# Learning Semantic Complexities of NYT Connections

Stanford CS224N Custom Project

**Emily Zhang**[*]
Department of Computer Science
Stanford University
emily49@stanford.edu

**Yanan Jiang**[*]
Department of Mathematics
Stanford University
yjiang26@stanford.edu

**Peixuan Ye**[*]
Department of Computer Science
Stanford University
pxye@stanford.edu

## Abstract

While modern NLP models are adept at handling syntactically rich prose and sentence structure, they still struggle with the flexible problem solving and general reasoning ability that humans exhibit. In this paper, we apply diverse NLP methods to one example of a semantically complex reasoning task, *The New York Times* Connections game. We explore baseline methods that are specific to the semantic grouping task, such as K-Means clustering on various word embeddings. Then, we benchmark in-context learning on a large LLM as well as finetune a smaller LLM using existing examples of solved puzzles. We also explore unique methods of introducing complex linguistic and metalinguistic reasoning capabilities of a larger LLM into a smaller LLM, through methods like fine-tuning with distilled rationales and Verbal Reinforcement Learning. We found promising results with Verbal RL, as the smaller model was able to absorb critique from a larger language model and improve on previous answers, similar to how human solvers approach language puzzles.

## 1 Key Information to include

- Mentor: Ryan Li

## 2 Introduction

The cognitive ability to draw associations between meaningful entities underlies human capabilities to learn complex ideas and build new knowledge.[1] This ability is well-captured by the NLP task of semantic grouping, which requires a language model to establish and evaluate word-to-word connections. Semantic grouping requires model capacity beyond simple classification and demands a nuanced understanding of different orders of word-to-word relationships, and a subtle learnable strategy of prioritizing different relations in order to yield the most sensible groupings.

Existing NLP approaches to semantic grouping often rely on performing clustering or graph-based algorithms directly on word embeddings, framing the problem in the paradigm of discrete optimization[2][3][4][5]. Though incorporating Deep Learning methods brings better performance[6], these models failed to address many semantic grouping tasks in interactive contexts, since word-embeddings have limited expressivity.

[*]These authors contributed equally to this work.

Building upon existing methods, our project aims to take a novel approach by training a language model to acquire semantic grouping capabilities through learning the New York Times game "Connections". Inspired by NLP research that utilizes game settings and train models to solve word puzzles to acquire generalizable abilities, we encapsulated the semantic grouping objective in a word-puzzle task and train. This game-centric method diverges from the conventional reliance on ML-based clustering techniques, seeking to exploit the highly-adaptive learning capabilities of language models.

To elicit better reasoning capabilities for the model, we fine-tuned our model using a combination of Distillation and Verbal Reinforcement Learning methods, integrating the complex reasoning abilities of LLM models into the fine-tuning process of a smaller Mistral Instruct Model. We've found that the model has significantly improved semantic grouping ability after our training pipeline.

# 3 Related Work

## 3.1 Semantic Grouping

Embedding-based approaches to semantic grouping often leverage techniques like clustering [2][3][4], graph-based methods [5], and dimensionality reduction[7][8]. General embedding-based clustering methods group words based on semantic similarity extracted from large text corpora. Transformer models have improved this process by dynamically understanding context within sentences, thereby enhancing the quality and relevance of the groupings.[6] Despite the advancements, these models often struggle with capturing the more interactive and playful aspects of human cognitive associations, such as those needed in word games or creative contexts, where flexibility and adaptability are crucial.

## 3.2 Solving Word Puzzles

There is extensive literature on NLP methods for solving word puzzles such as puns, anagrams, and various types of crosswords. Rozner, Potts, and Mahowald [9] investigated the T5 language model's ability to solve cryptic crosswords—a two-part crossword consisting of a definition and a wordplay cipher which require more linguistic flexibility and better understanding of higher semantic complexity. They synthesized the training crossword data and proposed a novel curriculum learning approach which involves pre-fine-tuning the model on related but simpler tasks such as word unscrambling and American crosswords, which appeared effective in improving the op-1 accuracy score on various types of data splits.

Drawing inspiration from Rozner, Potts, and Mahowald, we decided to focus on NYT Connections puzzles, as identifying the correct grouping among the 16 words in each puzzles also requires advanced semantic familiarity and significant creativity and linguistic flexibility. We adopted a similar K-Means clustering model for baseline evaluation with different word-vector embeddings. On the other hand, our approach differs from theirs primarily in our much smaller dataset consisting of authentic Connections puzzles and the incorporation of distillation in fine-tuning our language model instead of pre-fine-tuning on simpler tasks.

## 3.3 Distillation and Verbal RL

Hsieh et al. [10] introduced step-by-step distillation to effectively utilize less training data for fine-tuning small language models, while allowing them to outperform large language models (LLMs). Given the success of this distillation mechanism in improving the T5 model and the limited size of our training dataset, it is a natural choice to consider distillation for fine-tuning Mistral.

Verbal RL[11] is a form of reinforcement learning that incorporates natural language feedback to guide a model's learning process, allowing for more complex and nuanced information to be conveyed beyond traditional numeric rewards. By integrating verbal cues and instructions in Verbal RL Pipeline, our model could better understand and refine the reasoning behind its semantic grouping choices, thereby making more-informed decisions in future iterations.

# 4 Approach

Our approach involves two baseline approaches and three main approaches.

### 4.1 Baseline

The baseline involves using word-embedding methods, namely GloVe [12] and word2Vec [13], to obtain word embedding for each of the 16 input words in the game, and directly performs high-dimensional clustering on these word vectors. Since every instance of the Connections game results in 4 groups of 4 words each, we chose "Balanced K-Means" as our clustering method. (We wrote our own code for Balanced K-Means Implementation).[14]

### 4.2 In-Context Learning with DBRX

The first main approach uses an in-context learning method.[15] We prompted the DBRX model to learn from a few example puzzles and solutions (3 examples per prompt) with minimal descriptions for the groups. We also included necessary instructions on how to play the Connections Game from *New York Times*'s official guide along with hints to avoid common mistakes.

### 4.3 Finetuning Mistral-Instruct

We also finetuned a Mistral Instruct model (Mistral-7B-Instruct-v0.2 [16]) on the training split of our task-specific Connections dataset. We included the *NYT* instructions in our prompt (see Appendix B) without direct in-context examples. For fine-tuning, we use Quantized Low Rank Adaptation (QLoRA)[17], an improvement on LoRA that further increases parameter efficiency and speeds up training.

After vanilla finetuning yielded poor results, we drew on the ideas of Hsieh et al. and added a distillation component to our finetuning process, which we hypothesized would improve the model's capability to perform the chain-of-thought reasoning that humans employ to play the *Connections* game.[18] We extracted rationales from the larger DBRX-Instruct model and used them as additional supervision while finetuning Mistral-Instruct.

In particular, for all training and validation examples, we inputted into the model:

1. The same instructions and puzzle of 16 words that the Mistral model received.
2. The correct answer and brief, few-word descriptions of the provided given by *NYT* that the Mistral model *did not* previously receive.
3. Context that it was teaching a student to play the game.

We prompted it to generate chain-of-thought, pedagogical rationales behind the groupings. The full prompt can be found in Appendix B. After generating DBRX's rationales, we fine-tuned the smaller Mistral model to refer to them as ground truths and generate similar-capability rationales followed by the groupings. The prompt we used for this step can be found in Appendix B. Distillation allowed us to transfer some of the reasoning ability of the large model to our small Mistral model that we can reasonably fine-tune.

### 4.4 Verbal Reinforcement Learning

To enabled our fine-tuned model to verbally reflect on its misgrouping instances and thereby arrive at a better solution, we tried applying various versions of Verbal RL [11] at the end of our fine-tuning pipeline. In every iteration where the model generates a version of answer, we feed that that answer to a DBRX meta-model so that it could provide critiques, comments, or feedbacks. We then reprompt our fine-tuned model with these relflective rationales, in hopes to induce better decision-making performances for the subsequent trials.

### 4.5 Prompt Engineering (see Appendix B)

We followed the 26 ordered prompt engineering principles provided by Bsharat et al. [19] that have proven effective in questioning LLM models such as LLaMA-2 and GPT-4. For both Mistral fine-tuning and the distillation step with DRBX, we adopted a few-shot prompting technique by providing a few concrete examples in our prompts. Specifically, we demonstrated to the model three possible reasons why four words could be categorized into the same group and presented two example groupings for each reason. Additionally, we broke down the task of identifying four groups at once

by advising the model to identify the groups in increasing order of difficulty, providing hints on the possible form the easiest category takes, which is usually objects that belong in the same category or synonyms. To obtain the desired output format and content consistency, we instructed DBRX to assume a teacher role, guiding the model to produce reasoning that a "student"—Mistral—could follow. Mistral was instructed to output its responses in a JSON list, which proved much more effective than our initial attempt at enforcing a line-by-line format that resulted in the model outputting irrelevant words. Lastly, we assumed an instructive and objective tone and emphasized penalties to ensure the model followed our instructions.

# 5 Experiments

## 5.1 Data

We used a total of 342 *New York Times* Connections puzzles that we scraped from the unofficial Connections archive website (`https://connections.swellgarfo.com/archive`). The archive ranks the four groups of words in each puzzle by their difficulty, which allowed us to create a difficulty-based metric that evaluates our models by the number of groups it correctly clusters in each difficulty category.

We divided the 342 puzzles into training, evaluation, and test datasets with an 80/10/10 split. Therefore, we had a total of 274 puzzles for training and 34 each for testing and evaluation respectively. For all LLM training and testing, we randomly shuffled the order of the 16 words in the prompt.

## 5.2 Evaluation method

We used the following metrics to evaluate our models.

- **Number of Perfect Matches.** The number of puzzles that the model answers completely correctly, i.e. all 4 groups are right.
- **Average Number of Matches per Difficulty.** For each difficulty group, the total number of correct groups of that difficulty divided by the total number of puzzles.
- **Average Number of Matches.** The total number of correct clusters divided by the total number of puzzles. This number is equal to the sum of the four values in **Average Number of Matches per Difficulty**.
- **Average Entropy.** We defined the classification probability as the ratio of the intersection size between each predicted and true clusters divided by the cluster size (4). We took the average of all predicted cluster entropies. Formally, the entropy of each cluster $w$ is defined by: $H(w) = -\sum_{c \in C} P(w_c) \log_2 P(w_c)$, and the entropy per puzzle is given by $H(\Omega) = \sum_{w \in \Omega} \frac{N_w}{N} H(w)$
- **Average Intersection Over Union (IOU).** For each predicted group, its IOU ratio is maximal value among the four IOUs when compared with the four true clusters. This ratio is a fraction between $1/7$ and 1 (perfect match). We took the average of all predicted cluster IOUs.

## 5.3 K-Means Baseline

We used a standard sk-learn KMeans model and set the maximum iteration to be 100, with 0.0001 tolerance, then we applied a rebalancing function to ensure that all clusters represent 4-word groups. We tried two different word embedding methods, namely GloVe and word2Vec.

## 5.4 In-Context Learning

We ran inference on Databrick's DBRX model (132B parameters) using Together AI's API. In the prompt (see Appendix A), we included detailed instructions and 3 examples of correct puzzle solutions along with short descriptions of the groups.

### 5.5 Mistral-Instruct Finetuning and Distillation

For finetuning Mistral-Instruct (8B) with QLoRA, we trained for 3 epochs (after which validation loss started to increase) using AdamW-32bit optimization on an NVIDIA A100 GPU. The other hyperparameters we used are listed in Table 1.

| Parameter Name | Value |
|---|---|
| Batch size | 4 |
| Gradient accumulation steps | 4 |
| Learning rate (initial) | $2e^{-4}$ |
| Weight decay | 0.001 |
| Ratio of steps for linear learning rate warmup | 0.03 |
| Maximum gradient for clipping | 0.3 |
| LoRA update matrix rank | 64 |
| LoRA $\alpha$ scaling factor | 16 |
| LoRA dropout probability | 0.1 |

Table 1: Parameters for Finetuning Mistral-Instruct

Note that even after iterations of careful prompting, the small Mistral model failed at following the instructions and output format, frequently hallucinating words, answering with more or less than four groups, and putting more or less than four words in each group. Our evaluations in Table 2 were run after truncating the groups that were larger than four words and padding the groups with fewer with empty strings. This significantly impacted performance based on on these metrics, as we'll discuss in Analysis.

To maintain comparability, our Mistral-Instruct model with a distillation component was fine-tuned with the same parameters in Table 1. We saw much better task understanding with fewer format errors, but only a small improvement in accuracy, as shown in Table 2.

### 5.6 Verbal Reinforcement Learning

In the Verbal RL Pipeline, we first collected the un-criticized 1st-iteration answer from the model being trained, and feed that answer to the DBRX supervisor model responsible for generating feedbacks. We tested 2 different configurations, one where the DBRX supervisor model is presented with the ground truth of the correct answer, and another where the supervisor only relies on the model-provided answer itself to offer critiques. Then the DBRX-provided feedback is fed to the fine-tuned model as its new prompt, and the model will generate a new iteration of Connections answers. We ran iterative evaluations for the duration of 3 iteration, essentially giving the model 3 chances of trials and error.

| | Baseline | | Main | | | |
|---|---|---|---|---|---|---|
| | GloVe | Word2Vec | In-context DBRX | Finetuned Mistral | With distillation | Verbal RL |
| **Perfect Matches** | 0 | 0 | 0 | 0 | 0 | 1 |
| Avg. Matches (0) | 0.0588 | 0 | 0.1176 | 0.0294 | 0.0588 | 0.222 |
| Avg. Matches (1) | 0.0588 | 0.0294 | 0.1765 | 0.0294 | 0.0588 | 0.222 |
| Avg. Matches (2) | 0.0294 | 0.0294 | 0.0882 | 0.0588 | 0 | 0.111 |
| Avg. Matches (3) | 0.0882 | 0.2058 | 0 | 0 | 0.0588 | 0.111 |
| **Avg. Matches (All)** | 0.2353 | 0.2647 | 0.3823 | 0.1176 | 0.1765 | 0.666 |
| **Avg. Entropy** | 0.7581 | 0.7565 | 0.7338 | 0.8312 | 0.7710 | 0.357 |
| **Avg. IOU** | 0.4513 | 0.4454 | 0.4867 | 0.3491 | 0.3359 | 0.357 |

Table 2: Evaluation of baseline and main methods. We use $0, 1, 2, 3$ to denote the difficulty levels with $0$ being the easiest and $3$ the hardest.

# 6 Analysis

## 6.1 Baseline Clustering

As our baseline, clustering methods based on word-embedding yielded considerably stable performance, but fail consistently on some word-grouping tasks. Notably, word-embeddings have limited expressivity.

This means that a polysemous could suffer from the constraints of a single word-embedding, therefore not able to fall into the correct grouping since the embedding is an averge resting across the border of 2 groups. For instance in one game, the word "look" should be associated to the group of "style", "manner", "dress" due to one of its definition as "appearance", but instead "look" is being clustered together with "sight" due to its other definition as "see/view".

What's more, semantic proximity shouldn't be the only indicator of semantic associativity. Just because 2 words are semantically close to each other doesn't mean that they will fall into the same group in the Connections game, because possible grouping criteria extend beyond "words found in the similar context (semantic proximity)" and could include "words in the same utility category", which is not well reflected by semantic proximity in word embeddings. As from the ealier example, "look" and "sight" are semantically close, but it makes more sense for "look" to associate with "style", "manner", "dress" since they belong to the same category.

## 6.2 In-Context Learning

We found that in-context learning on a larger LLM performed better than baseline clustering methods, but did not come close to human performance[2]., with an average number of matches of $0.3823$. This is an improvement from the word embedding model, which means that by leveraging the in-context learning ability of LLM, our model has obtained a better grasp of knowledge of the word and their corresponding relationships than what's capture by Word2Vec or Glove embeddings.

## 6.3 Finetuning and Distillation of Mistral-Instruct

We found that finetuning the small Mistral model on only the prompt and answers was insufficient for the model to improve linguistic and metalinguistic capabilities on *Connections*. It was even unable to grasp the output structure and the task of grouping 16 words into 4 disjoint, equal subsets. This caused it to omit and hallucinate words, which led to worse performance across the board in terms of matches, entropy, and IOU than the domain-specific, yet rudimentary K-Means method. This was unexpected and illustrates a distinct advantage of designing a task-specific model, as several research efforts listed in Section 3 did for the task of language puzzles.

While we originally hypothesized that distillation would both help the model understand the task better as well as obtain the linguistic and metalinguistic reasoning to complete it, we unfortunately found that only the former was true. Since the ground-truth rationales we obtained for DBRX were commonly in order of the answer groups, the Mistral model did learn to solve group-by-group without repeating words or responding with more or less than 4 groups. However, we found that distillation only slightly increased the average number of matches from 0.1176 to 0.1765, despite being successful at significantly decreasing entropy, since more groupings had 2 or 3 matching words. Because it achieves less than half of the 0.3823 average matches of in-context DBRX, distillation was unsuccessful in transferring task-specific reasoning skills to the LLM. This could be one of two reasons, assuming that distillation works for most specific tasks and small models, as shown in [ADD]. The rationales we obtained were after DBRX was already provided the answer. This caused the rationales to be less helpful in the case of reasoning from scratch, which we could fix by prompting the DBRX model in a different way to extract better chain-of-thought rationales. The alternate case is that because DBRX actually cannot solve the puzzle in the vast majority of cases, the rationales it gives are poor and cannot be significantly improved by prompting. Such is an example of a poor rationale for the answer grouping "CLOVER", "HORSESHOE", "MOON", "RAINBOW" with the description "LUCKY CHARMS", since all are marshmallows in the well-known *Lucky Charms* brand cereal. DBRX instead gives the rationale:

[2]This was not rigorously defined, but informally, most humans can usually solve the easiest grouping

> The third group of words can be formed by identifying the common theme of lucky charms. The words CLOVER, HORSESHOE, MOON, and RAINBOW are all considered to be lucky charms in various cultures. This group is formed by recognizing the specific category that these words belong to.

For this example, and others that require niche pop culture knowledge, all the methods struggle and even the larger DBRX struggles with reasoning after being given a correct grouping, cementing this as a distinctly human task that is difficult for LLMs to perform.

### 6.4 Verbal RL

Because of the complexities of this task, we found that the only method that achieved close to human-level performance was Verbal RL with DBRX on our fine-tuned Mistral-Instruct model. The critiques that DBRX gave were helpful inputs to the small model to improve upon itself. Here is an example critique on the grouping ["SIGHT", "SMELL", "TASTE", "DRESS"]:

> Your first group of words is related to the senses, which is a good start. However, you have missed including 'SIGHT' in this group. You could consider revising your group to include 'SIGHT' as it is also a sense like 'SMELL', 'TASTE', and 'TOUCH'.

After this critique was fed back into the Mistral model on the next iteration, it answered with the correct grouping. This shows that our model, after incorporating Verbal RL, is capable of leveraging effective reflection to solve some of the semantic grouping problems, but doesn't offer perfect solutions in all cases.

## 7 Conclusion

We discovered that *Connections* is a difficult task for language models to perform on their own, because of niche human associations between words and the metalinguistic reasoning needed to form certain groups. Our baseline K-Means clustering was a specialized but rudimentary method to solve semantic grouping, while more complex methods like finetuning a small model with fewer parameters failed at understanding the task itself.

Furthermore, our findings indicate that while in-context learning improved over baseline clustering methods, achieving human-like performance remains challenging. The integration of distilled knowledge and Verbal Reinforcement Learning showed promise, particularly the latter, which significantly enhanced model reasoning and accuracy through iterative feedback.

This task underscores the current limitations of LLM in matching the creative and flexible nature of human cognition. Future work could focus on more context-aware training methods that specialize on language puzzles and feedback mechanism enhancements for RL.

## 8 Ethics Statement

Our project introduces specific ethical challenges and potential societal risks. Firstly, there's a risk of reinforcing biases present in the training data, which can lead the model to amplify stereotypes found in language associations or word meanings. This could cause harm or perpetuate injustices, especially when sensitive attributes such as race, gender, or cultural backgrounds are involved. Secondly, there is the risk of misuse where, if extended beyond its intended recreational use, such technology could be used to manipulate linguistic data or generate misleading content.

To mitigate these risks, we propose several strategies. For tackling biases, it's possible to conduct regular audits of both the training data and model outputs, specifically focusing on detecting and correcting biased language associations. This could involve curating the dataset to ensure balanced representation and using techniques like adversarial training to make the model robust against learning biases. To prevent misuse, we will incorporate strict usage guidelines and access controls if the technology were to be deployed or integrated into broader applications.

## References


[1] Mary Beth Machuga Steven A. Stahl, Joyce L. Burdge and Sally Stecyk. The effects of semantic grouping on learning word meanings. *Reading Psychology*, 13(1):19–35, 1992.

[2] Patrick Pantel and Dekang Lin. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613–619. ACM, 2002.

[3] Alexander Strehl, Joydeep Ghosh, and Raymond Mooney. Impact of similarity measures on web-page clustering. *Workshop on Artificial Intelligence for Web Search (AAAI 2000)*, 58:64–73, 2000.

[4] Sugato Banerjee, Sugato Basu, and Srujana Merugu. A model-based approach for clustering topically related web documents. In *KDD Workshop on Text Mining*, volume 400, pages 275–291, 2005.

[5] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 33–41, 2009.

[6] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 3982–3992, 2019.

[7] Thomas K Landauer and Susan T Dumais. A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211, 1997.

[8] Geoffrey E Hinton and Sam T Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, pages 857–864, 2002.

[9] Josh Rozner, Christopher Potts, and Kyle Mahowald. Decrypting cryptic crosswords: Semantically complex wordplay puzzles as a target for nlp. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 11409–11421. Curran Associates, Inc., 2021.

[10] 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.

[11] Kevin Tam, Alexandra Rousseau, Akhila Saravanan, Charles Tucker, Jure Leskovec, and Christopher D. Manning. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 2013.

[14] Mikko Ilmari Malinen and Pasi Fränti. Balanced k-means for clustering. In *International Symposium on Intelligent Data Analysis*, pages 32–42. Springer, 2014.

[15] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901, 2020.

[16] Yoshua Bengio, Reza Yazdani Aminabadi, Saizheng Zhang, and Anirudh Goyal. Mistral: Efficient training of machine learning models with fine-grained computational steering. In *Advances in Neural Information Processing Systems*, 2021.

[17] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized llms. In *Proceedings of the Conference (Conference Name Here)*, page Page numbers if available. Publisher Name if available, 2023. Access it online at [URL].

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[19] Sondos Mahmoud Bsharat, Aidar Myrzakhan, and Zhiqiang Shen. Principled instructions are all you need for questioning llama-1/2, gpt-3.5/4, 2024.

# A In-context Baseline Prompt

```
"""
Given 16 items, 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.’’
Each puzzle has exactly one solution.
Watch out for words that seem to belong to multiple categories!
The following are three examples of 16 words and their corresponding answers.
You are also provided a brief description for what each group shares.
Learn from these examples and respond with just the answers for the following puzzle.
Respond in the same format as the examples: 4 lines, with a group on each line.
Follow the restrictions given at the end of the prompt.
The order within a group and between groups doesn’t matter, but the groups in the example answers and
descriptions are given in the order of easiest to trickiest to solve.

Examples:

Example 1.
HYPE, HITCH, AMP, ACOUSTIC, GAS, CABLE, ELECTRIC, SONIC, LINK, FIRE, WATER, COUPLE, HEARD, PUMP, AUDITORY, TIE

Answers:
CABLE, ELECTRIC, GAS, WATER
ACOUSTIC, AUDITORY, HEARD, SONIC
COUPLE, HITCH, LINK, TIE
AMP, FIRE, HYPE, PUMP

Descriptions:
Monthly bills
Related to sound/hearing
Connect
Excite, with ‘‘up’’

Example 2.
COMPLAINT, LAWSUIT, ACTION, HANGAR, FOXGLOVE, CLAIM, RUNWAY, WINDSOCK, TERMINAL, RING, GUMSHOE, CLUB, TURNCOAT, TARMAC, BEANBAG, TORCH

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

Descriptions:
Parts of an airport
Legal terms
Things a juggler juggles
Words ending in clothing

Example 3.
FESTER, SUNDAY, FRIDAY, ROT, LURCH, CAT, CHANCE, LIP, SPOIL, THING, THURSDAY, TURN, SATURDAY, WEDNESDAY, SOUR, TUESDAY

Answers:
FRIDAY, SATURDAY, SUNDAY, THURSDAY
ROT, SOUR, SPOIL, TURN
FESTER, LURCH, THING, WEDNESDAY
CAT, CHANCE, LIP, TUESDAY

Descriptions:
Days of the week
Go bad
The Addams Family characters
Fat ___

Now answer for these 16 words. Follow these restrictions for the output:
1. DO NOT include the descriptions. DO NOT have any "Descriptions:" text.
2. ONLY include the 16 words given. There should be NO OTHER WORDS.
3. DO NOT include any preceding text, like "Answers:", or line numbers, like "1.".
"""
```

# B Distillation and Fine-Tuning Prompts

Distillation prompt is as follows.

```
"""
You are a teacher teaching a student to play the following game.
```

```
Given 16 words, split them into 4 groups of 4 words, so each group is related by a certain theme or category.

Follow the restrictions at the end of the instruction.

Here are some hints.
    1. The 4 words can be items that belong in the same category.
    Example:
    Fish: BASS, SALMON, TROUT, FLOUNDER
    Body parts: HEAD, KNEES, SHOULDER, TOES

    2. The 4 words can each form a common phrase when paired with another word.
    Example:
    Fire __: ANT, DRILL, ISLAND, OPAL
    ___ Change: CHUMP, CLIMATE, LOOSE, SEA

    3. The 4 words can be associated with a certain verb phrase or action
    Example:
    Things to crack: EGG, KNUCKLES, SMILE, WINDOW
    Removes the covering of: PARES, PEELS, SHELLS, SHUCKS

You have an example game with a answer key and teacher's guide, and your task is to explain to a student how to arrive at
the solutions from the 16 words.

Here is the example game:
{Entries}

Answers:
{Answers}

Teacher's Guide:
{Descriptions}

Explain with clean and clear logic, refer to word-level details and explain how the groups of words are formed sequentially.
"""
```

Mistral Fine-tuning prompt is as follows.

```
"""<s>[INST]
# CONTEXT #
You are playing a linguistic game that requires chain-of-thought reasoning and has only one correct answer.
Given 16 words, split 16 words into 4 groups of 4 words, so each group is related by a certain theme or category.

Provide your reasoning for the groupings along with your answers.

Each of the 16 words is in EXACTLY ONE of 4 groups. Each group has EXACTLY 4 words. Think through this step by step.
You should start by identifying the easiest group, which is usually four items that belong in the same category of objects.

Here are some hints.
    1. A reason 4 words are in a group can be that they are items that belong in the same category.
    Example:
    Fish: BASS, SALMON, TROUT, FLOUNDER
    Body parts: HEAD, KNEES, SHOULDER, TOES

    2. A reason 4 words are in a group can be that they each form a common phrase when paired with another word.
    Example:
    Fire __: ANT, DRILL, ISLAND, OPAL
    ___ Change: CHUMP, CLIMATE, LOOSE, SEA

    3. A reason 4 words are in a group can be that they are associated with a certain verb phrase or action.
    Example:
    Things to crack: EGG, KNUCKLES, SMILE, WINDOW
    Removes the covering of: PARES, PEELS, SHELLS, SHUCKS

#########

# OBJECTIVE #
Provide your reasoning and the correct 4 groups of 4 words for these 16 words: input

#########

# RESPONSE FORMAT #
You answer should follow the following format exactly. Otherwise you will suffer consequences beyond imagination.

Respond with chain-of-thought reasoning for your grouping choices followed by "So the answer is:" and a JSON object with a single key
"answer" that has a list of 4 lists, each of which has 4 words.
Make sure each of the 16 words appears in EXACTLY one group and each group has EXACTLY 4 words.

DO NOT include anything after this JSON object.

#########
```

```
# EXAMPLE #
Given the 16 words:
SHALLOW, MAKE UP, COO, SURFACE, IMPROV, SIDE, AD-LIB, BABBLE, COSMETIC, CRAWL, DOMINO, FREESTYLE, PLACEBO, NURSE, BUTTERFLY, EXTERNAL

The correct response would be:

[REASONING_HERE]

So the answer is:

"answer": [["AD-LIB","FREESTYLE","IMPROV","MAKE UP"],["BABBLE","COO","CRAWL","NURSE"],["COSMETIC","EXTERNAL","SHALLOW","SURFACE"],
["BUTTERFLY","DOMINO","PLACEBO","SIDE"]]
[/INST]"""
```

# C Verbal RL Prompts

Supervisor (DBRX) prompt is as follows:

```
"""

You are a helpful assistant helping a student play a word game.

The instructions given to the student were:
###start of instructions###
instructions
###end of instructions###


The student's answer is:
response

And the student's reasoning is:
reasoning

Observe if the student's grouping of the words and corresponding reasoning are reasonable.

Your job is to talk to the student, comment on the student's current performance, and help the student improve the answer.

Format your response should address the student as if talking to him directly, starting with "You should notice these details:"
"""
```