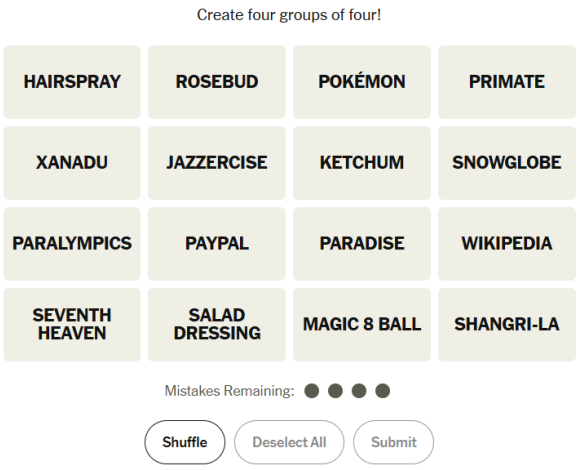
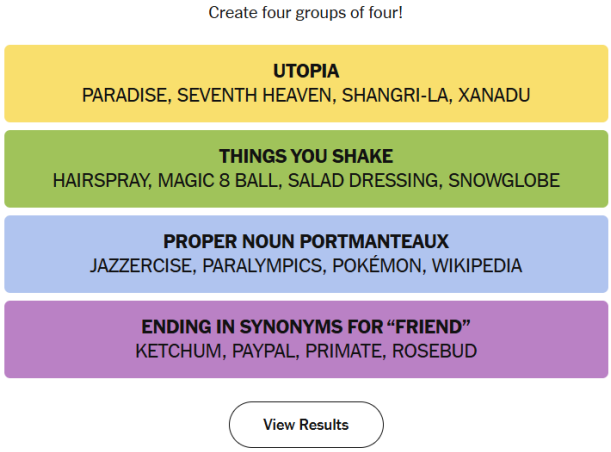
Triplet prompting: Combining Chain of Thought Prompting with Triplet Generation for Connections in New York Times

Swapnika Dulam
szd5775@psu.edu



(a): Connections puzzle



(b) Solutions to puzzle in (a)

Figure 1. Connections puzzle and solutions

1. Task

The Connections is a daily word puzzle in the New York Times [1] has been captivating many players with its tricky ways to come with solutions. This game was first introduced on June 12th, 2023 [2]. In this game as shown in Figure 1 (a), a set of 16 words are displayed as tiles in a 4x4 grid. The player must identify four groups of four words from these sixteen words. The player has four chances to do so. The player can shuffle the grid to get new ideas on word associations.

Since it is a recent game, there is limited research in this area. Solving this game is a test for the player's ability to identify connections between the words that are not apparent and least expected. The game deviates seasoned players by sometimes choosing the most apparent visible connection to be the theme word. A theme word is a set of four words which highlight the theme of the puzzle, but the individual words belong to four different categories. The difficulty of word associations increases with color of group. Yellow being the most straightforward followed by green, blue and purple which is supposed to be the trickiest.

The ability to solve this game is represented by abstract reasoning [3] and testing the available SOTA models to be able to solve this game will prove models' capability in understanding English language and applying abstract reasoning and thinking skills to solve the puzzle. This project aims to solve the connections puzzle using different techniques and compares success rates..

2. Related Work

LLMs seem like an obvious choice for language related tasks and the methods proposed in [4] are based on pairwise cosine similarity of sentence encodings of all words combinations and chain-of-thought (CoT) prompting approach to SOTA LLM models on a set of 250 puzzles. The sentence embedding models utilizing a cosine similarity strategy outperformed GPT-3.5 Turbo, solving 11.6% of puzzles. For the next method they observed that GPT-4 Turbo achieved a 38.93% success rate using Chain of Thought prompting.

[3] used a few-shot Chain-of-Thought prompting on a dataset of 200 words using more advanced LLMs like Gemini 1.5 Pro, Claude 3 Opus, Llama 3 70B, ChatGP-4o and compared the performance. Additionally, they compared some results with few human players and categorized the knowledge required to solve various puzzle types. They too report that at the most 18% of puzzles were solved by the best LLM and cannot beat expert human players.

While [5] is related to the same connection puzzle, their focus is more towards using LLMs for puzzle generation under various difficulty levels and compare it to the puzzles generated by humans. The game also has a lot of amateurs trying to solve it using GPTs available to them, some of which are posted on reddit. In [6], they attempt to solve the puzzle using ChatGPT manually (without API and version is unspecified) for a 60-day period and report that

ChatGPT was successfully able to solve 39 puzzles fully. All these works are prompt based techniques relying on the LLM's ability to identify facts from LLMs internal knowledge.

A relation-based key-value pair generation to get more relevant information from an LLM for a given word is proposed in [7]. A triplet-based knowledge crawling was also discussed in [8] for cognitive architecture. The triplet is of the form (word, relation, related to). Combining the above two approaches, we can query LLMs to get knowledge triplets which can be more accurate and less susceptible to hallucinations. For example: the word sandal can have triplets like (sandal, instance of, footwear), (sandal, has suffix, wood) etc.

So, in this project I try to employ various techniques learnt from previous homework assignments and above papers to solve the connections game and compare the methods highlighting the benefits and shortcomings of each method.

Specifically, I used the following approaches displayed in Table 1 to solve the puzzle and compare their results

Notes		Method description
[4] pt 1 MPNet	(i)	MPNet for text embeddings
[8]	(ii)	Semantic Similarity of Wikidata triplets
	(iii)	Similarity for image embeddings from CLIP
	(iv)	Fusing image & text embeddings from CLIP
[3] llama3	(v)	CoT Prompting llama 3
[7]	(vi)	Semantic similarity of triplets by llama 3
[3]+[7]	(vii)	Combining (v) and (vi)

Table 1: Approaches used to solve connections puzzles.

3. Approach

3.1. My Strategy

I have been solving this connections puzzle for the past four months and my research is based on extracting inherent knowledge from large language models. So, I implicitly used the techniques from my research based on [7],[8] to solve the puzzle which has improved my chances of winning.

When I solve the game, I think of all the attributes that are associated with the word. Attributes like size, shape, color, category, common prefixes, common suffixes, popular terms in TV, movies, books etc. My success rate with the game has been 50% so far. Trying to guess the connections (the puzzle setters had in mind) in a mere four attempts is not easy. Moreover, we just use single attempt in this paper to validate the approaches.

3.2. Semantic similarities

Trying to find similarity between word meanings is achieved by semantic similarity [9]. The first method proposed by authors in [4] is also based on semantic similarity between every set of four words selected from the given sixteen words.

For **(i), (ii), (vi)** semantic similarity is used to solve the puzzle. In **(i)** the code provided by [4] is replicated for part 1 using their best quoted result using MPNet for my dataset. In **(ii), (vi)** both of which are based on triplets generated by wiki data [10] querying for **(ii)** and LLM prompting to llama 3b 8192 using Groq's free API [11] for **(vi)**. While second approach in [4] generates text embeddings for $16C_4$ = 1820 combinations and then finds pairwise similarity using 4 models (Bert, Roberta, MPNet, and MiniLM), I generated text embeddings using MiniLM [12] for the words along with their triplets and then computed a pairwise similarity matrix and later performed a k-means clustering to group the words into four groups of four words each.

3.3 Vision Models

Since, a lot of puzzle categories are based on visual appearances like shape, size, color etc. Using images to find similarities between words is an option. By using multiple images for the same word and then finding similarities between visual features can be done to find connections between words.

Initially, I used google-images-search [13] to search images for the given word. However, the free API limits exceeded after doing a search for data from 6 puzzles (~100 requests). So, I later used Stable Diffusion model [14] to generate images from the given text. These images were then used to generate image embeddings using CLIP [15] in **(iii)**. Another variant using the words as text embeddings and fusing them with the generated image embeddings was done in **(iv)**. The clustering algorithm for these embeddings was similar to semantic similarity, by performing k-means clustering to group the words into four groups pf four words each.

3.4 LLM Prompting

Finally, as mentioned in [3] Chain-of-thought LLM prompting was replicated based on prompts from [3],[4] in **(v)**. But since both the papers used models that are not free or open source, I used the same prompt templates with the free Groq API mentioned in section 3.2 for Llama 3 model.

For **(vi)** I created a new template based on [7] with prompt specific to this use case and generated triplets

using the same llama 3 model's Groq API. These triplets were analyzed in the same way as **(ii)** mentioned in section 3.2.
For **(vii)** I combined prompt template from [3],[7] to come up with a new prompt template. The results for which will be discussed below.

3.5 Implementation
Only the prompt templates from [3],[4] were used for **(v)** and code for **(i)** was from [4]. The remaining (ii) to (vii) was mostly the code that was created for this project.

Step 1: Data creation.
(ii) - Data was pulled from wikidata API using SPARQL graph query language.
(iii), (iv) - Data (images) was created using Stable diffusion model [14] on my local machine.
(v), (vi), (vii) - Data was pulled by sending prompt to Llama3 API from Groq.

Step 2: Data Processing.
(ii) - Triplets from wikidata were further processed to combine (word, relation, related word) -> (relation related word) and duplicates were removed.
(iii), (iv) – Image embeddings were generated using CLIP and for (iv) were fused with text embeddings of the word.
(v), (vi), (vii) – Data from llama3 response was processed to pick the triplets for (vi) and categorized groups for (v), (vii) and added to new json files for validation. (Due to erratic responses from LLMs some of the responses were not received in specified template. I manually parsed and populated categories to verify the results)

Step 3: Clustering
(ii), (vi) – Semantic similarity for the text and triplets is performed followed by K Means for clustering the words into four groups of four words each. The grouped words are added to json files for validation.
(iii), (iv) – The clustering is also performed on embeddings using K Means to find four groups of four words each. The grouped words are added to json files for validation.

Step 4: Validation
All the data generated from the above steps is validated against the data set to find number of matches.

4. Dataset
Data for this puzzle is available in the free website connections archive [16] or in [1] with a subscription. [3],[4] also have provided with dataset. [3] reported 200 puzzles in their paper but later released data for 443 puzzles. [4] have released data of 250 puzzles. For this paper, I have used the exact dataset of [3] with 443 records, but the results are reported for the first 200 records like the paper reported and also to enable fair comparison amongst the papers.

5. Results
5.1 Replicated results
The results from paper [3],[4] are major baselines for this project. I replicated results from [4] for approach one, which used text embeddings, shown in Figure 2. I was able to get the same results as those of the paper [4]. For the second approach where they used LLM prompting, I was not able to replicate because of ChatGPT API access which is paid and not free.

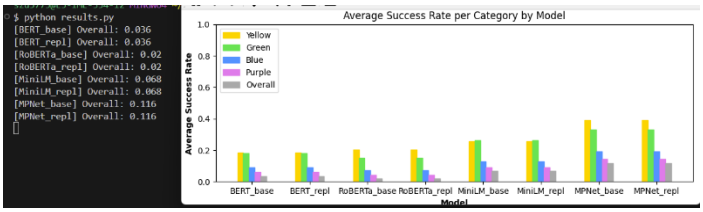


Figure 2: Replicating results from [4] approach 1.

For paper [3] also, they used all paid APIs for the results. Instead, I replicated the results with both the prompts from [3],[4] on GROQ's free llama 3 API. For paper [3], these results are considered as my baseline for **(v)**. For paper [4], the API generated responses, which are in the git repository in a json file, but the model did not output in the requested format as a result of which parsing the response from the model has become very difficult (Despite changing the response format in the prompt and using same model for other scenarios). So, I was not able to validate those responses

5.2 Results from testing the seven approaches
The results from the methods (i) to (vii) were evaluated in one shot against the correct answers for 200 puzzles. The results are displayed in Table 2.

	Method description	Count (#/ 800)	Success rate	Success rate reported by papers
(i)	MPNet text embeddings	98	12.25%	11.60%
(ii)	Sem Sim of Wikidata triplets	79	9.88%	
(iii)	Sim for img emb	5	0.63%	
(iv)	Fusing img and txt emb	11	1.38%	
(v)	CoT Prompting llama 3	492	61.50%	22%
(vi)	Sem sim of triplets llama3	24	3%	
(vii)	Combining (v) and (vi)	504	63%	

Table 2: Results from the approaches

Every puzzle has 4 categories that need to be discovered, so the total number of categories or groups is 800. I used the number of correctly classified puzzle groups as the metric (count which is out of 800 total groups to be identified in 200 puzzles) to measure success of the model. Specifically, the
success rate = Count / Total %.

Additionally, these results are compared with the results reported in their respective papers. The better results in this experiment in (i) may be due to different data, because the model in [3] was tested on 250 pairs. It is astonishing to see the vast difference in performance on Llama 3 model for the same hyper parameters and the same data. Finally, the new combined approach proved to identify the highest number of group categories in one-shot. Figure 3 represents the same data graphically.

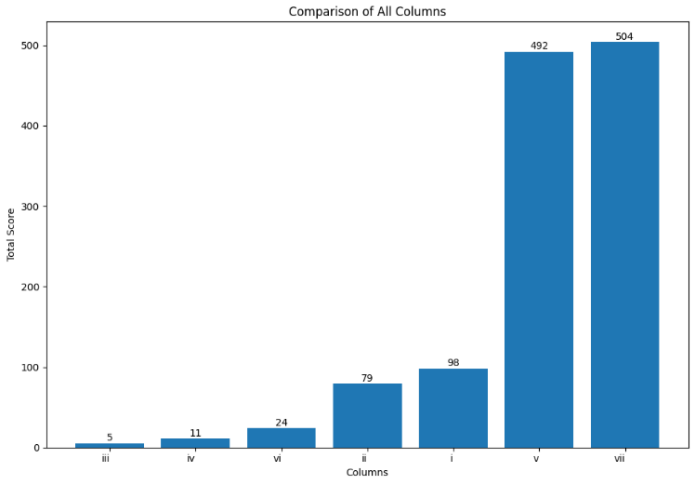


Figure 3: Comparing the groups identified by all approaches.

Further, I group the responses provided by the model to identify the total number of puzzles solved. A puzzle is considered solved, if all four categories have been accurately identified in a single attempt. A puzzle is partially completed if 1 or 2 or 3 groups were correctly identified. Figure 4 compares (v), (vii). The approach in (v) was able to solve higher puzzles completely, however my approach failed in less number of puzzles. In real-time a partially solved puzzle is easier to solve in second or third or fourth guess.

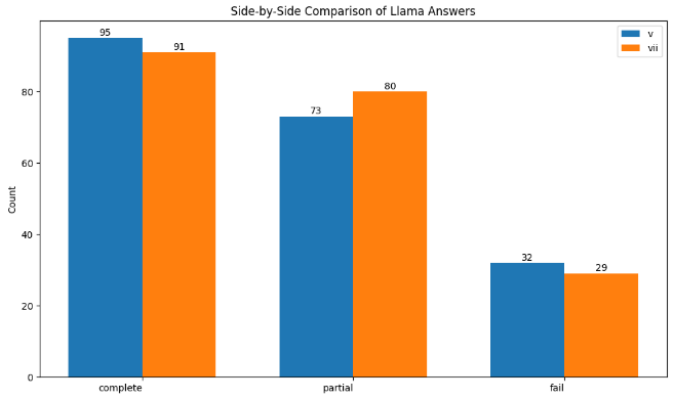


Figure 4: Comparing puzzle wise results for approaches (v) and (vii)

6. Possible Improvements and Results

Although the puzzle gives four chances to players to solve it, I just use the one-shot approach as opposed to iterative prompting (where immediate feedback of the first guess is given, to choose a better combination on the next guess),

Reasons to choose One Shot Approach:

- 200 puzzles: Because I am dealing with 200 puzzles as opposed to solving a single puzzle as in [6], it is efficient to compare all the models at once.
- Multiple technique comparison: Because I evaluated multiple techniques a common ground was to choose one shot to be fair to all the methods.
- Lack of resources: Most of the LLMs mentioned in papers were accessed through APIs, which had rate limits if they were free or were paid. Because I was dealing with 200+ records a four-shot attempt would make more API calls, pushing towards paid access.

Benefits of iterative prompting:

This prompt from (vii) can be easily adapted to make it into an iterative style of prompting. Iterative style prompting involves incorporating feedback provided by puzzle, to choose a better set to finish the puzzle. This is especially helpful when three words of one group were chosen. In that case, picking the fourth word closest to the three words already picked is relatively easy. This feedback to an LLM can help with a better choice of pairs. An example of this scenario is provided in Figure 5. The words “SENT”,” SPAM”,” TRASH” belong to the email-folders category but “PHISH” does not. Therefore, it is easy to observe that the fourth word is “DRAFTS”.

Figure 5: Depicting selection of three words of one category.

Additionally, Puzzle on 12/12/24 as shown in Figure 6 was an image-based puzzle, which can be solved in a better way by extracting embeddings from them using (iii), (iv). This would eliminate inaccurate image generation by Stable diffusion and generate better groups. However, this would have an extra step of image identification.

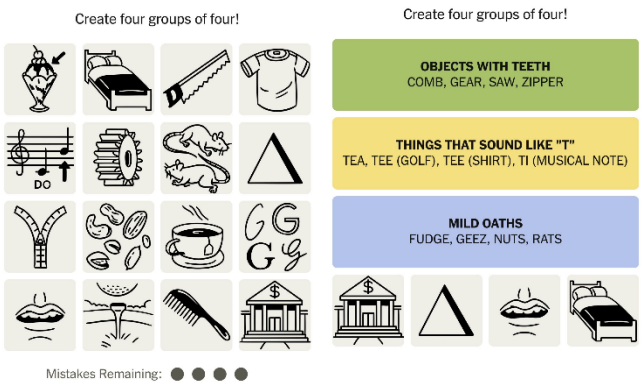


Figure 6: Image based connections on 12/12/24.

The approach of combining Chain of Thought prompting along with triplet generation proved to have the highest success rate in given puzzles. By using this prompt template, I had 100% success in solving the connections puzzles in the last few days, given that I have four attempts to solve the puzzle. More time, and access to paid resources can definitely help get a higher success rate on puzzles.

7. Code Repository

Github link: **https://github.com/SwapnikaD/VL_Project**
Drive link for folders too large for github: https://drive.google.com/drive/folders/11mWaQl7lpYUt_phm-IhjQx3trf5MYqm-?usp=drive_link

References

1) Connections - Group words that share a common thread. Connections. https://www.nytimes.com/games/connections
2) Liu, W. (2023). How our new game, connections, is put together. https://www.nytimes.com/2023/06/26/crosswords/new-game-connections.html
3) Samadarshi, P., Mustafa, M., Kulkarni, A., Rothkopf, R., Chakrabarty, T., & Muresan, S. (2024). Connecting the Dots: Evaluating abstract reasoning capabilities of LLMs using the New York Times Connections word game. arXiv. https://doi.org/10.48550/arxiv.2406.11012
4) Todd, G., Merino, T., Earle, S., & Togelius, J. (2024). Missed Connections: lateral thinking puzzles for large language models. arXiv . https://doi.org/10.48550/arxiv.2404.11730
5) Merino, T., Earle, S., Sudhakaran, R., Sudhakaran, S., & Togelius, J. (2024). Making New Connections: LLMs as puzzle generators for the New York Times' Connections word game. arXiv. https://doi.org/10.48550/arxiv.2407.11240
6) Can ChatGPT solve NYT's connections?. https://crossword-solver.io/chatgpt-vs-connections/
7) Cohen, R., Geva, M., Berant, J., & Globerson, A. (2023). Crawling the Internal Knowledge-Base of Language models. arXiv. https://doi.org/10.48550/arxiv.2301.12810
8) Salvucci, D. (2014). Endowing a cognitive architecture with world knowledge. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 36, No. 36).
9) Wikipedia contributors. (2024, November 6). Semantic similarity. Wikipedia. https://en.wikipedia.org/wiki/Semantic_similarity
10) Query Wikidata (n.d). https://query.wikidata.org/
11) GroqCloud. (n.d.). https://console.groq.com/playground?model=llama3-8b-8192
12) Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020). MINILM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. arXiv.. https://doi.org/10.48550/arxiv.2002.10957
13) Google-Images-Search 1.4.7. (n.d.). https://pypi.org/project/Google-Images-Search/
14) Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2021). High-Resolution Image Synthesis with Latent Diffusion Models. arXiv. https://doi.org/10.48550/arxiv.2112.10752
15) Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. arXiv. https://doi.org/10.48550/arxiv.2103.00020
16) Archive - Connections NYT game. (n.d). https://connectionsgame.org/archive/