In contrast, the Multiword Expression (MWE) module shows minimal overlap with the others, suggesting it contributes distinct correct groups that are missed by other sources.

Src A \ B	Sem	Enc	Gen	MWE
Sem	1.000	0.573	0.772	0.028
Enc	0.602	1.000	0.659	0.028
Gen	0.701	0.569	1.000	0.011
MWE	0.273	0.318	0.136	1.000

Table 4: Proportion of correct group from each module (row, A) that have a $\geq$3-word overlap with correct groups from another module (column, B). For this analysis, we define a group as correct if it has a $\geq$3-word overlap with a ground-truth group. Results for the LLaMA-3.1 70B model show overlap among the Semantic, Encyclopedic, and General modules, while the MWE module shows minimal overlap.

F.2 LLaMA-3.1 8B Model

The results for the ablations for the LLaMA-3.1 8B Model can be seen in Table 5, Table 6 and in Table 7.

Metric	Partial Credit	Unique Credit
MWE	6.67	4
Encyclopedic	16.5	2
Semantic	37.17	12
General	32.67	8

Table 6: Attribution of correct final groups to each concept generation module when using the LLaMA-3.1 8B model. The Semantic and General modules had the broadest overall impact, while the MWE module, despite contributing to fewer groups overall, was uniquely responsible for 4 groups.

Metric	Full Pipeline	No Leftover Words	No Constraint Sat.†	No Refinement†
Unweighted Score	1.15	0.93	0.80	0.51
Weighted Score	2.46	1.82	1.58	1.04
completely solved Puzzles	13	0 (+10)*	0 (+6)*	4 (+0)*

Table 5: Ablation results on LLaMA-3.1 8B showing the impact of removing Leftover Words handling, the Constraint Satisfaction Algorithm, or Refinement. Removing any component reduces performance across all metrics. *Numbers in parentheses indicate additional puzzles effectively solved with 3 out of 4 correct groups, see Note. †The Leftover Words stage is also omitted when Constraint Satisfaction or Refinement is ablated, to better isolate the effect of those components without the effect of adding new candidates.

Src A \ B	Sem	Enc	Gen	MWE
Sem	1.000	0.276	0.572	0.038
Enc	0.471	1.000	0.507	0.007
Gen	0.568	0.265	1.000	0.032
MWE	0.182	0.045	0.455	1.000

Table 7: Proportion of correct group from each module (row, A) that have a $\geq$3-word overlap with correct groups from another module (column, B). For this analysis, we define a group as correct if it has a $\geq$3-word overlap with a ground-truth group. Results for the LLaMA-3.1 8B model show overlap among the Semantic, Encyclopedic, and General modules, while the MWE module shows minimal overlap.

F.3 Phi-4 14B Model

The results for the ablations for the Phi-4 14B Model can be seen in Table 8, Table 9 and in Table 10.

Metric	Partial Credit	Unique Credit
MWE	7.92	5
Encyclopedic	28.58	3
Semantic	44.25	7
General	44.25	9

Table 9: Attribution of correct final groups to each concept generation module when using the Phi-4 14B model. The Semantic and General modules had the broadest overall impact, while the MWE module, despite contributing to fewer groups overall, was uniquely responsible for 5 groups.

Src A \ B	Sem	Enc	Gen	MWE
Sem	1.000	0.523	0.732	0.017
Enc	0.617	1.000	0.728	0.028
Gen	0.654	0.555	1.000	0.018
MWE	0.182	0.227	0.227	1.000

Table 10: Proportion of correct group from each module (row, A) that have a $\geq$3-word overlap with correct groups from another module (column, B). For this analysis, we define a group as correct if it has a $\geq$3-word overlap with a ground-truth group. Results for the Phi-4 14B model show overlap among the Semantic, Encyclopedic, and General modules, while the MWE module shows minimal overlap.

Metric	Full System	No Leftover Words	No Constraint Sat.†	No Refinement†
Unweighted Score	1.37	1.28	1.12	0.81
Weighted Score	2.95	2.60	2.25	1.54
completely solved Puzzles	16	8 (+8)*	2 (+11)*	3 (+1)*

Table 8: Ablation results on Phi-4 14B showing the impact of removing Leftover Words handling, the Constraint Satisfaction Algorithm, or Refinement. Removing any component reduces performance across all metrics. *Numbers in parentheses indicate additional puzzles effectively solved with 3 out of 4 correct groups, see Note. †The Leftover Words stage is also omitted when Constraint Satisfaction or Refinement is ablated, to better isolate the effect of those components without the effect of adding new candidates.

F.4 Mistral-3.1-Small 24B Model

The results for the ablations for the Mistral-3.1-Small 24B Model can be seen in Table 11, Table 12 and in Table 13.

Metric	Partial Credit	Unique Credit
MWE	6.92	5
Encyclopedic	38.92	2
Semantic	44.92	5
General	51.25	7

Table 12: Attribution of correct final groups to each concept generation module when using the Mistral-3.1-Small 24B. The Semantic and General modules had the broadest overall impact, while the MWE module, despite contributing to fewer groups overall, was uniquely responsible for 5 groups.

Src A \ B	Sem	Enc	Gen	MWE
Sem	1.000	0.544	0.721	0.013
Enc	0.526	1.000	0.703	0.016
Gen	0.561	0.581	1.000	0.032
MWE	0.091	0.136	0.318	1.000

Table 13: Proportion of correct group from each module (row, A) that have a $\geq$3-word overlap with correct groups from another module (column, B). For this analysis, we define a group as correct if it has a $\geq$3-word overlap with a ground-truth group. Results for the Mistral-3.1-small 24B model show overlap among the Semantic, Encyclopedic, and General modules, while the MWE module shows minimal overlap.

F.5 Gemma-3 27B Model

The results for the ablations for the Gemma-3 27B Model can be seen in Table 14, Table 15 and in Table 16.

Metric	Partial Credit	Unique Credit
MWE	7.58	5
Encyclopedic	38.42	4
Semantic	42.58	5
General	47.42	6

Table 15: Attribution of correct final groups to each concept generation module when using the Gemma-3 27B model. The Semantic and General modules had the broadest overall impact, while the MWE module, despite contributing to fewer groups overall, was uniquely responsible for 5 groups.

Src A \ B	Sem	Enc	Gen	MWE
Sem	1.000	0.633	0.724	0.010
Enc	0.518	1.000	0.618	0.035
Gen	0.598	0.630	1.000	0.037
MWE	0.091	0.364	0.455	1.000

Table 16: Proportion of correct group from each module (row, A) that have a $\geq$3-word overlap with correct groups from another module (column, B). For this analysis, we define a group as correct if it has a $\geq$3-word overlap with a ground-truth group. Results for the Gemma-3 27B model show overlap among the Semantic, Encyclopedic, and General modules, while the MWE module shows minimal overlap.

Metric	Full System	No Leftover Words	No Constraint Sat.†	No Refinement†
Unweighted Score	1.62	1.44	1.47	0.96
Weighted Score	3.39	2.81	2.91	1.84
completely solved Puzzles	18	4 (+13)*	2 (+14)*	6 (+2)*

Table 11: Ablation results on Mistral-3.1-Small 24B showing the impact of removing Leftover Words handling, the Constraint Satisfaction Algorithm, or Refinement. Removing any component, except the constraint satisfaction algorithm, reduces performance across all metrics. This is the only model tested in which the constraint satisfaction stage did not improve results. *Numbers in parentheses indicate additional puzzles effectively solved with 3 out of 4 correct groups, see Note. †The Leftover Words stage is also omitted when Constraint Satisfaction or Refinement is ablated, to better isolate the effect of those components without the effect of adding new candidates.

Metric	Full System	No Leftover Words	No Constraint Sat.†	No Refinement†
Unweighted Score	1.73	1.38	1.26	0.97
Weighted Score	3.80	2.74	2.43	1.98
completely solved Puzzles	22	2 (+13)*	0 (+10)*	5 (+3)*

Table 14: Ablation results on Gemma-3 27B showing the impact of removing Leftover Words handling, the Constraint Satisfaction Algorithm, or Refinement. Removing any component reduces performance across all metrics. *Numbers in parentheses indicate additional puzzles effectively solved with 3 out of 4 correct groups, see Note. †The Leftover Words stage is also omitted when Constraint Satisfaction or Refinement is ablated, to better isolate the effect of those components without the effect of adding new candidates.

G Human-Likeness of Mistakes

We analyze whether `CoNStruct` mistakes resemble common human errors. We construct a dataset of all the incorrect groups suggested by `CoNStruct` in the solution after the Constraint Satisfaction stage.

For each wrong grouping, we use the reported percentage of people making the same mistake. Because some groups are chosen by less than 1% of people, we convert those into a numeric estimate (e.g., treating $< 1\%$ as 0.5%) for summary statistics.

As shown in Table 17, 73.68% of incorrect groupings in our dataset are made by at least 1% of human players, but only 28.95% reach $\geq$10% and only 7.89% reach $\geq$50%. This indicates a substantial overlap between `CoNStruct`'s errors and human confusions, while suggesting that most errors are not among the most frequent "human-like" traps. One possible explanation that these human statistics reflect iterative gameplay with intermediate guesses, whereas `CoNStruct` produces a complete solution in a single pass, which may reduce exposure to the most salient early distractors that humans tend to try before refining their hypotheses.

Min. freq. (%)	% $\geq$ min	# groups
0.5	100.00	38
1.0	73.68	28
2.0	47.37	18
5.0	36.84	14
10.0	28.95	11
20.0	23.68	9
30.0	18.42	7
40.0	13.16	5
50.0	7.89	3

Table 17: Human prevalence of incorrect groupings. **Min. freq.** is the minimum percentage of human players reported to make a specific incorrect grouping. **% $\geq$ min** is the fraction of incorrect groupings in our dataset whose human frequency meets or exceeds that threshold, and **# groups** is the corresponding count.

H Label Semantic Similarity Analysis

As an additional analysis, we examine the semantic alignment between `CoNStruct`'s group labels and the official NYT labels for correctly predicted groups (using our results with the LLaMA-3.1 70B model). We note that rationale alignment is not required for success — just as human solvers are evaluated solely on correct groupings, `CoNStruct`'s labels need not match the NYT's, as multiple valid rationales can exist for the same group. Take for example the words RUBY, SAPPHIRE, JASPER, LAPIS LAZULI could be *Gems* but also *Characters From Steven Universe*. Nevertheless, as a supplementary observation, we compute cosine simi-

larity between label embeddings using all-mpnet-base-v2, excluding fallback "Remaining Group" labels. As a lower bound, we compute similarity between randomly paired labels from different puzzles. Since NYT labels are often terse and cryptic while CoNStruct's are more descriptive, we also prompt the LLM used by CoNStruct to rephrase each NYT label into a descriptive form while preserving the original meaning and staying as close to the original wording as possible (see Appendix J.7 for prompt), using the similarity between rephrased and original NYT labels as an upper bound. As shown in Table 18, CoNStruct's labels achieve a mean similarity of 0.48 against raw NYT labels and 0.50 against rephrased ones, well above the random baseline of 0.10. The upper bound of 0.75 reflects the stylistic gap between terse NYT labels and descriptive ones, and CoNStruct reaches approximately 64% of this ceiling. This suggests that CoNStruct's rationales, while not required to match the NYT's, are generally not arbitrary — they tend to reflect semantically related concepts to the official ones.

	Mean	**Median**	**Std**
Random baseline	0.1	0.09	0.09
vs. NYT labels	0.48	0.47	0.21
vs. rephrased labels	0.5	0.51	0.19
Upper bound	0.75	0.81	0.18

Table 18: Cosine similarity between CoNStruct's labels and NYT labels for correctly predicted groups. "vs. NYT labels" compares CoNStruct directly to the original NYT labels; "vs. Rephrased labels" compares CoNStruct to descriptive rephrasings of the NYT labels; "Upper bound" measures similarity between rephrased and original NYT labels.

I Taxonomy and Distribution of Knowledge Types

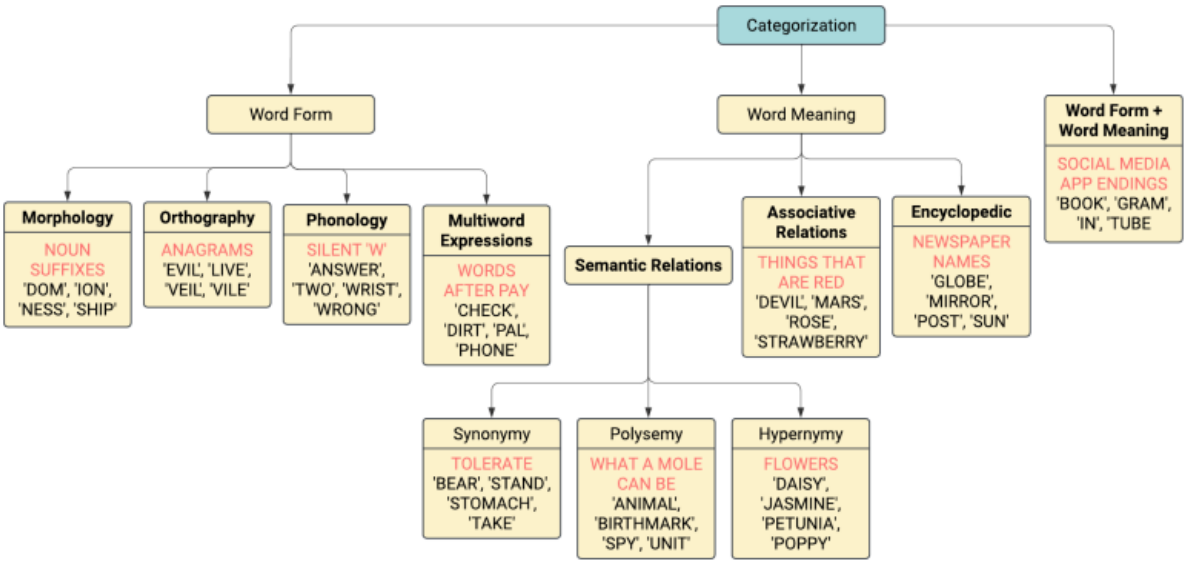

Figure 14: Taxonomy of knowledge types required to solve the Connection games. Reproduced from Samdarshi et al. (2024)

Word Form		Word Meaning			Word Meaning + Word Form
Phonology/Orthography/ Morphology	Multiword Expressions	Semantic Relations	Associative Relations	Encyclopedic	92
44	168	1045	137	266	

Figure 15: Distribution of different knowledge types required to categorize words across 438 games. From the paper of Samdarshi et al. (2024)

J Prompts

J.1 Baseline

J.1.1 Reasoning Prompt:

User Message:

```
Solve today's NYT Connections game. Here are the
     instructions for how to play this game:
Find groups of four items that share something
    in common.

Category Examples:
FISH: Bass, Flounder, Salmon, Trout
FIRE ___: Ant, Drill, Island, Opal
Categories will always be more specific than '5-
    LETTER-WORDS', 'NAMES', or 'VERBS.'

Example 1:
Words: ['DART', 'HEM', 'PLEAT', 'SEAM', 'CAN', '
    CURE', 'DRY', 'FREEZE', 'BITE', 'EDGE', '
    PUNCH', 'SPICE', 'CONDO', 'HAW', 'HERO', '
    LOO']

Groupings:
1. Things to sew: ['DART', 'HEM', 'PLEAT', 'SEAM
    ']
2. Ways to preserve food: ['CAN', 'CURE', 'DRY',
     'FREEZE']
3. Sharp quality: ['BITE', 'EDGE', 'PUNCH', '
    SPICE']
4. Birds minus last letter: ['CONDO', 'HAW', '
    HERO', 'LOO']

Example 2:
Words: ['COLLECTIVE', 'COMMON', 'JOINT', 'MUTUAL
    ', 'CLEAR', 'DRAIN', 'EMPTY', 'FLUSH', '
    CIGARETTE', 'PENCIL', 'TICKET', 'TOE', '
    AMERICAN', 'FEVER', 'LUCID', 'PIPE']

Groupings:
1. Shared: ['COLLECTIVE', 'COMMON', 'JOINT', '
    MUTUAL']
2. Rid of contents: ['CLEAR', 'DRAIN', 'EMPTY',
    'FLUSH']
3. Associated with "stub": ['CIGARETTE', 'PENCIL
    ', 'TICKET', 'TOE']
4. ___ Dream: ['AMERICAN', 'FEVER', 'LUCID', '
    PIPE']

Example 3:
Words: ['HANGAR', 'RUNWAY', 'TARMAC', 'TERMINAL',
     'ACTION', 'CLAIM', 'COMPLAINT', 'LAWSUIT',
    'BEANBAG', 'CLUB', 'RING', 'TORCH', '
    FOXGLOVE', 'GUMSHOE', 'TURNCOAT', 'WINDSOCK
    ']

Groupings:
1. Parts of an airport: ['HANGAR', 'RUNWAY', '
    TARMAC', 'TERMINAL']
2. Legal terms: ['ACTION', 'CLAIM', 'COMPLAINT',
     'LAWSUIT']
3. Things a juggler juggles: ['BEANBAG', 'CLUB',
     'RING', 'TORCH']
4. Words ending in clothing: ['FOXGLOVE', '
    GUMSHOE', 'TURNCOAT', 'WINDSOCK']

Categories share commonalities:
- There are 4 categories of 4 words each
- Every word will be in only 1 category
- One word will never be in two categories
- As the category number increases, the
    connections between the words and their
    category become more obscure. Category 1 is
    the most easy and intuitive and Category 4
    is the hardest
- There may be red herrings (words that seem to
    belong together but actually are in separate
     categories)
- Category 4 often contains compound words with
    a common prefix or suffix word
- A few other common categories include word and
     letter patterns, pop culture clues (such as
     music and movie titles) and fill-in-the-
    blank phrases

You will be given a new example (Example 4) with
     today's list of words.
First explain your reason for each category and
    then give your final answer following the
    structure below (Replace Category 1, 2, 3, 4
     with their names instead)

Groupings:
Category1: [word1, word2, word3, word4]
Category2: [word5, word6, word7, word8]
Category3: [word9, word10, word11, word12]
Category4: [word13, word14, word15, word16]

Remember that the same word cannot be repeated
    across multiple categories, and you need to
    output 4 categories with 4 distinct words
    each. Also do not make up words not in the
    list. This is the most important rule.
    Please obey.

Example 4:

Words: <puzzle words>

Groupings:
```

J.1.2 JSON Extraction Prompt:

System Message:

```
You extract JSON lists from language reasoning.
    Only output valid JSON.
```

User Message:

```
Based on the reasoning and the groupings below,
    extract the groups and present them in **
    valid JSON** format.

Each group must have a **name** describing the
    category, and a list of words that belong in
     the group.

---

Reasoning:
<reasoning output from previous prompt>

---

Format the output exactly like this:

[
```

```
    {{ "group_name": "Group Name", "words": ["
        word1", "word2", "word3, word4"] }},
    {{ "group_name": "Another Group", "words": ["
        word5", "word6", "word7", "word8"] }},
    {{ "group_name": "Another Group", "words": ["
        word9", "word10", "word11", "word12"] }},

    {{ "group_name": "Another Group", "words": ["
        word13", "word14", "word15", "word16"] }}

]

Rules:
- Only include the JSON array.
- Do NOT use triple backticks or Markdown
    formatting.
- No extra explanations or commentary.
- Group names should reflect the shared meaning
    or category.
- Use the exact original words.
- Include all suggested groups from the
    reasoning
```

J.2 Semantic Relations Module

J.2.1 Reasoning Prompt:

System Message:

```
You are a language expert solving a word
    association puzzle. Think out loud to find
    semantic connections like synonyms or
    categories.
```

User Message:

```
You are solving a word puzzle. There are 16
    words. Your goal is to find as many possible
     groups of 3-5 words that share a **clear
    semantic connection**. At this stage, focus
    only on groups where the words are **
    synonyms** or **"type of"** relationships (
    hypernyms).

Each group should be based on:
- Synonyms (words with very similar meanings)
- Hypernyms (words that are all types of a
    shared category)
- Simple connections - avoid abstract or
    metaphorical themes

Each group must have **3 to 5 words**. Some
    words may appear in more than one group. It'
    s okay to suggest **more than 4 groups**.

---

### Example Puzzle:

Words:
EGG, STORY, SUN, SCREEN, MOON, REEL, STREAK,
    POST, GLOBE, DECK, SPEAKER, FLOOR, TOILET
    PAPER, MIRROR, LEVEL, PROJECTOR

**Reasoning:**
- PROJECTOR, REEL, SCREEN, SPEAKER are all
    related to equipment found in a classic
    movie theater.
- DECK, FLOOR, LEVEL, STORY are all words that
    describe levels or tiers of a structure.
- GLOBE, MIRROR, POST, SUN are names of popular
    newspapers.
- EGG, MOON, STREAK, TOILET PAPER are all verbs
    that describe common pranks.

---

### Example Puzzle:

Words:
DART, HEM, PLEAT, SEAM, CAN, CURE, DRY, FREEZE,
    BITE, EDGE, PUNCH, SPICE, CONDO, HAW, HERO,
    LOO

**Reasoning:**
- DART, HEM, PLEAT, SEAM are all things you can
    sew.
- CAN, CURE, DRY, FREEZE are all ways to
    preserve food.
- BITE, EDGE, PUNCH, SPICE are all ways to say
    something has a "sharp quality".
- CONDO, HAW, HERO, LOO are all bird names with
    the last letter removed.

---

### Example Puzzle:

Words:
COLLECTIVE, COMMON, JOINT, MUTUAL, CLEAR, DRAIN,
     EMPTY, FLUSH, CIGARETTE, PENCIL, TICKET,
    TOE, AMERICAN, FEVER, LUCID, PIPE

**Reasoning:**
- COLLECTIVE, COMMON, JOINT, MUTUAL all mean
    shared.
- CLEAR, DRAIN, EMPTY, FLUSH relate to removing
    contents.
- CIGARETTE, PENCIL, TICKET, TOE are all
    associated with the word "stub".
- AMERICAN, FEVER, LUCID, PIPE can complete the
    phrase "__ dream".

---

### Example Puzzle:

Words:
HANGAR, RUNWAY, TARMAC, TERMINAL, ACTION, CLAIM,
     COMPLAINT, LAWSUIT, BEANBAG, CLUB, RING,
    TORCH, FOXGLOVE, GUMSHOE, TURNCOAT, WINDSOCK

**Reasoning:**
- HANGAR, RUNWAY, TARMAC, TERMINAL are all parts
     of an airport.
- ACTION, CLAIM, COMPLAINT, LAWSUIT are legal
    terms.
- BEANBAG, CLUB, RING, TORCH are things a
    juggler might juggle.
- FOXGLOVE, GUMSHOE, TURNCOAT, WINDSOCK are all
    words that end with a type of clothing.

---

Now, analyze this new set of 16 words and
    suggest all the synonym/hypernym-based
    groups you can find.

Words:
<puzzle words>
```

```
Give your reasoning and group suggestions.
```

J.2.2 JSON Extraction Prompt:

System Message:

```
You are a precise JSON generator. Given
    reasoning, extract structured word groups in
     correct JSON format without any extra text.
```

User Message:

```
Based on the reasoning below, extract the groups
     and present them in **valid JSON** format.

Each group must have a **name** describing the
    category, and a list of words that belong in
     the group.

---

Reasoning:
<reasoning output from previous prompt>

---

Format the output exactly like this:

[
  {{ "group_name": "Group Name", "words": ["
       word1", "word2", "word3"] }},
  {{ "group_name": "Another Group", "words": ["
       word1", "word2", "word3", "word4"] }}
]

Rules:
- Only include the JSON array.
- Do NOT use triple backticks or Markdown
    formatting.
- No extra explanations or commentary.
- Group names should reflect the shared meaning
    or category.
- Use the exact original words.
- Include all suggested groups from the
    reasoning
```

J.2.3 Group Expansion Prompt:

System Message:

```
You are an expert language model. Only reply
    with YES or NO.
```

User Message:

```
The current group is:
"<current group name>": <current group words>

Does the word "<current word>" belong in this
    group based on meaning or semantic
    similarity?

Respond with exactly one word: YES or NO.
```

J.3 General Connections Module

J.3.1 Reasoning Prompt:

System Message:

```
You are a puzzle expert thinking out loud about
    connections between words.
```

User Message:

```
You are solving a word puzzle. There are 16
    words. Your goal is to find as many possible
     groups of 3-5 words that share a **clear
    connection**.

Each group must have **3 to 5 words**. Some
    words may appear in more than one group. It'
    s okay to suggest **more than 4 groups**.

---

### Example Puzzle:

Words:
EGG, STORY, SUN, SCREEN, MOON, REEL, STREAK,
    POST, GLOBE, DECK, SPEAKER, FLOOR, TOILET
    PAPER, MIRROR, LEVEL, PROJECTOR

**Reasoning:**
- PROJECTOR, REEL, SCREEN, SPEAKER are all
    related to equipment found in a classic
    movie theater.
- DECK, FLOOR, LEVEL, STORY are all words that
    describe levels or tiers of a structure.
- GLOBE, MIRROR, POST, SUN are names of popular
    newspapers.
- EGG, MOON, STREAK, TOILET PAPER are all verbs
    that describe common pranks.

---

### Example Puzzle:

Words:
DART, HEM, PLEAT, SEAM, CAN, CURE, DRY, FREEZE,
    BITE, EDGE, PUNCH, SPICE, CONDO, HAW, HERO,
    LOO

**Reasoning:**
- DART, HEM, PLEAT, SEAM are all things you can
    sew.
- CAN, CURE, DRY, FREEZE are all ways to
    preserve food.
- BITE, EDGE, PUNCH, SPICE are all ways to say
    something has a "sharp quality".
- CONDO, HAW, HERO, LOO are all bird names with
    the last letter removed.

---

### Example Puzzle:

Words:
COLLECTIVE, COMMON, JOINT, MUTUAL, CLEAR, DRAIN,
     EMPTY, FLUSH, CIGARETTE, PENCIL, TICKET,
    TOE, AMERICAN, FEVER, LUCID, PIPE

**Reasoning:**
- COLLECTIVE, COMMON, JOINT, MUTUAL all mean
    shared.
- CLEAR, DRAIN, EMPTY, FLUSH relate to removing
    contents.
- CIGARETTE, PENCIL, TICKET, TOE are all
    associated with the word "stub".
```

```
- AMERICAN, FEVER, LUCID, PIPE can complete the
    phrase "__ dream".

---

### Example Puzzle:

Words:
HANGAR, RUNWAY, TARMAC, TERMINAL, ACTION, CLAIM,
     COMPLAINT, LAWSUIT, BEANBAG, CLUB, RING,
    TORCH, FOXGLOVE, GUMSHOE, TURNCOAT, WINDSOCK

**Reasoning:**
- HANGAR, RUNWAY, TARMAC, TERMINAL are all parts
     of an airport.
- ACTION, CLAIM, COMPLAINT, LAWSUIT are legal
    terms.
- BEANBAG, CLUB, RING, TORCH are things a
    juggler might juggle.
- FOXGLOVE, GUMSHOE, TURNCOAT, WINDSOCK are all
    words that end with a type of clothing.

---

Now, analyze this new set of 16 words and
    suggest all the groups you can find.

Words:
<puzzle words>

Give your reasoning and group suggestions.
```

J.3.2 JSON Extraction Prompt:

System Message:

```
You extract JSON lists from language reasoning.
    Only output valid JSON.
```

User Message:

```
Based on the reasoning below, extract the groups
     and present them in **valid JSON** format.

Each group must have a **name** describing the
    category, and a list of words that belong in
     the group.

---

Reasoning:
<reasoning output from previous prompt>

---

Format the output exactly like this:

[
  {{ "group_name": "Group Name", "words": ["
      word1", "word2", "word3"] }},
  {{ "group_name": "Another Group", "words": ["
      word1", "word2", "word3", "word4"] }}
]

Rules:
- Only include the JSON array.
- Do NOT use triple backticks or Markdown
    formatting.
- No extra explanations or commentary.
- Group names should reflect the shared meaning
    or category.
- Use the exact original words.
- Include all suggested groups from the
    reasoning
```

J.3.3 Group Expansion Prompt:

System Message:

```
You are an expert language model. Only reply
    with YES or NO.
```

User Message:

```
The current group is:
"<current group name>": <current group words>

Does the word "<current word>" belong in this
    group based on theme or category?

Respond with exactly one word: YES or NO.
```

J.4 Encyclopedic Knowledge Module

J.4.1 Filter Wikipedia Titles Prompt:

System Message:

```
Answer with just 'Yes' or 'No'.
```

User Message:

```
Answer with just 'Yes' or 'No'.

Is '<word>' a type of '<Wikipedia title>'
```

J.4.2 Reasoning Prompt:

System Message:

```
You are an expert linguist analyzing semantic
    relationships.
```

User Message:

```
You are given a list of 16 words and a set of
    Wikipedia pages.
Each page contains 3-word combinations where all
     words are considered types or examples of
    the page topic.

Your task is to group these 16 words into **four
     distinct groups of 4 words**
Each group should share a strong 'type of'
    relationship - a clear, meaningful category
    that connects them all.

For example:
- 'PROJECTOR, REEL, SCREEN, SPEAKER' -> Movie
    Theater Equipment
- 'SUN, POST, GLOBE, MIRROR' -> Newspaper Names

Here is the list of words:

<puzzle words>

Here are the Wikipedia pages where specific 3-
    word combinations were found:
```

```
Words <3-word combination> matched these pages:
    <pages>
Words <3-word combination> matched these pages:
    <pages>
.
.
.

Now, based on this information, divide the list
    into **four groups of 4 words**.
Each group must have a clear and specific
    category label.

Be sure to pick the most meaningful and well-
    supported groupings based on the Wikipedia
    context.
```

J.4.3 JSON Extraction Prompt:

System Message:

```
You are a precise JSON extractor. Follow the
    format exactly.
```

User Message:

```
Based on the reasoning below for how to group 16
     words, extract the groups and present them
    in **valid JSON** format.

Each group must have a **name** describing the
    category, and a list of words that belong in
     the group.

---

Reasoning:
<reasoning output from previous prompt>

---

Format the output exactly like this:

{{
    "group_1": {{
    "label": "Group Label",
    "words": ["word1", "word2", "word3", "word4"]
    }},
    "group_2": {{
    "label": "Group Label",
    "words": ["word5", "word6", "word7", "word8"]
    }},
    "group_3": {{
    "label": "Group Label",
    "words": ["word9", "word10", "word11", "
        word12"]
    }},
    "group_4": {{
    "label": "Group Label",
    "words": ["word13", "word14", "word15", "
        word16"]
    }}
}}

Rules:
- Only include the JSON array.
- Do NOT use triple backticks or Markdown
    formatting.
- No extra explanations or commentary.
- Group names should reflect the shared meaning
    or category.
- Use the exact original words.
- Include all suggested groups from the
    reasoning
```

J.4.4 Group Expansion Prompt:

System Message:

```
You are an expert language model. Only reply
    with YES or NO.
```

User Message:

```
The current group is:
"<current group name>": <current group words>

Does the word "<current word>" belong in this
    group based on theme or category?

Respond with exactly one word: YES or NO.
```

J.5 Refinement

J.5.1 Reasoning Prompt:

System Message:

```
You are a linguist analyzing semantic subgroups.
```

User Message:

```
You are given a group of words called "<
    group_name>":

<words>

Your task is to analyze whether a **subgroup of
    3 or more words** exists that has a **
    stronger and more specific connection** than
     the full group.

If such a subgroup exists, explain why the
    subgroup is stronger, and which words form
    that subgroup.

If not, explain why the full group is cohesive
    enough and does not need refinement.
```

J.5.2 JSON Extraction Prompt:

System Message:

```
You are a precise JSON extractor. Follow the
    format exactly.
```

User Message:

```
Below is a reasoning explanation. Based on that
    explanation, extract a subgroup and give it
    a clear label.

Respond with a JSON object like this:

{{
  "group_name": "Refined Label",
  "words": ["word1", "word2", "word3"]
}}
```

```
If the reasoning clearly says no better subgroup
     exists, respond with just the string:

"No better subgroup"

---

Reasoning:
<reasoning output from previous prompt>
```

J.5.3 Group Expansion Prompt:

System Message:

```
You are an expert language model. Only reply
    with YES or NO.
```

User Message:

```
The current group is:
"<current group name>": <current group words>

Does the word "<word>" belong in this group
    based on theme or category?

Respond with exactly one word: YES or NO.
```

J.6 After Remaining Stage

J.6.1 Reasoning Prompt:

System Message:

```
You are a puzzle expert thinking out loud about
    connections between words."
```

User Message:

```
You are solving a word puzzle. There are <number
     of puzzle words> words. Your goal is to
    find groups of 4 words that share a **clear
    connection**.

Each group must have **4 words**. Make sure each
     word appears in exactly one group. Suggest
    exactly **<number of puzzle words divided by
     4> groups**.



---

### Example Puzzle:

Words:
EGG, STORY, SUN, SCREEN, MOON, REEL, STREAK,
    POST, GLOBE, DECK, SPEAKER, FLOOR, TOILET
    PAPER, MIRROR, LEVEL, PROJECTOR

**Reasoning:**
- PROJECTOR, REEL, SCREEN, SPEAKER are all
    related to equipment found in a classic
    movie theater.
- DECK, FLOOR, LEVEL, STORY are all words that
    describe levels or tiers of a structure.
- GLOBE, MIRROR, POST, SUN are names of popular
    newspapers.
- EGG, MOON, STREAK, TOILET PAPER are all verbs
    that describe common pranks.

---

Now, analyze this new set of <number of puzzle
    words> words and suggest groups.

Words:
<puzzle words>

Give your reasoning and group suggestions.
    Remember that the same word cannot be
    repeated across multiple categories
```

J.6.2 JSON Extraction Prompt:

System Message:

```
You extract JSON lists from language reasoning.
    Only output valid JSON.
```

User Message:

```
Based on the reasoning below, extract the groups
     and present them in **valid JSON** format.

Each group must have a **name** describing the
    category, and a list of words that belong in
     the group.

---

Reasoning:
<reasoning output from previous prompt>

---

Format the output exactly like this:

[
  {{ "group_name": "Group Name", "words": ["
      word1", "word2", "word3", "word4"] }},
  {{ "group_name": "Another Group", "words": ["
      word1", "word2", "word3", "word4"] }}
]

Rules:
- Only include the JSON array.
- Do NOT use triple backticks or Markdown
    formatting.
- No extra explanations or commentary.
- Group names should reflect the shared meaning
    or category.
- Use the exact original words.
- Include all suggested groups from the
    reasoning
```

J.7 Label Semantic Similarity Analysis

J.7.1 Rephrasing NYT Labels Prompt:

System Message:

```
You are a precise language model. Only reply
    with the rephrased label.
```

User Message:

```
Expand the following short puzzle category name
    into a slightly more descriptive label,
    while staying as close to the original
    wording as possible.
```

```
Only add words if necessary to make it clearer
    do not change the meaning or introduce new
    concepts.

The words in the category are: <Category Words>.

Category name: <NYT Label>

Respond with only the expanded label, nothing
    else.
```