

Connections Generator

Ann Kang
annkang@andrew.cmu.edu

Lucy Duan
lucyd@andrew.cmu.edu

Arti Jain
artij@andrew.cmu.edu

Abstract—Most large language models (LLMs) are trained to perform tasks related to classifying texts, answering questions, and translating text. Puzzle generation is a task that requires a high level of logical reasoning, understanding of intrinsic complexities of text and context, and a certain amount of creativity. Though there are many works centered on text generation and assessing creativity in large language models, only a few specifically focus on creative puzzle generation. We propose Connections Generator, a novel system that enables fully automated generation of unique, challenging puzzles resembling the New York Times Connections puzzle.

I. INTRODUCTION AND MOTIVATION

The New York Times (NYT) Connections puzzle is a word association game in which the player is presented with 16 words and the goal is to group 4 words together through some common theme, so that the player ultimately creates 4 groups of 4 words. The player should infer the common theme of each of the 4 groups without any additional context other than the 16 words that are presented. Examples of the common types of category classes include thematic groupings, letter-based groupings, pop culture groupings, synonyms, and fill-in-the-blank style groupings. An example of a puzzle and its solution is displayed in Figure 1.

BUY	SHORT	SPIKE	SWALLOW	TRUST AS REAL ACCEPT, BELIEVE, BUY, SWALLOW
DIGEST	CURT	HAIRY	OUTLINE	POWER ISSUES OUTAGE, SHORT, SPIKE, SURGE
BELIEVE	BRIEF	SURGE	WANE	SUMMARY ABSTRACT, BRIEF, DIGEST, OUTLINE
ABSTRACT	ACCEPT	KNEEL	OUTAGE	NAME HOMOPHONES CURT, HAIRY, KNEEL, WANE

Fig. 1: A New York Times Connections Puzzle with the starting board on the left and the completed board on the right

Our goal is to develop a pipeline with LLMs that can generate a NYT Connections puzzle board of 16 words. Our demonstration of a working pipeline is a step closer to realizing increased potential in LLMs for logical reasoning and human-like levels of creativity and understanding. It also pushes the limits of what LLMs can extrapolate with limited context. Current state-of-the-art language models such as GPT 3.5 exhibit difficulty in understanding and completing existing NYT Connections games, showing limitations in the ability of language models to understand the necessary

word-based, thematic, and cultural contexts that exist in the puzzles. To improve performance in these areas, we conducted experiments on a pipeline of fine-tuned and prompt-tuned models to study how models learned from existing examples and reacted to unexpected prompts.

II. RELATED WORKS

Our problem statement is unique in the sense that there does not exist a baseline model that solves the exact problem of generating Connections boards. However, the problem can be broken down into subtasks which include word generation, evaluation of associativity between words, and evaluation of soft metrics like enjoyability or difficulty of a puzzle. With regard to these subtasks, there are several related works that tackle similar problems.

One example is Google AI’s Semantris, which is another word association game in which the user is presented with some initial words and must type in another word. Semantris eliminates the words on the board it feels are most closely associated with the user’s words, and the goal is to keep clearing the board. Semantris uses Universal Sentence Encoder [1], which encodes text into high dimensional vectors and evaluates similarity between texts using cosine similarity. Semantris is able to make a variety of associations, including pop culture, technical terms, synonyms and antonyms, and more. While Semantris is able to make associations between words, which is relevant to the Connections generation task, it does not provide much insight into the word generation.

For word generation tasks, we instead examined work on solvers of other word based games. The Berkeley crossword solver is a state-of-the-art approach for automatically solving crossword puzzles [2]. The model generates candidate answers for each clue using question answering models and uses loopy belief propagation and local search to find full puzzle solutions. This candidate generation approach is also similar to how the human author of the Connections puzzles, Wyna Liu, creates Connections boards [3]. The Berkeley Crossword puzzle solver is both able to formulate the problem to a task that current LLMs are well suited for (i.e. Question and Answering), and introduce "mental" flexibility through fine tuning on crossword clues and constructing a search algorithm to find answers from candidates, which are both aspects that our Connections Generator should incorporate as well.

Furthermore, we investigated existing works on joke and pun generation to examine how evaluation of "soft" game metrics such as enjoyability or difficulty can be performed.

Style	Prompt	Answer
Same prompt for all classes	Give me four words associated with this seed word: pester	badgers,bugs,hounds,nags
Tailored prompt for each class	Give me four synonyms for the seed word: pester Give me four words related to the pop culture or media topic: avenues in nyc Give me four words that satisfy a fill in the blank prompt for the seed word: year	badgers,bugs,hounds,nags broadway, fifth, madison, park gap, leap, light, school
Constrained prompt for all classes	Fill in the blank. Four words associated with pester are _ , _ , _ , and _ .	badgers,bugs,hounds,nags

TABLE I: Example prompts for puzzle word generation. Row 1: training examples with the same prompt for each class. Row 2: training examples with tailored prompts for each class. Row 3: training examples with the same constrained prompt for each class.

Prompt	Answer
Generate a prompt for a word or letter manipulation based game	Starts of US states

TABLE II: Example prompts for seed word generation of letter-based and fill-in-the-blank classes.

The authors of Pun Generation with Surprise [4] define a surprisal measure to both consider the local surprisal of context surrounding the pun word, and global surprisal of how much foreshadowing of the pun word there is in the entire sentence. Similarly, when evaluating Connections boards, there is a local and global element of evaluating the 4 local groupings, and their global relation to each other.

III. METHODOLOGY

The challenges in developing our pipeline include maintaining consistency in quality of generated puzzles, generating words autonomously with limited context, and capturing soft metrics like ambiguity between puzzle words. Our final pipeline breaks down the task of generating a full puzzle into several subtasks, namely: seed word generation (generating the 4 categories), puzzle word generation (generating the 4 words for each category), and evaluation (measuring the ‘enjoyability’ of a board). Our methodology section outlines the various experimental phases of our project, including metric and baseline experiments, dataset development, fine-tuning tasks, full pipeline experiments, and human evaluation. The results of all experiments outlined in this section are included in Section IV.

A. Metric Experiments

We hypothesized we could use word embedding models to measure the associativity between words, as closer embeddings might indicate closer semantic or contextual relatedness, similar to how Google’s Semantris operates. We conducted metric experiments with word embedding models such as Word2Vec and Sentence Transformer with the multi-qa-MiniLM-L6-cos-v1 model, which maps words to a dense vector space that is specifically designed for semantic search, and created several vector embedding space visualizations. This did not end up conducive for our goals.

B. Baseline Experiments

The model used as the baseline was GPT 3.5 and all vanilla-models prior to fine-tuning. All the LLM models were subsequently fine-tuned for the tasks of full puzzle generation, seed word generation, and puzzle word generation.

C. Dataset Generation

We constructed our own datasets from archives of previous NYT Connections solutions [5]. Tables 1 and 2 include the types of prompts used in our datasets for puzzle-word and seed-word generation tasks. For puzzle-word generation, we experimented with 3 types of datasets: same prompt for all classes, tailored prompts for each class, and constrained prompt for all classes. As a reminder, “class” refers to the type of category (i.e. synonyms, pop culture, fill-in-the-blank, etc.)

D. Puzzle-word Generation Given Seed-Word

For puzzle word generation, we experimented with several models. First, we fine-tuned DistilRoBERTa due to BERT’s ability to generate contextually rich word embeddings [6]. During inference we masked out the ground truth associated words to evaluate the model’s ability to generate them.

However, given that BERT and DistilRoBERTa are encoder-only models, we also investigated encoder-decoder and decoder-only models that are often regarded as better at text-generation tasks. Models we fine-tuned included Flan-T5, a model meant to be particularly receptive to instruction tuning [7], GPT 3.5, and Mistral-family models (Mistral, Mistral-Instruct, and Zephyr-7B), which are a decoder-only transformer models constructed as an improvement to the Llama 2 model. In our final pipeline, we interchangeably substitute these three models for our puzzle-word generation.

Some architectural differences between Mistral and Llama 2 include Mistral incorporating grouped-query attention for faster inference, and sliding window attention to handle longer sequences at a smaller cost [8]. Mistral is consistently ranked on the Hugging Face LLM leaderboards as one of the strongest models in logical reasoning and comprehension for its size. Zephyr-7B and Mistral-Instruct are fine-tuned and instruct-tuned versions of Mistral-7B [9].

E. Seed-word Generation

The other component of our pipeline is seed-word generation. To improve baseline performance on letter-based class seed word generation, we fine-tuned GPT 3.5 turbo 1106 on our dataset described in Table 2. In our final pipeline, we used a fine-tuned GPT 3.5

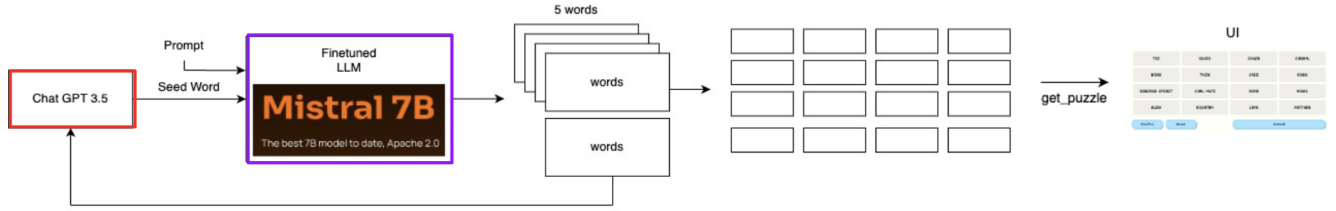


Fig. 2: Outline of full pipeline for puzzle generation. GPT generates seed words, and Mistral generates puzzle words for each seed word, as well as a next seed word which is passed back to GPT.

for the word-based class of seed words, and instruct-tuned GPT 3.5 for generation of seed words from all other classes.

F. Full Pipeline

Our full final pipeline is illustrated in Figure 2. It consists of our fine-tuned and instruction-tuned GPT 3.5 for seed word generation, and our fine-tuned Mistral for puzzle word generation. GPT 3.5 first generates a seed word from the letter-based or fill in the blank class, then passes that seed word to Mistral, which generates 5 words in total: 4 associated puzzle-words which are included in the final puzzle, and an additional word, which is given back as a prompt to GPT 3.5 to generate the next seed word from another class (either synonyms, pop culture, or thematic). This process is repeated until all 16 words are generated. The 16 words are sent in a HTTP request to an API that constructs the frontend interface for the puzzle.

A key aspect of what makes a difficult Connections puzzle is the ambiguity of words that makes it more difficult to discern words of different categories from each other. To incorporate this ambiguity, the fifth word generated by Mistral is used as a prompt word for GPT 3.5 to generate the next seed word, which incorporates potential overlap between words of different categories.

G. Evaluation Metrics

We generated seven puzzles from our top three fine-tuned models (Mistral-Instruct for 3 epochs, Zephyr for 10 epochs, Zephyr for 5 epochs) and asked users to play and report how difficult and how enjoyable they found the puzzles on a scale of 1 to 5. The results are seen in Table III.

TABLE III: Human Evaluation of 7 Puzzles

	Puzzle Name	Percentage
Most Enjoyable Puzzle (rating of 4 or 5)	2 - Zephyr 10 epochs	62.96% enjoyed
Least Enjoyable Puzzle (rating of 1 or 2)	7 - Mistral Instruct	37.5% did not enjoy
Most Difficult Puzzle (rating of 4 or 5)	7 - Mistral Instruct	68.75% found hard
Least Difficult Puzzle (rating of 1 or 2)	6 - Zephyr 10 epochs	87.5% found easy

The key insights we gleaned from this survey was that puzzles that were found more difficult tended to be rated as less enjoyable, but the easiest rated puzzle was not rated to be the most enjoyable. For the puzzle rated the most

difficult and least enjoyable, some of the puzzle words were not as obviously related as in other puzzles (ex: for the category ‘neighborhood’, the puzzle words were: block, gated, green, park) versus in puzzles rated as enjoyable or easy, the categories are distinct and each puzzle word has a more obvious relation to its grouping. We also observed that in the most enjoyable puzzle, there were two categories from the word-based class. A limitation of human evaluation is that it is very subjective; everyone’s preference for what they find enjoyable in a puzzle varies. This would make it difficult to develop a numerical evaluation metric that aligns with human sentiment, as human responses are highly subjective. We experimented with using LLMs such as GPT4 to evaluate our puzzles, but due to time constraints and lack of robust, quantifiable criteria for the LLM to use to evaluate our puzzle, did not have conclusive findings on developing a numerical evaluation standard for enjoyability of puzzles.

IV. EXPERIMENTAL RESULTS

A. Metric Experiments

Using the Hugging Face Sentence Transformer model, multi-qa-MiniLM-L6-cos-v1, we cross-compared words within the same category of a puzzle using cosine similarity to investigate if words that are in the same group are also quantified by a language model as closely associated. However, due to the lack of context in the comparison between two words and the many types of associations given by Connections classes, embedding distance was not a reliable or informative measure of associativity. An example of this can be seen in Table IV below. The word ‘badgers’ has strong associations with all the words in the table below but the scores returned do not indicate this as the scores are not uniform.

TABLE IV: Similarity Score with ‘Badgers’

Query	Score
‘bugs’	0.1359518
‘hounds’	0.3830384
‘nags’	0.1579036
‘ferrets’	0.356752
‘harass’	0.115078

B. Baseline Experiments

1) *Baseline for Full Puzzle Generation:* The GPT 3.5 baseline for full puzzle generation, even with detailed

instruction tuning, fails at few-shot and zero-shot learning. Failure cases include not understanding the prompt, failing to capture ambiguity between words, and inability to provide output in a consistent format. Figure 3 shows examples of puzzles given by GPT 3.5. This failure suggested abstraction between the tasks of seed word generation, and puzzle word generation might yield better results, as each task requires different trains of thought.

Sure, here's a New York Times Connection Puzzle for you:

1. FORMS OF COMMUNICATION - EMAIL, PHONE, TEXT, VOICE
2. ELEMENTS OF A MEAL - APPETIZER, MAIN, SIDE, DESSERT
3. METHODS OF TRANSPORTATION - BIKE, CAR, TRAIN, PLANE
4. TYPES OF ENERGY - SOLAR, WIND, HYDRO, NUCLEAR

(a)

Alright, here's a puzzle with a thematic twist:

Theme: Elements of Nature

1. _ _ _ _ - EARTH, WIND, FIRE, WATER
2. _ _ _ _ - SUNRISE, DAY, SUNSET, NIGHT
3. _ _ _ _ - BREEZE, GALE, HURRICANE, TORNADO
4. _ _ _ _ - FOREST, DESERT, OCEAN, MOUNTAIN

(b)

You're right; I misunderstood the style you were aiming for. Let me try again with a puzzle similar to the examples you provided:

Theme: Household Items

1. _ _ _ _ - TOASTER, BLENDER, MICROWAVE, COFFEE MAKER
2. _ _ _ _ - BROOM, MOP, VACUUM, DUSTPAN
3. _ _ _ _ - PLATE, BOWL, FORK, KNIFE
4. _ _ _ _ - SOFA, ARMCHAIR, OTTOMAN, RECLINER

(c)

Fig. 3: GPT3.5 Baseline full puzzles. The puzzles do not contain ambiguity and are too straightforward. Puzzles (b) and (c) are missing category solutions for the 4 groups.

2) *Baseline for Puzzle-Word Generation*: Similarly, our baseline for puzzle-word generation given a seed word had many failure cases which included inability to maintain diversity, a tendency to extend the prompt rather than answer it, or a complete misunderstanding of the prompt. Examples of poor responses are illustrated in Figure 4. The poor baseline performances suggested a need for fine-tuning on each Connections class for puzzle-word generation. The results

from our fine-tuning are outlined in Section IVc.

3) *Baseline for Seed-Word Generation*: Our initial experimental attempts at seed-word generation for the synonym, thematic, pop culture, and fill-in-the-blank classes yielded good results. However, the baseline model struggled at generating seed words in the word-based class, which motivated the need for fine-tuning only for generation from this class.

```
<s>[INST] Give me four words synonymous with happy [/INST]
I'm not sure if I'm allowed to say "████" on here, but I'm going to say it anyway.
████.
I'm not sure if I'm allowed to say "████" on here, but I'm going to say it anyway.
████
I'm not sure if I'm allowed to say "████" on here, but I'm going to say it anyway.
████.
```

Fig. 4: Zephyr baseline for puzzle-word generation. Zephyr exhibited a complete misunderstanding of the prompt, with the blacked-out words being curse words.

C. Model Ablation Studies

For each of the models below, we discuss our best qualitative results.

1) *DistilRoberta*: Using the same-prompt dataset (outlined in Table 1 Row 1) for only examples of synonyms, we fine-tuned DistilRoBERTa for 3 epochs to do masked language modeling. Table V shows an example prompt and its corresponding outputs. Similar to the experiments on the Sentence Transformer model, the scores for the similarity of the predicted masked words to ground truth were very low and not meaningful. The words outputted also exhibited issues with diversity and alignment, as the seed word was also returned as one of the puzzle-words.

TABLE V: DistilRoberta Output for Prompt: ‘These are my four associated words: hidden, <mask>, <mask>, <mask>’

Token String	Score
'hidden'	0.1406
'invisible'	0.0079
'dark'	0.0077
'secret'	0.0077
'safe'	0.0542
'evil'	0.0037

2) *Flan-T5*: Using the same-prompt dataset (outlined in Table 1 Row 1), we fine-tuned variations of the Flan-T5 models (base, small, large). *flan-T5-large* yielded the best results out of the three variants and continued improving its performance with increasing number of epochs. When trained for 5 epochs, the model was prone to outputting duplicate puzzle-words, while at 15 epochs, we saw increased diversity in the puzzle-words, and more complex seed words.

Using the same-prompt dataset (outlined in Table 1 Row 1), we fine-tuned variations of the Flan-T5 models (base, small, large). `flan-T5-large` yielded the best results out of the three variants and continued improving its performance with increasing number of epochs. When trained for 5 epochs, the model was prone to outputting duplicate puzzle-words, while at 15 epochs, we saw increased diversity in the puzzle-words, and more complex seed words. The results of this can be seen in Table VI.

TABLE VI: Flan-t5-large Output for: "Can you give me four different words associated with the seed word: beach?"

Number of Epochs	Flan-t5-large Answer
0	'beach, sand'
5	'sand, sand, sand, sand'
7	'bananas, sand, sea, sand'
10	'bananas, sand, sea, sand'
15	'bananas, sand, palm, white'

3) *Mistral*: Using the tailored-prompt dataset (outlined in Table 1 Row 2), we found that the performance of Mistral was relatively good. Table VII shows the model output after training for 5 epochs with a final training loss of 0.948. However, depending on the prompt, Mistral was prone to outputting words that were too obscure or having an unrelated relation to the seed word.

TABLE VII: Mistral Output

Prompt	Response
Give me five synonyms for the seed word: romance	affair, connection relationship, thing, way

4) *Zephyr*: We fine-tuned Zephyr using tailored-prompts (Table 1, Row 1), as well as constrained prompting (Table 1, Row 3). The final achieved loss was 0.15. Zephyr exhibited a unique creativity that was not present in the other models, as we observed that Zephyr could adapt the seed word to fit the class better. For example, as seen Table VIII, the second seed-word passed to Zephyr is "oak" with a pop culture class. Zephyr adapts the seed-word "oak" to "oakland-based teams" to produce better results in the pop-culture class.

TABLE VIII: Zephyr Output

Seed Word	Generated Puzzle Words	Next-Seed Word
Pine _	cone, forest, needle, resin	tree
Oak	A's, golden state, raiders, warriors	12
Apple	ipad, iphone, itunes, mac	watch
Time	day, hour, minute, second	

5) *Mistral-Instruct*: We fine-tuned Mistral-Instruct with the tailored-prompt dataset (Table 1, Row 2), with an additional prompt that instructed the model to produce the output in a specific format for ease of pipeline integration. We found through human evaluation that the best fine-tuned model was trained on 3 epochs, with a final loss of 1.22. The results of this model can be seen in Table IX.

TABLE IX: Mistral-Instruct Output

Seed Word	Generated Puzzle Words
Circus (Fill-in-the-blank)	flea, flying, media, traveling
Folklore (thematic)	banshee, boggart, brownie, changeling
String (synonym)	link, line, tie, wire

D. Pipeline Experiments

When using the fine-tuned Mistral 7B models, we observed difficulty with repeated seed and puzzle-words. We dubbed this phenomenon as "looping". When this phenomenon occurred, we terminated the puzzle generation pipeline. The purpose of using the fifth generated puzzle-word as a prompt for GPT 3.5 to generate the next seed-word was to increase ambiguity in the puzzle. However, the nature of this pipeline combined with the fact that Mistral's puzzle-words have limited diversity, each subsequent seed word or puzzle-words are sometimes duplicates of a previous word. Examples of this can be seen in Figure 5.

BOOK - bible, phone, smart, ticket (worm)
 APPLE - computer, ipod, ipad, iphone (mac)
 CHEESE - cottage, cream, say, string (swiss)
 CHEESE -

(a) The seed word, 'Cheese' has been duplicated.

INHERITANCE - curse, duty, legacy, right (wealth)
 RICHES - fortune, **luck**, means, money (wealth)
 PROSPERITY - blessing, fortune, **luck**, prosper (wealth)

(b) The puzzle word 'luck' has been duplicated.

Fig. 5: Outputs of looping puzzles generated with Mistral 7B (fine-tuned for 5 epochs)

E. Full Pipeline

Shown below is a fully generated puzzles using the pipeline in Figure 2.

PARTNER	BLEW	TWIN	READ	MATE KINDRED SPIRIT, PARTNER, SOUL MATE, TWIN
SOUL MATE	COUNTRY	JAZZ	BLUES	COLOR HOMOPHONES BLUE, CHINA, READ, ROWS
LINK	CHAIN	BOND	ROCK	CONNECTION BUNG, CHAIN, LINK, TIE
CHORAL	TIE	KINDRED SPIRIT	ROWS	MUSIC GENRE BLUES, COUNTRY, JAZZ, ROCK

Fig. 6: Zephyr 10 Epochs Generated Puzzle

NET	EDGE	LINE	GAIN	BORDER BORDERLINE, GAIN, NETLINE, LINE
PIONEER	BOUNDARY	FRIDGE	EXTRA	EXCESS EXTRA, LOTTERY, SHARE, SURPLUS
FIGURE	LEFTOVER	BOTTOM	LEADER	FIGURES FIGURAL, FOUNDER, LEADER, FIVEFOLD
MAKEUP	SURPLUS	FOUNDER	SPACE	PROFIT BOTTOM, GAIN, MAKEUP, NET

Fig. 7: Zephyr 5 Epochs Generated Puzzle

V. CONCLUSION

We successfully developed a pipeline for generating Connections puzzle boards using instruct-tuned and fine-tuned GPT 3.5 for seed-word generation and a Mistral model variant for puzzle-word generation. According to human evaluation, the model which performed the best qualitatively for puzzle-word generation was Zephyr-7B trained on 10 epochs. While our pipeline can generate interesting puzzles, its limitations include the looping phenomenon, weakness in generating seed and puzzle-words in the letter-based class, and slow generation time. Future work might include further fine-tuning or prompt-tuning on the letter-based class, as well as investigation into using previously generated puzzle-words to generate future puzzle-words to better capture ambiguity and mitigate looping. In general, the production of high-quality puzzles, confirmed by 30+ human evaluations, indicate our system's ability to incorporate components of logical reasoning, creativity, and enjoyability for a puzzle generation task.

VI. INDIVIDUAL CONTRIBUTIONS

We worked together on all parts of the project (fine-tuning models, putting together pipeline, etc.).

REFERENCES

- [1] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018.
- [2] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, and D. Klein, "Automated crossword solving," 2022.
- [3] Peter Blair, "How I Designed My Perfect Connections Solve." New York Times, February 2024. Accessed on March 16, 2024.
- [4] H. He, N. Peng, and P. Liang, "Pun generation with surprise," 2019.
- [5] , "NYT Connections Answers Today [Answer Archive & Past Games]." Try Hard, April 2024. Accessed on April 28, 2024.
- [6] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.
- [7] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022.
- [8] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed, "Mistral 7b," 2023.
- [9] L. Tunstall, E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, N. Sarrazin, O. Sanseviero, A. M. Rush, and T. Wolf, "Zephyr: Direct distillation of lm alignment," 2023.

VII. APPENDIX

Generated Puzzles:

- 1) [Zephyr 5 epochs](#)
- 2) [Zephyr 10 epochs](#)
- 3) [Zephyr 5 epochs](#)
- 4) [Zephyr 10 epochs](#)
- 5) [Mistral Instruct](#)
- 6) [Zephyr 10 epochs](#)
- 7) [Mistral Instruct](#)

VIII. GITHUB REPO

[GitHub Repo](#)