LLMike: Exploring Large Language Models' Abilities in Wheel of Fortune Riddles

Ejdis Gjinika[1,*], Nicola Arici[1], Andrea Loreggia[1], Luca Putelli[1], Ivan Serina[1] and Alfonso Emilio Gerevini[1]

[1]*Università degli Studi di Brescia, Via Branze 38, Brescia, Italy*

Abstract

A riddle from the game show "Wheel of Fortune" consists of a hidden sentence that can be discovered starting from a simple clue and by iteratively guessing its letters. Although the game is very popular and intuitive, solving one of these riddles is not trivial. In fact, for interpreting the clue, identifying the most probable letters, and leveraging the game's mechanics effectively, a player requires linguistic abilities, world knowledge, and even some form of strategic thinking. The goal of this study is to verify whether Large Language Models (LLMs) possess the necessary abilities to solve Wheel of Fortune riddles. We propose a software framework called LLMike in which an algorithmic Game Master interacts with an LLM: prompting it, enforcing the game's rules, updating the hidden sentence based on the model's guesses, and evaluating their correctness. We study several models with different sizes, evaluating their performance, behavioural patterns, and common types of errors. Our dataset and code are available at https://github.com/ejdisgjinika/LLMike.

Keywords

Large Language Models, Wheel of Fortune, Model Evaluation, Benchmarks

1. Introduction

Assessing linguistic and reasoning abilities of Large Language Models (LLMs) is an open challenge [1, 2, 3, 4]. Especially in the last few years, LLMs have proved to address many Natural Language Processing tasks (such as text classification, summarization, machine translation, etc.) and their benchmarks, with performance that previously seemed unreachable. However, LLMs come with several limitations, such as hallucinations [5], reasoning issues [6], and lack of trustworthiness [7, 8]. Therefore, researchers have started developing new methods or more challenging tasks to assess different types of abilities that LLMs may or may not possess [9, 10, 11].

A popular research line is based on games [12, 13], especially text-based games such as word association games [14, 15] or crossword puzzles [16, 17] which focus on linguistic aspects. For instance, in a crossword puzzle LLMs would obviously need linguistic abilities to interpret the clues and to insert all the words correctly. Moreover, the clues may refer to general knowledge and trivia, which must be known by the LLM. However, this game does not need particular reasoning capabilities, such as for choosing which words to complete first: LLMs may start wherever they want and complete the puzzle with knowledge alone.

With non-textual games, such as Connect-4 or Tic-Tac-Toe [12, 18] we can have a different situation. In fact, both of these games require a more refined strategy to win. For instance, Connect-4 is a game in which two players compete with each other. They insert coloured disks into a board, trying to form a line (vertical, horizontal, or diagonal) of four disks of the same colour, while preventing the other player from doing the same. In order for an LLM to win, clearly it would need a solid strategy to choose all its actions in a specific order, to evaluate the situation on the board and consider all its options.

Addressing linguistics, knowledge, and strategy, in this work we propose a task based on the popular "Wheel of Fortune" game show. An example of how this game works is available in Figure 1. In order to win, a player has to guess a sentence from a simple clue. At first, only the number of words and the number of letters for each word are available. Next, the player has to spin a wheel (into which each wedge gives a different amount of money) and say a consonant which will be revealed in the hidden sentence (if present). With some of the money earned, the player can decide to buy a vowel, which will make the guess easier. This procedure can be repeated several times until the player decides to guess the hidden sentence. If the guess is correct, the player effectively takes the money and the overall goal is to accumulate as much money as possible. To solve this task, of course, an LLM would need linguistic capabilities to understand the rules, expressed in natural language. World knowl-

CLiC-it 2025: Eleventh Italian Conference on Computational Linguistics, September 24 — 26, 2025, Cagliari, Italy

*Corresponding author

ejdis.gjinika@unibs.it (E. Gjinika); nicola.arici@unibs.it (N. Arici); andrea.loreggia@unibs.it (A. Loreggia); luca.putelli@unibs.it (L. Putelli); ivan.serina@unibs.it (I. Serina); alfonso.gerevini@unibs.it (A. E. Gerevini)

0009-0006-9817-5846 (E. Gjinika); 0009-0000-9713-6630 (N. Arici); 0000-0002-9846-0157 (A. Loreggia); 0009-0008-5055-6812 (L. Putelli); 0000-0002-7785-9492 (I. Serina); 0000-0001-9008-6386 (A. E. Gerevini)

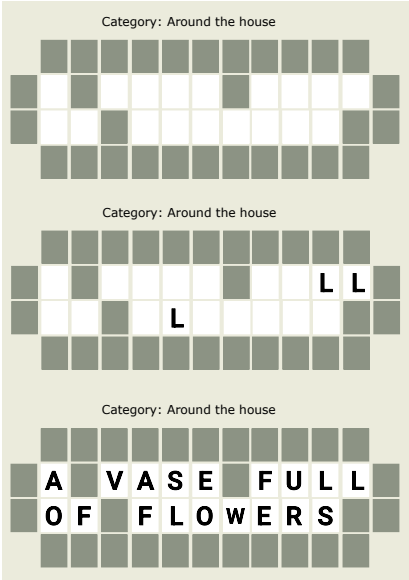

Figure 1: Example of the gameplay of the Wheel of Fortune game. At the top, we show how the game starts, i.e., with a completely hidden riddle. In the middle, we show the partially completed riddle after one participant spins the wheel and chooses the letter "L". At the bottom, we show the solution of the game.

edge is also needed to solve many of the clues based on places, movies, etc. Finally, choosing which consonants to say, whether to buy a vowel, or when to try to guess the sentence also needs some basic strategic skills.

In this paper, we create LLMike, an algorithmic framework that allows LLMs to play Wheel of Fortune games. The name comes from the TV presenter of the first editions of the Italian version of Wheel of Fortune, Mike Bongiorno. LLMike prompts the LLMs with all the procedures of the game and interacts with it depending on its responses. The framework allows simple budget management and the checking of different types of errors. We tested both open-source and commercial models to see whether these models are capable of completing such difficult tasks. We manually created a dataset based on some publicly available riddles. Finally, we analysed the answers provided by the models in order to understand their behaviour in the games they won, their main errors, and to give some insight into their strategy.

2. Related Work

Games and puzzles are a recurrent testbed for assessing the capabilities of deep learning systems, especially to implement complex reasoning abilities [16, 13, 15, 19, 20]. For instance, Wallace et al. [16] use a neural network approach combined with a local search to choose possible word candidates and rank them for completing crossword puzzles. This game covers different aspects, such as common sense, general knowledge, and metalinguistic patterns. Another work on crossword puzzles with human evaluation has also been proposed in [17]. The authors of [14] propose a challenge in which participants submit systems for the "Ghigliottina", an Italian text game where some semantic knowledge is needed to link a group of words. Most of the proposed systems are based on techniques that leverage the similarity between the vector representations of words.

With the growing popularity of LLMs, rather than creating ad-hoc models to play and complete games, researchers have begun using these games to benchmark the general abilities of LLMs [21, 22]. Qiao et al. [20] introduce the concept of evaluating LLMs using conversational games, such as a round-based interaction between a questioner and an answerer called Ask-Guess. One of the main claims of this study is that conversational games can differentiate the capabilities of different LLMs. Manna et al. [13] assessed that the leading commercial models (i.e. GPT-4 and Gemini-Pro) struggle in completing a semantic connection game such as the "Ghigliottina" [14]. A similar work was presented by Samardashi et al. [15], focusing on the New York Times Connections word game, which similarly requires semantic knowledge.

Another interesting work is [23], which focuses on role-playing abilities of LLMs combined with external tools. Similarly, the authors of [19] evaluated the abilities of several LLMs in a multi-agent scenario to solve a detective-style game. Although linguistic and world knowledge are needed, their evaluation focuses more on the strategies the agents use to play the game.

More generally, the knowledge possessed by LLMs has been the subject of many studies [24], focusing on world knowledge [25, 26], semantics [27] and specific knowledge, such as the medical domain [28].

3. Methodology

In this section, we explain how we structure our evaluation of the capabilities of LLMs in Wheel of Fortune riddles. First, we describe the original rules of the game; then, we describe our adaptation and implementation of the game.

3.1. Wheel of Fortune

As introduced earlier, the Wheel of Fortune is a game show that lets multiple contestants compete with each other to win the game and earn money. The goal is to correctly guess an hidden riddle by iteratively discovering its letters until the player is confident enough to formulate

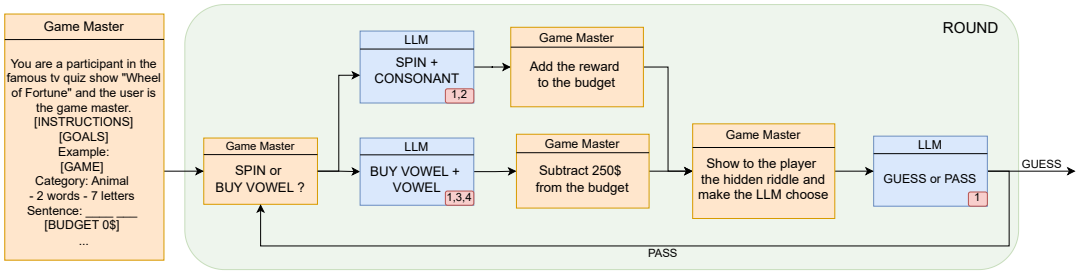

Figure 2: Interaction schema of LLMike. In orange, we show the actions of the Game Master, in blue, we show the actions of the LLM that plays the game. The first Game Master block shows a brief of the prompt given to the LLM at the beginning of each game. All the LLM blocks also report, in the bottom right corner, the rule numbers that the LLM has to follow to complete the action correctly (Section 3.2).

a guess. The game works in several rounds. In the beginning, it is shown the word puzzle (with no letters present, as at the top of Figure 1) which can reveal a sentence, a name of a person, a place, etc. Each participant has a budget that starts at 0 $ and can gradually grow over the rounds. Starting from the first participant, he/she can spin a wheel composed of several wedges, with different amounts of money associated with each wedge. Next, the participant chooses a consonant: if the consonant is revealed in the hidden riddle (as in the middle of Figure 1), the participant earns the amount of cash indicated by the wedge times the number of occurrences of the consonant chosen. Next, he/she can spin the wheel again and continue to play another round. If the consonant is not present in the riddle, the participant passes the turn to another player. As the rounds progress and the player has enough money, he/she can buy a vowel for a fixed amount of the budget and has to indicate which vowel he chooses. If the vowel is present in the riddle, it will be revealed, but if it is not, the player passes the turn. At any time in his/her game, the player can guess the riddle by giving their final solution. If the correct answer is given, the player wins the budget he earned. However, if the answer is wrong, the player passes the turn.

In the original game show, some special wedges of the wheel are also present: “Bankrupt”, which resets the player’s budget and passes the turn; and “Lose a turn”, which makes the player skip his/her turn.

3.2. LLMike: Evaluating LLM’s Abilities at Wheel of Fortune

In the adaptation we created for evaluating LLMs’ abilities at solving Wheel of Fortune riddles, we defined two main roles: the Game Master, which is a specifically coded algorithm (not based on artificial intelligence tools) that interacts with the LLM and evaluates its answers, and the LLM, which acts as a player of the game.

An overview of our adaptation is presented in Figure 2. The Game Master gives the prompt, which contains the rules, the goals, and an example of the game, and asks the LLM to select an action, starting a round. The LLM selects an action and its budget is updated. Next, the Game Master shows the new conditions of the game, i.e. the hidden riddle partially revealed and the new budget. Finally, it asks the LLM to provide a guess or pass to the next round.

We redesign the game by adapting the rules to a single-participant scenario with a slightly different round structure, as shown in Figure 2. First, we removed the special wedges from the wheel (i.e., “Bankrupt” and “Lose a turn”), because they depend only on luck, and this can lead to a non-systematic analysis of the LLM’s abilities. Therefore, our wheel has only cash wedges, all between 100 $ and 1.000 $.

In our interaction schema, first the Game Master asks the LLM to spin the wheel or to buy a vowel for 250 $. After the choice made by the LLM, the riddle and the budget are adjusted accordingly and subsequently communicated to the LLM. Then, the LLM has the option to give a guess or to pass and start another round. Since we have only one LLM playing, a key difference is that in our adaptation of the game, if the LLM gives a letter that is not present in the riddle, it does not lose the turn in favour of another player, but only its budget is set to 0 $. The goal we give to the LLM is to complete the game and to maximize the amount of money earned by solving the riddles. These goals are in line with the goals a real player playing the Wheel of Fortune would have.

We also formalize some rules specifically for the LLMs’ interaction with the game, intending to control and better understand the ability of the models to follow instructions. This formalizations results in four rules:

- **Rule 1**: The LLM cannot choose to do an action that is not possible in a given situation; for instance, the LLM can’t pass the turn when it is required to spin the wheel or buy a vowel.

- **Rule 2**: If the LLM spins the wheel, it has to choose a consonant and not a vowel.
- **Rule 3**: If the LLM buys a vowel, it has to choose a vowel and not a consonant.
- **Rule 4**: The LLM has to buy a vowel if and only if it has enough money to do so.

If the model violates one of the rules, it will automatically lose the game.

In Figure 2 also shows a brief version of the prompt used during the games. The prompt contains a short description of the context, followed by the instructions for playing the game, the goals, and an example. The goals are expressed in simple sentences, and the examples represent a standard conversation between an LLM and the Game Master. The complete prompt is available in the GitHub repository.[1]

Please note that the riddle cannot be solved by simply choosing all the letters in it, one at a time. In fact, all riddles are composed of consonants and vowels. However, the player can choose only consonants, which leads him/her to always deal with an incomplete riddle. This leads to two major possible decisions: buying vowels or guessing the sentence, which cannot be easily implemented in simple baseline approaches.

4. Experimental Evaluation

In this section, we present how our experiments were conducted, the models and data we used, how the performance was evaluated, and the results. Then, we present an analysis of the main errors made by the models and provide some intuition on their strategy.

Models and implementation details. We selected 29 open-source models available through Ollama[2], which are available in Table 1. Ollama is a framework designed to facilitate the local execution of open-source LLMs. The models considered differ considerably in terms of architecture, family, and number of parameters.

Moreover, we select three commercial models: GPT-4.1, Mistral Large 2 and Gemini 2.0 Flash[3]. The exact size of GPT-4.1 and Gemini 2.0 Flash has not been disclosed publicly. However, they are much bigger than any of the open-source models we considered. Mistral Large 2 has about $123B$ parameters.

For both open source and commercial models, the responses are generated using the default parameters.

[1] https://github.com/ejdisgjinika/LLMike

[2] https://ollama.com/

[3] Specifically, we use the "mistral-large-2411", "gemini-2.0-flash-001", and "gpt-4.1-2025-04-14" snapshots.

Table 1
List of the open-source models tested on our task. For each model, we consider its standard and quantized versions provided by Ollama.

Model	Size
Aya Expanse	32B
Cogito	3B, 8B, 14B, 32B
Command-R	35B
Gemma	2B
Gemma 2	2B, 9B, 27B
Gemma 3	1B, 4B, 12B, 27B
Llama3.2	1B, 3B
Mistral Small	24B
Mistral Small 3.1	24B
Olmo 2	7B, 13B
Phi 3	3.8B, 14B
Phi 4	3.8B, 14B
Qwen 2.5	0.5B, 1.5B, 3B, 7B, 14B

Data. Our dataset is composed of 80 riddles in English taken from a publicly available dataset[4] and repurposed. The riddles are of variable length and divided into 16 categories. The shortest sentence is made up of 2 words while the longest is made up of 9 words. In terms of the number of characters, the range is from 9 to 47 characters. The average lengths are 19.47 and 3.16 in terms of characters and words, respectively.

Metrics. Several metrics were introduced to measure the performance of LLMs in our Wheel of Fortune task. First, we consider the number of games won (# Wins) and the average amount of money won by the LLM (Total Final Budget). Other metrics are more complex and are based on the game rules listed in Section 3.2. First, we consider a group of metrics to evaluate the model behaviour, such as the number of letters chosen by the LLM (# Letters), the percentage of the letters that were actually found in the riddle (% Correct Letters), and the percentage of completion of the riddle when the LLM gives the right guess (% Riddle Completion). Next, we consider several error-related metrics, to understand when the model does not follow the rules (perhaps, by not selecting a letter, or by trying to buy a vowel with an insufficient budget), when it just provides a wrong guess or when it reaches the maximum number of possible consonants.

4.1. Results of the Best Performing Models

In this section, we report the performance of LLMs in the Wheel of Fortune game. Of the more than 30 mod-

[4] https://www.kaggle.com/datasets/darrylljk/wheel-of-fortune-answers

Table 2
Results for the best performing models, ordered by the number of games won (# Wins). In the first four rows, we report the results for the open source models, whereas in the last three rows we report the commercial models. In the columns we report the average number of letters chosen (# Letters), the percentage of the correct letters (% Correct Letters), the riddle completion percentage at the moment of giving the guess (% Riddle Completion) and the average final budget obtained (Total Final Budget).

Model	# Letters	% Correct Letters	% Riddle Completion	Total Final Budget	# Wins
Gemma 3 27B	11.00	62.73	71.64	20.6K	20
Gemma 2 27B	8.38	68.66	71.30	5.55K	8
Phi 4 14B	14.12	62.83	85.21	4.35K	8
Gemma 3 12B	16.80	51.19	86.46	2.45K	5
GPT-4.1	10.53	67.99	71.27	65.7K	62
Gemini 2.0 Flash	13.23	64.15	81.66	24.6K	35
Mistral Large 2	12.08	54.97	69.73	15.25K	25

els tested, only 9 managed to guess at least one solution: three commercial models and six open-source LLMs, four of which belong to the Gemma family. Except for Gemma 2 9B, all models have more than 10B parameters. Furthermore, all models with more than 25B parameters can guess at least one correct solution, with the exception of Aya Expanse and Command-R.

In Table 2, we show the results ordered by the number of games won. The best open-source model, by far, is Gemma 3 27B with 20 wins in 80 games, followed by Gemma 2 27B and Phi 4 14B with 8 wins, and Gemma 3 12B with 5. Although they reached one and two victories, respectively, we did not include in Table 2 Gemma 2 9B and Cogito 32B due to the low significance of their results with such a small sample.

However, these victories can come from two different abilities. The first is that a model may guess as many letters as possible and progressively fill in the riddle, until the guess becomes very simple. The second is that a model may not need to fill the riddle as much as possible, because it has enough knowledge to find the correct solution of a more complicated riddle. Analysing the ability of the model of choosing letters, the best open source model is Gemma 2 27B, with 68.7% of correct letters. This ability is reflected in the number of letters required to provide a correct solution, which is 8.38, the lowest of all models. The other LLMs perform worse, ranging from 51.19 (Gemma 3 12B) to 62.73 (Gemma 3 27B). All the other open-source models tend to select a higher number of letters, ranging from 11.00 to 16.8. Interestingly, the former has the tendency to select as many letters as possible, filling the riddle up to 86.46%, on average.

Analysing the guessing capabilities, Gemma 3 27B obtains 20 victories not only by selecting letters, but also by guessing from a quite low completion of the riddle (71.30), whereas the least performing models require a higher completion. Instead, Phi 4 14B requires an average 85.21 completion to solve a total of only 8 games. This may suggest a higher understanding and knowledge possessed by Gemma 3 27B, with respect to Phi 4. A similar comparison can be made with Gemma 3 12B, which obtains only 5 wins with a riddle completion of 86.46. In this case, the difference seems entirely dependent on the different number of parameters.

Significantly better results are obtained with commercial LLMs: GPT-4.1 gets 62 wins, Gemini 2.0 Flash 35, and Mistral Large 2 25. Nevertheless, these models have similar performance with respect to the open-source models in terms of number of letters (all between 10.53 and 13.23), percentage of correct letters (which does not exceed 68%), and percentage of riddle completion. This behaviour suggests that although these larger models possess a similar ability in guessing the correct letters and completing the masked riddle, they are much better at providing the correct solution.

Table 2 also reports the final budget earned by the models. The best performing model is GPT-4.1, with more than 65K \$. Notably, Gemma 3 27B obtains a higher amount of money (20.6K) with respect to Mistral Large 2 (15.25K), despite obtaining fewer wins (20 versus 25). Since every time a model chooses a wrong consonant, the budget is set to 0, this is probably due to its higher percentage of correct letters (62.73 versus 54.97).

4.2. Typical Errors

In this section, we discuss the most common errors made by the models considered. Since, an important first result of our experiments is that 23 LLMs over a total of 32 were unable to give a single correct solution, we first analyse their main flaws.

In Figure 3, we show six types of errors made by those LLMs considered and their frequency calculated for all 80 games. The most common error (in blue) is definitely *Insufficient Budget* (33.1%), in which an LLM tries to

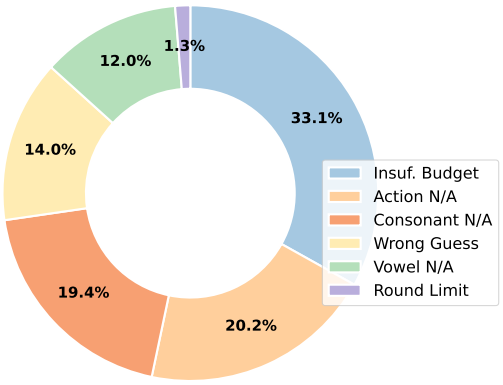

Figure 3: Error frequency for the LLMs unable to guess a single riddle. Each colour represents a different error category. The frequency of each error, in the form of a percentage over all the 80 games for each LLM, is reported inside each sector.

buy a vowel without the necessary money. The next error, *Action Not Allowed (N/A)*, is quite more complex. As we show in Figure 2, the model is forced to generate specific text such as [SPIN], [BUY VOWEL] or a single consonant at different times during the game. This text indicates the choice of executing a specific action in a strict way and any other answer is considered as an Action N/A error. This error recurs 20.2% of the time. Similarly, *Consonant N/A* (19.4%) refers to those times that the model, after choosing to buy a vowel, selects a consonant instead. Both Action N/A and Consonant N/A denote a lack of understanding of the game rules and of the prompt instructions provided by the Game Master. *Wrong Guess* (14.0%) happens when the model simply provides a wrong solution to the riddle. In our analysis, an important aspect of this type of error is that often the LLM does not respect the format of the riddle, selecting words with the wrong number of letters. Moreover, some models (such as Olmo 2 and Llama 3.2) can be considered "overconfident", choosing to guess the solution with a very limited amount of letters. As *Vowel N/A* (12.0%), we refer to those times the model, instead of choosing a consonant, selects a vowel instead. As for Action N/A and Consonant N/A, this error depends on not understanding the game rules. Finally, the remaining 1.3% of the errors occur when the model exceeds the round limit imposed (20 rounds), continuously spinning the wheel or buying vowels without trying to guess the solution of the riddle.

Table 3
Analysis of the letter chosen by the models. For the best performing models, we report the number of different first pairs (# Pairs) and first triplets (# Triplets) of letters provided by the model. We also report the mean number of vowels bought (# Vowels)

LLM	# Pairs	# Triplets	# Vowels
Gemma 3 27B	11	28	2.30
Gemma 2 27B	9	15	2.38
Phi 4 14B	35	61	4.00
Gemma 3 12B	22	40	3.80
GPT-4.1	9	25	2.63
Gemini 2.0 Flash	10	24	3.31
Mistral Large 2	17	39	2.52

Overviews of the Error Made by the Best Performing Models In the following, we investigate the flaws made by the best performing models, i.e. those reported in Table 2. Starting from GPT-4.1, the major cause its losses is the Wrong Guess (55.56%): i.e. the model, at a certain riddle completion, has enough "confidence" to try to guess the riddle but provides the wrong answer. Despite GPT-4.1 being the best model at following the instructions, it still shows some limitations on letter choosing (11.11% of Vowel N/A and 5.56% of Consonant N/A) and managing the budget (11.11% of Insufficient Budget Error). Gemini 2.0 Flash shows a different behaviour in terms of errors. In fact, it manifests lots of problems on instruction adherence and budget management (respectively 40% of Instruction Error and 33.3% on Insufficient Budget Error). Interestingly, Mistral Large 2 is good at following instructions, managing its budget and choosing the letters in the right contexts. However, it provides many wrong answers (Wrong Guess 87.27%). An interesting fact is that Mistral Large 2 and Gemma 3 27B obtain a comparable number of wins (respectively 25 and 20 wins) even if they have a significantly different number of parameters (123B and 27B respectively). Although Gemma 3 27B has a lower percentage of Wrong Guess (56.7%), its limitations in dealing with single letters (Vowel N/A 20% and Consonant N/A 5%) and budget management (10%) deteriorates its performance.

4.3. Hints on Strategy

In this section, we report some information regarding the strategy followed by the best performing models.

We think that a total absence of strategy would result in picking random consonants. Instead, a smarter approach would be to select consonants which appear frequently in English words. To highlight this behaviour, we analyse the first letters chosen by the model. Results are available in Table 3, in which we report:

Table 4
Frequency of the five most common consonants in the English language (Std. Freq. column) and relative choosing frequency for the best open source LLM (Gemma 3 $27B$) and commercial LLM (GPT-4.1).

Consonant	Std. Freq.	Gemma 3	GPT-4.1
T	9.1	10.40	10.08
N	6.7	10.40	9.69
S	6.3	9.49	10.59
H	6.1	4.29	3.49
R	6.0	11.83	9.82
Total	34.2	46.41	43.67

- the number of different pairs of letters chosen by the LLM at the start of the game (*# Pairs*);
- the number of different triplets of letters chosen by the LLM at the start of the game (*# Triplets*);
- the number of *# Vowels* the model decided to buy;

We can see that there are notable differences among the models with respect to the number of distinct pairs and triples chosen at the start of different games. Phi 4 $14B$ has the highest variability, selecting 35 different pairs and 61 different triples of letters across the 80 riddles in our dataset. Instead, the best performing models (such as GPT-4.1, Gemini 2.0 Flash and Gemma 3 $27B$) present a much lower variability, with respectively 9, 10 and 11 different pairs and less than 30 different triples. This suggests that they start many riddles with a similar strategy.

Analysing the number of vowels bought by our models, we can see some other relevant information. The models with highest variability in terms of letters chosen (Phi 4 $14B$ and Gemma 3 $12B$) also tend to buy more vowels (respectively, 4.00 and 3.80 on average). Comparing these results with those in Table 2, we can see that this strategy does not provide notable advantages: in fact, they win only 8 and 5 games respectively. Instead, the best performing models (the commercial models and Gemma 3 $27B$) tend to buy fewer vowels (only 2.30 for Gemma 3 $27B$ and 2.63 for GPT-4.1) obtaining a definitely higher number of wins. Moreover, since buying vowels requires subtracting 250 \$ from the budget, this decision can be considered good also for the declared goal of maximizing the earnings.

In Table 4 we compare the standard frequency of the first five consonants in the English language[5] (Std. Frequency) with the percentage of times that such consonants are chosen by two LLMs: the best performing open source one, Gemma 3 $27B$, and the best commercial one, GPT-4.1. We can see that the most frequent consonants (which in English are *T*, *N*, *S*, *H*, and *R*) are definitely those generated more frequently by the models. In fact, considering Gemma 3 $27B$ these consonants are the 46.41% of all the letters chosen by the model. Similarly, for GPT-4.1 they are 43.67%. Although this differs from the standard frequency in the English language (into which these five consonants reach a total of 34.2%), we can say that both models know which are the most common consonants and exploit this information in their games, combining both linguistic knowledge and basic strategy. Both models have a very similar behaviour, with T, N, S and R being the preferred consonants (with a frequency around 10%), and H is considered less important, with a frequency that does not exceed 4%. This is quite different from the statistics calculated for the English language, in which H has a frequency of 6.1, quite similar to *R* (6.0), and *S* (6.3). This is probably due to the fact that H is very present in very common stop words such as *the*, *which*, *this*, which may not be particularly important to solve our riddles. More specifically, models tend to start with the two most frequent consonants (*T*, *N* or *S*) and then buy a vowel (mostly *E* or *A*). This behaviour is constant for most of the 80 riddles of our dataset, regardless of the sentence length or other characteristics.

5. Conclusions and Future Work

In this paper, we proposed a novel textual game based on the famous "Wheel of Fortune" game show with the aim of assessing linguistic and reasoning abilities. We created a framework for allowing LLMs to play under strict rules and showed how the task was structured, the data, and the metrics used for the evaluations. We analysed 29 open source models and 3 commercial models to evaluate a variety of models with different model's architecture and sizes. Only 9 LLMs out of 32 managed to solve at least one riddle. The most problematic aspects are their little ability to follow the instructions, such as the constraint of choosing only consonants. The best performing open-source model is Gemma 3 $27B$, with 20 wins out of 80 riddles, whereas the commercial model GPT-4.1 solves 65 riddles. Analysing their strategy, we see that the best performing models select the most frequent consonants in the English language, resulting in a progressively easier riddle. However, they can also guess the right solution with a completion of around 70%.

As future work, we want to analyse performance of Large Reasoning Models (LRM), such as Deepseek-R1, o3 and o4-mini, and to expand the framework to let several models play with each other. Moreover, another interesting direction would be to exploit Multimodal LLMs to create a visual version of the game. We would also like to consider data in other languages. Finally, we would like to implement new games and analyse the behaviour of models in a more complex environment.

[5] https://en.wikipedia.org/wiki/Letter_frequency

Acknowledgments

This work was carried out while the author, Ejdis Gjinika, was enrolled in the *Italian National Doctorate on Artificial Intelligence* run by Sapienza University of Rome in collaboration with the University of Brescia.

This work has been partly funded by Regione Lombardia through the initiative "Programma degli interventi per la ripresa economica: sviluppo di nuovi accordi di collaborazione con le università per la ricerca, l'innovazione e il trasferimento tecnologico" - DGR n. XI/4445/2021.

References

[1] A. Y. Uluslu, G. Schneider, Investigating linguistic abilities of LLMs for native language identification, in: R. Muñoz Sánchez, D. Alfter, E. Volodina, J. Kallas (Eds.), Proceedings of the 14th Workshop on Natural Language Processing for Computer Assisted Language Learning, University of Tartu Library, Tallinn, Estonia, 2025, pp. 81–88. URL: https://aclanthology.org/2025.nlp4call-1.7/.

[2] Y. Lu, W. Zhu, L. Li, Y. Qiao, F. Yuan, LLaMAX: Scaling linguistic horizons of LLM by enhancing translation capabilities beyond 100 languages, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 10748–10772. URL: https://aclanthology.org/2024.findings-emnlp.631/. doi:10.18653/v1/2024.findings-emnlp.631.

[3] P. Cheng, Y. Dai, T. Hu, H. Xu, Z. Zhang, L. Han, N. Du, X. Li, Self-playing adversarial language game enhances llm reasoning, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, volume 37, Curran Associates, Inc., 2024, pp. 126515–126543.

[4] J. Peng, S. Cheng, E. Diau, Y. Shih, P. Chen, Y. Lin, Y. Chen, A survey of useful LLM evaluation, CoRR abs/2406.00936 (2024). URL: https://doi.org/10.48550/arXiv.2406.00936. doi:10.48550/ARXIV.2406.00936. arXiv:2406.00936.

[5] P. Sahoo, P. Meharia, A. Ghosh, S. Saha, V. Jain, A. Chadha, A comprehensive survey of hallucination in large language, image, video and audio foundation models, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 11709–11724. URL: https://aclanthology.org/2024.findings-emnlp.685/. doi:10.18653/v1/2024.findings-emnlp.685.

[6] I. Mirzadeh, K. Alizadeh, H. Shahrokhi, O. Tuzel, S. Bengio, M. Farajtabar, Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2024. URL: https://arxiv.org/abs/2410.05229.

[7] L. Mo, B. Wang, M. Chen, H. Sun, How trustworthy are open-source LLMs? an assessment under malicious demonstrations shows their vulnerabilities, in: K. Duh, H. Gomez, S. Bethard (Eds.), Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), Association for Computational Linguistics, Mexico City, Mexico, 2024, pp. 2775–2792. URL: https://aclanthology.org/2024.naacl-long.152/. doi:10.18653/v1/2024.naacl-long.152.

[8] Z. Jiang, J. Araki, H. Ding, G. Neubig, How can we know when language models know? on the calibration of language models for question answering, Transactions of the Association for Computational Linguistics 9 (2021) 962–977. URL: https://aclanthology.org/2021.tacl-1.57/. doi:10.1162/tacl_a_00407.

[9] P. Laban, W. Kryscinski, D. Agarwal, A. Fabbri, C. Xiong, S. Joty, C.-S. Wu, SummEdits: Measuring LLM ability at factual reasoning through the lens of summarization, in: H. Bouamor, J. Pino, K. Bali (Eds.), Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore, 2023, pp. 9662–9676. URL: https://aclanthology.org/2023.emnlp-main.600/. doi:10.18653/v1/2023.emnlp-main.600.

[10] B. Wang, X. Yue, H. Sun, Can chatgpt defend its belief in truth? evaluating LLM reasoning via debate, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023, Association for Computational Linguistics, 2023, pp. 11865–11881. URL: https://doi.org/10.18653/v1/2023.findings-emnlp.795. doi:10.18653/V1/2023.FINDINGS-EMNLP.795.

[11] J. Li, R. Li, Q. Liu, Beyond static datasets: A deep interaction approach to LLM evaluation, CoRR abs/2309.04369 (2023). URL: https://doi.org/10.48550/arXiv.2309.04369. doi:10.48550/ARXIV.2309.04369. arXiv:2309.04369.

[12] J. Duan, R. Zhang, J. Diffenderfer, B. Kailkhura, L. Sun, E. Stengel-Eskin, M. Bansal, T. Chen, K. Xu, Gtbench: Uncovering the strategic reasoning capabilities of llms via game-theoretic evaluations, in: A. Globersons, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. M. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Sys-

tems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024.

[13] R. Manna, M. P. di Buono, J. Monti, Riddle me this: Evaluating large language models in solving word-based games, in: C. Madge, J. Chamberlain, K. Fort, U. Kruschwitz, S. Lukin (Eds.), Proceedings of the 10th Workshop on Games and Natural Language Processing @ LREC-COLING 2024, ELRA and ICCL, Torino, Italia, 2024, pp. 97–106. URL: https://aclanthology.org/2024.games-1.11/.

[14] P. Basile, M. Lovetere, J. Monti, A. Pascucci, F. Sangati, L. Siciliani, Ghigliottin-ai@evalita2020: Evaluating artificial players for the language game "la ghigliottina" (short paper), in: V. Basile, D. Croce, M. D. Maro, L. C. Passaro (Eds.), Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020), Online event, December 17th, 2020, volume 2765 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020. URL: https://ceur-ws.org/Vol-2765/paper155.pdf.

[15] P. Samdarshi, M. Mustafa, A. Kulkarni, R. Rothkopf, T. Chakrabarty, S. Muresan, Connecting the dots: Evaluating abstract reasoning capabilities of LLMs using the New York Times connections word game, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 21219–21236. URL: https://aclanthology.org/2024.emnlp-main.1182/. doi:10.18653/v1/2024.emnlp-main.1182.

[16] E. Wallace, N. Tomlin, A. Xu, K. Yang, E. Pathak, M. Ginsberg, D. Klein, Automated crossword solving, in: S. Muresan, P. Nakov, A. Villavicencio (Eds.), Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 3073–3085. URL: https://aclanthology.org/2022.acl-long.219/. doi:10.18653/v1/2022.acl-long.219.

[17] K. Zeinalipour, A. Fusco, A. Zanollo, M. Maggini, M. Gori, Harnessing llms for educational content-driven italian crossword generation, in: F. Dell'Orletta, A. Lenci, S. Montemagni, R. Sprugnoli (Eds.), Proceedings of the Tenth Italian Conference on Computational Linguistics (CLiC-it 2024), Pisa, Italy, December 4-6, 2024, volume 3878 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024. URL: https://ceur-ws.org/Vol-3878/110_main_long.pdf.

[18] O. Topsakal, C. J. Edell, J. B. Harper, Evaluating large language models with grid-based game competitions: An extensible llm benchmark and leaderboard, 2024. URL: https://arxiv.org/abs/2407.07796. arXiv:2407.07796.

[19] D. Wu, H. Shi, Z. Sun, B. Liu, Deciphering digital detectives: Understanding LLM behaviors and capabilities in multi-agent mystery games, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 8225–8291. URL: https://aclanthology.org/2024.findings-acl.490/. doi:10.18653/v1/2024.findings-acl.490.

[20] D. Qiao, C. Wu, Y. Liang, J. Li, N. Duan, Gameeval: Evaluating llms on conversational games, 2023. URL: https://arxiv.org/abs/2308.10032. arXiv:2308.10032.

[21] Y. Wu, X. Tang, T. M. Mitchell, Y. Li, Smartplay : A benchmark for llms as intelligent agents, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net, 2024. URL: https://openreview.net/forum?id=S2oTVrlcp3.

[22] J. Huang, E. J. Li, M. H. Lam, T. Liang, W. Wang, Y. Yuan, W. Jiao, X. Wang, Z. Tu, M. R. Lyu, Competing large language models in multi-agent gaming environments, in: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025, OpenReview.net, 2025. URL: https://openreview.net/forum?id=DI4gW8viB6.

[23] M. Shanahan, K. McDonell, L. Reynolds, Role play with large language models, Nat. 623 (2023) 493–498. URL: https://doi.org/10.1038/s41586-023-06647-8. doi:10.1038/S41586-023-06647-8.

[24] A. Rogers, O. Kovaleva, A. Rumshisky, A primer in BERTology: What we know about how BERT works, Transactions of the Association for Computational Linguistics 8 (2020) 842–866. URL: https://aclanthology.org/2020.tacl-1.54/. doi:10.1162/tacl_a_00349.

[25] F. Petroni, T. Rocktäschel, S. Riedel, P. Lewis, A. Bakhtin, Y. Wu, A. Miller, Language models as knowledge bases?, in: K. Inui, J. Jiang, V. Ng, X. Wan (Eds.), Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 2463–2473. URL: https://aclanthology.org/D19-1250/. doi:10.18653/v1/D19-1250.

[26] M. Wang, Y. Yao, Z. Xu, S. Qiao, S. Deng, P. Wang, X. Chen, J.-C. Gu, Y. Jiang, P. Xie, F. Huang, H. Chen, N. Zhang, Knowledge mechanisms in large language models: A survey and perspective, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen

(Eds.), Findings of the Association for Computational Linguistics: EMNLP 2024, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 7097–7135. URL: https://aclanthology.org/2024.findings-emnlp.416/. doi:10.18653/v1/2024.findings-emnlp.416.

[27] L. Serina, L. Putelli, A. E. Gerevini, I. Serina, Synonyms, antonyms and factual knowledge in BERT heads, Future Internet 15 (2023) 230. URL: https://doi.org/10.3390/fi15070230. doi:10.3390/FI15070230.

[28] L. Putelli, A. E. Gerevini, A. Lavelli, T. Mehmood, I. Serina, On the behaviour of bert's attention for the classification of medical reports, in: C. Musto, R. Guidotti, A. Monreale, G. Semeraro (Eds.), Proceedings of the 3rd Italian Workshop on Explainable Artificial Intelligence co-located with 21th International Conference of the Italian Association for Artificial Intelligence(AIxIA 2022), Udine, Italy, November 28 - December 3, 2022, volume 3277 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 16–30.