Random Numbers Improve Output Diversity in Language Models

Anonymous ACL submission

Abstract

A frequent roadblock in AI research and its real-world applications is that there are only so many potential answers one can get from a single prompt. In this paper we present RESt – a prompting technique yielding diverse outputs from a single prompt without any human intervention. We explore AI's proven divergent thinking capabilities and supplement them with the addition of random numbers, which spark association between different concepts. We show that, just like humans, machines can be creative by drawing inspiration from external stimuli.

1 Introduction

In an age where AI is being integrated into many aspects of life, a lot of effort is spent on refining its ability to provide the correct answer to a prompt. The correct answer can take many forms: a real-world fact, a solution to a problem, an assessment of the user's work, an interesting story. This line of reasoning, where the goal is to converge on a single answer, is called convergent thinking. In contrast stands **divergent thinking** – an ideation process with the intention of generating solutions (Razumnikova, 2013). AI capable of divergent thinking can be a great tool for inspiration that also has practical uses in and of itself, especially in contexts where a large number of different outcomes is the target. Examples include: sentence generation for language learning apps, brainstorm guidance during team meetings, or background dialogues for characters in a video game.

In this paper, we present RESt – a prompting method developed for maximizing the diversity of outputs from a single prompt. It performs better than traditional approaches, with an improvement of up to 3400% in topic diversity.

We review the literature in Section 2. In Section 3, we explain how RESt works and provide an example of a RESt prompt. We describe our corpus and evaluate our method's performance compared to other approaches in Section 4. Afterwards, in Section 5, we discuss our findings and potential future applications of RESt and draw conclusions in Section 6.

2 Literature review

A lot of research has been conducted on the creative capabilities of LLMs, with various interpretations of the term. Many scientists focus on the artistic aspect of creativity, with Franceschelli and Musolesi (2024) providing a comprehensive overview of current research and its implications. They adopt a view that creativity consists of two elements: the actual content, and the reason behind its creation – in other words, the intent.

Others focus only on the output of the LLM and analyze its creative value. Under that approach, creativity is generally measured using two primary metrics: originality and usefulness. Originality (also *novelty*, *uniqueness*) is the subject of the divergent thinking niche of research, where AI has been found to outperform humans in tasks such as the Divergent Association Task (DAT) (Chen and Ding, 2023), the Alternative Uses Test (AUT) (Stevenson et al., 2022) and the Consequences Task (CT) (Hubert et al., 2024). The usefulness (also *value*) element ensures that the ideas conceived by the LLM can be used for practical purposes. Mehrotra et al. (2024) found that increasing AI's originality tends to decrease its usefulness with the exception of storytelling, where increasing originality also increases usefulness. In that study, usefulness is measured by how interested in the story the reader was. It is likely that for this medium in particular, originality and usefulness are therefore inherently linked.

Divergent thinking is a powerful ability of LLMs due to the multitude of its applications. It can be

used for concept generation to aid with creativity (Zhu and Luo, 2022), improving the quality of the output (Liang et al., 2024) or ensuring efficiency of the solution by exploring different lines of reasoning (Yao et al., 2023). Although existing research highlights the advantages of AI's divergent thinking, these capabilities are often memory-dependent. The ideas AI generates are diverse, but limited – eventually, it will return ideas it has come up with before. When evaluating divergent thinking, prompts are often structured in a way to receive multiple answers at once, e.g. "[...] List 10 creative uses for a book" (Stevenson et al., 2022). While this approach works well for single-use scenarios, it does not retain information across sessions, so ideas are going to repeat. In a chat setting, this problem is temporarily solved by memory. However, this memory is also finite, and is not reliable for large-scale operations. One way researchers circumvent this problem is by keeping track of responses that have already been generated and feeding them back into the LLM (Girotra et al., 2023). This strategy works in the short term, but it relies on blocking out certain ideas, rather than inspiring new ones. This can be an issue in highly specialized contexts, where the AI has not had much input on the topic during its training. Such environments are prone to creating hallucinations (Perković et al., 2024). Continuing this line of research, in this paper, we develop a method for generating diverse outputs from the same prompt, even in highly specialized settings. We also measure the quality of the results through expert annotations to ensure that there are no hallucinations.

Our approach mimics a method that people use to come up with creative ideas – *associative thinking*. It is the ability to find connections between various concepts (related and unrelated alike) and draw inspiration from such links. Research suggests that, in practical terms, associative thinking refers to a person's capability of navigating through semantic memory, with more creative individuals capable of making connections across larger semantic distances (Beaty and Kenett, 2023). As DAT studies prove (Chen and Ding, 2023; Cropley, 2023; Hubert et al., 2024), LLMs outperform humans in their ability to make such connections, indicating that associative thinking has the potential to enhance the diversity of their output.

Associative thinking always requires a starting node – the original concept, from which an individual will make connections to others. In Mehrotra et al. (2024), who showed that this strategy results in enhanced creativity from LLMs, using a random object as the starting node yields better results than using the original concept. For example, if the prompt is "Create an original idea for a mug", starting the association process at *ball* leads to a more creative solution than starting at *mug*. Due to AI's inability to produce random output (Liu, 2024), we believe that their method could be improved by providing the LLM with an *external* stimulus.

3 RESt

We propose the Random External Stimulus (RESt) prompting method, usable in zero-shot, one-shot and few-shot environments. A RESt prompt consists of five elements:

An externally generated random number from 0 to 99 (Figure 1).[1] It is important to note here that the random stimulus need not be a number in order to spark associative thinking. For example, it can be a randomly picked word from a predetermined set. However, the benefit of using numbers is that their meaning is more neutral and should not influence the interpretation of the instruction that comes afterwards in the prompt. Its most important feature is that it is generated externally, which ensures true randomness across larger samples. As Liu (2024) has shown, GPT can not output random numbers on its own.

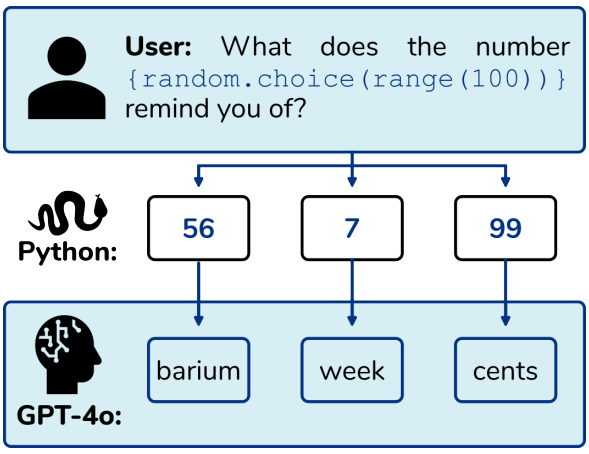

Figure 1: The random number is externally generated, for example by Python, before it is passed on to the LLM.

The instruction (Figure 2). This element is responsible for what the LLM should produce. It

[1]Numbers greater than 100 often result in associations that are related to the properties of the number itself (such as being divisible by 2, being prime, or being a palindrome), resulting in reduced diversity.

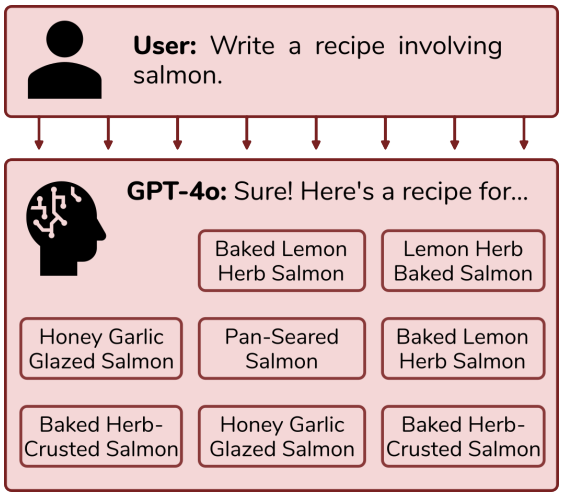

Figure 2: Example instruction. Note that when providing the same instruction to an LLM across separate sessions, the answers repeat.

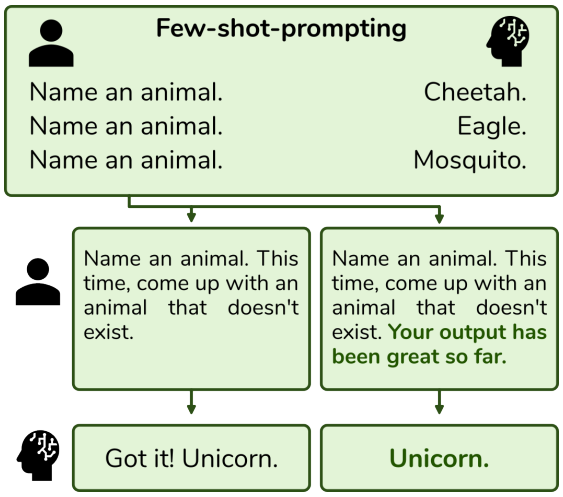

Figure 3: Adding format reinforcement can help prevent the LLM from deviating from the desired format.

should involve a command (such as *write*, *generate*, *come up with*) and the desired output (such as *a homework assignment*, *a fun question*, *a costume idea*). While there are no restrictions on the scope of what the instruction can be, it is best to stick to open-ended requests that do not have correct or incorrect answers.

Reinforcement of the format (Figure 3). It is a message encouraging the LLM to retain the structure it had used in previous responses. While not relevant for zero-shot learning scenarios, its purpose is to prevent deviations when the input changes in the final message. We find that it helps keep the format of the output consistent, but it is not an important element of the prompt and can be omitted if consistency is not a concern. Adherence to structure can also be achieved in other ways, such as a system message at the start.

The association algorithm (Figure 4). It can be any chain-of-thought process consisting of nodes and transformations. Nodes are concepts the LLM uses as intermediate steps. The starting node is the external stimulus and the final node is the desired output. The algorithm should clearly state how the LLM should get from one node to the next, such as: "Say what <node X> reminds you of," an example of association. Steps other than association can be used too, but LLMs have proven to be particularly good at making connections between concepts. A viable alternative could be disassociation: "Name

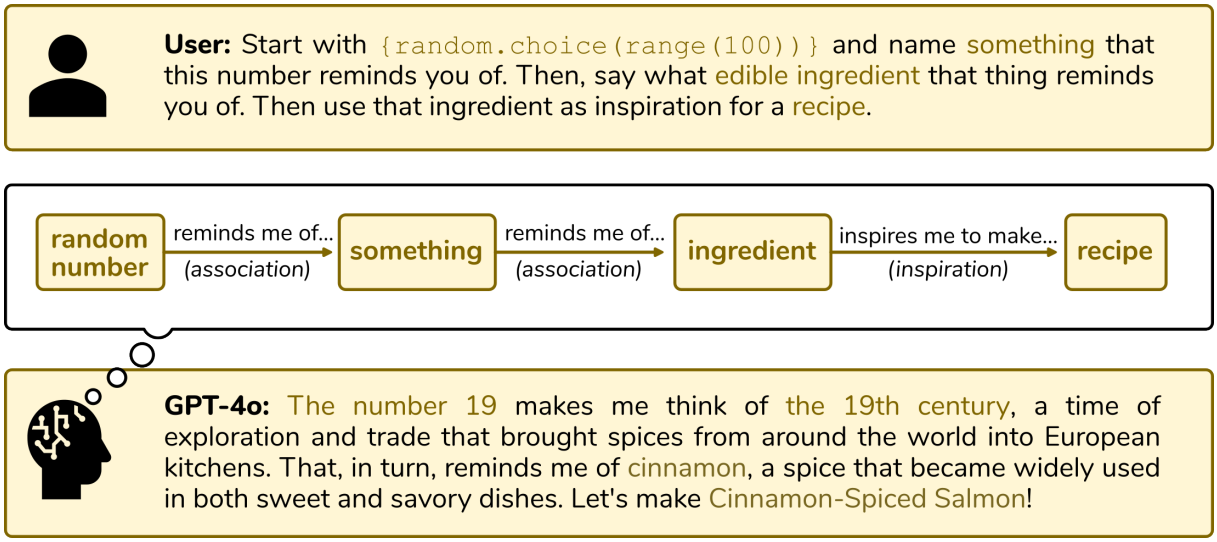

Figure 4: Example association algorithm.

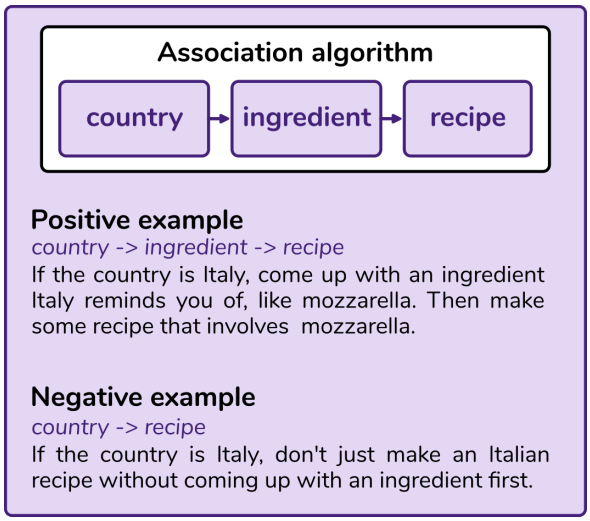

Figure 5: Negative and positive examples. Without them, the LLM is prone to skipping steps in the association algorithm.

the most unrelated thing to <node X> that you can think of."

A negative and a positive example (Figure 5). Their goal is to show the LLM how to properly follow the algorithm. The negative example can be a reminder to ignore previous nodes in the association algorithm. In an algorithm [X → Y → Z], X should not directly influence Z. In other words, a negative example can be a demonstration that the association algorithm is intransitive. It can also take other forms, and if an LLM consistently makes a particular kind of error, it would be worthwhile to include it in a negative example. A positive example should show an acceptable reasoning process.

At its core, RESt is a prompting method for converting one concept into another. However, the inclusion of an external stimulus that can be created without human or LLM input makes it an efficient technique for generating ideas. A full RESt prompt can be found in Figure 6.

Context
Few-shot prompting. User messages are "Write a recipe that involves salmon," LLM messages are recipes in a consistent format.

RESt prompt
`({random.choice(range(100))})` Write a recipe that involves salmon. Your output has been great so far. This time I want you to start with a short line about the number in front of my message and a thing that it reminds you of. Then, I want you to say what edible ingredient that thing reminds you of. And then use that ingredient as inspiration for your recipe. For example, if the number reminds you of Italy, don't write a recipe for pizza with salmon, but instead try to come up with your own dish that contains salmon and mozzarella.

GPT-4o response
The number 19 makes me think of the 19th century, a time of exploration and trade that brought spices from around the world into European kitchens. That, in turn, reminds me of cinnamon, a spice that became widely used in both sweet and savory dishes.
GPT-4o then provides recipe for Cinnamon-Spiced Salmon with Roasted Sweet Potatoes.

Figure 6: An example RESt prompt for writing recipes with salmon. The constituents are, in order of appearance: random number (blue); instruction (red); reinforcement (green); association algorithm (yellow); examples (purple). The prompting takes place in Python.

4 Evaluating RESt

We test our method on OpenAI's GPT-4o model (OpenAI et al., 2024). The primary goal of RESt is to ensure output diversity – how different the responses are from each other – which we measure through automatic analysis. Additionally, we measure output quality through human annotation. The aim of this evaluation is to determine whether the problems generated using RESt prompting perform comparably to those produced by humans and other generation methods. It is important to note that this evaluation does not validate the automatic diversity analysis but rather complements it by assessing the quality of the outputs.

4.1 Corpus

The corpus for this study is a collection of hands-on spontaneous problems from the Odyssey of the Mind (OotM) creativity competition . Each problem is a structured description of a task in which a team of 5-7 children and/or teenagers must use everyday materials to complete a challenge (such as building a structure, or a device for transporting items between two zones). There is always a scoring section, which dictates how many points the

System message

You are an intelligent designer of spontaneous problems for a competition for teams of 5-7 teenagers and children. You can come up with your own problem ideas and don't need to be prompted for the specifics, but you do understand the nature of the competition. In front of every message there is a number, you can use it to boost your creativity with concept ideas. For example, if the number is (7), you can think of something involving the 7 continents. Don't engage in conversation with the user, just write the instruction for the task.

Exemplars

User: "`({random.choice(range(100))})` Design a manual spontaneous problem."
Assistant: *(human-written problem from the corpus)*

Final user message

Do nothing (`temperature = 0.5, top_p = 0.5`):
Design a manual spontaneous problem. Your output has been great so far.

Boost (`temperature = 1, top_p = 0.95`):
Design a manual spontaneous problem. Your output has been great so far.

Encourage (`temperature = 0.5, top_p = 0.5`):
Design a manual spontaneous problem. Your output has been great so far. I want you to come up with a unique problem that you haven't seen before.

RESt (`temperature = 0.5, top_p = 0.5`):
(`{random.choice(range(100))}`) Design a manual spontaneous problem. Your output has been great so far. This time I want you to start with a short line about the number and what it reminds you of. Then, I want you to say what mechanical process that thing reminds you of. And draw inspiration for the task from that. For example, if something reminds you of the moon, don't call things in the instruction "rocket" or "lunar capsule", but instead make it about landing an object.

Figure 7: Prompts used to generate problems. Text colored in purple is used only in RESt.

team can get in each category (such as creativity, height of the structure, or the weight it can bear). An example problem can be found in Appendix A. This dataset was chosen because competition organizers place a lot of importance on not publishing official problems, so it is unlikely that any LLM has been trained on it.[2] The problems come from the Polish branch of the competition, and we have been granted permission to use and publish them through this study. We also machine-translated them into English. This extra precaution should further ensure that GPT-4o has not been trained on the corpus. Most problems were written by the same author, who has overseen the creation of all of them. The role of the corpus in this study is to provide examples for few-shot prompting and to compare GPT-4o's output to human-written text.

Besides human-written problems, RESt is compared against 3 other generation methods. They are:

- **Do nothing** – the prompt only contains the instruction and format reinforcement.
- **Boost** – the prompt contains only the instruction and format reinforcement, but the LLM has enhanced creativity parameters (`temperature = 1` and `top_p = 0.95`).

[2]This is only true for official problems. Some schools and organizations create their own for practice, which can be found on the Internet. However, they rarely follow the same format.

- **Encourage** – the prompt only contains the instruction and format reinforcement, but the instruction explicitly asks for a unique output.

We generate a total of 800 problems across 16 distinct experimental groups, defined by the combination of four different methods (**Do nothing**, **Boost**, **Encourage**, and RESt) and four prompting strategies: zero-shot learning and few-shot learning with 1, 3, and 5 examples. The generation process including the precise prompts used is outlined in Figure 7. Throughout the generation, there was a total of 2,030,612 input tokens and 1,950,126 output tokens.

4.2 Diversity evaluation

For evaluating the diversity, we employ machine classification into clusters through BERTopic. Aside from the 800 generated problems, we add 20 human-written problems, resulting in a total of 820 documents. We use PCA as the dimensionality reduction algorithm with random state equal to 42, and set the minimum cluster size to 2, the lowest possible value. We choose to set the minimum size this low because the primary goal of our method is to produce diverse outputs, and a smaller minimum cluster size allows us to capture this diversity more accurately. Clusters that cannot be further broken down will naturally remain cohesive, while those that can be split into smaller clusters will be, providing a more nuanced understanding of the diversity within our dataset. Once the problems are classified into clusters, we measure the diversity of each experimental group by summing up the total number of clusters in that group, treating every outlier as a separate cluster. Then, that sum is divided by the number of documents in each group, which is the final diversity score, on a scale from 0 to 1. A score of 1 indicates maximal diversity.

4.3 Diversity results

We observe that RESt outperforms every other prompting method at each level of few-shot learning (Figure 8) by a factor of over 100% over the second best performing approach, **Boost**, and up to 3400% over the worst method, **Do nothing**. RESt is also on par with the human-written corpus in terms of diversity, which is a major accomplishment, as the LLM has no memory of what problems it had written before, unlike a human. A noticeable pattern emerges where zero-shot prompting yields less diverse outputs compared to few-shot prompting across all generation methods. However, the specific trend varies for each method, potentially due to the limited sample size. For RESt in particular, the most diverse output is achieved through 3-shot prompting, with a diversity score of 0.72.

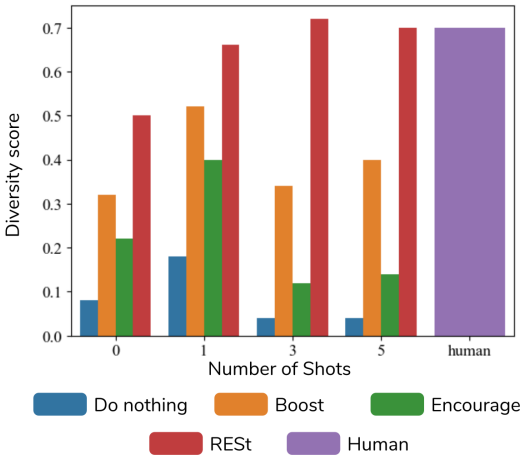

Figure 8: Diversity of the output across 16 generation groups compared to human-written problems.

4.4 Quality evaluation

To measure the quality of generated results, we ask experts to make judgments regarding several aspects of the problems. The experts consist of 3 people who have experience writing, solving and scoring these kinds of problems for the official Odyssey of the Mind competition. They are given 20 problems, which had been picked randomly from a sample of problems generated through 5-shot prompting[3] and the human-written corpus. The distribution is 4 problems from each of the 4 generation methods, plus 4 human-written problems. The experts judge the problems in 8 categories: readability, clarity, logic, practicability, novelty, scoring, acceptability and human element. A brief overview of what each category means is outlined in Table 1. Full questions can be found in Appendix B. The available responses for each category are: “bad”, “not good”, “unsure”, “not bad”, “good”. They are afterwards converted to a scale from 1 to 5, with 1 being “bad” and 5 being “good”. The participants do not know which problems are generated by AI and which are human-written.

[3]This decision was motivated by the fact that 5-shot represents the highest level of few-shot learning used in our experiments. By exposing the model to the largest number of examples, we aim to ensure that the structure of the generated problems is as similar to the human-written problems as possible.

Category	Explanation
Readability	Is the problem grammatically well written and well structured?
Clarity	Is the problem easy to understand?
Logic	Does the problem have no contradictions or missing steps?
Practicability	Is the problem possible to solve given the time and materials?
Novelty	Is the problem unique and/or fun?
Scoring	Does the problem have an appropriate scoring system?
Acceptability	Would you accept the problem at an OotM competition?
Human element	Do you think the problem was written by a human?

Table 1: Categories judged by experts in the annotation task.

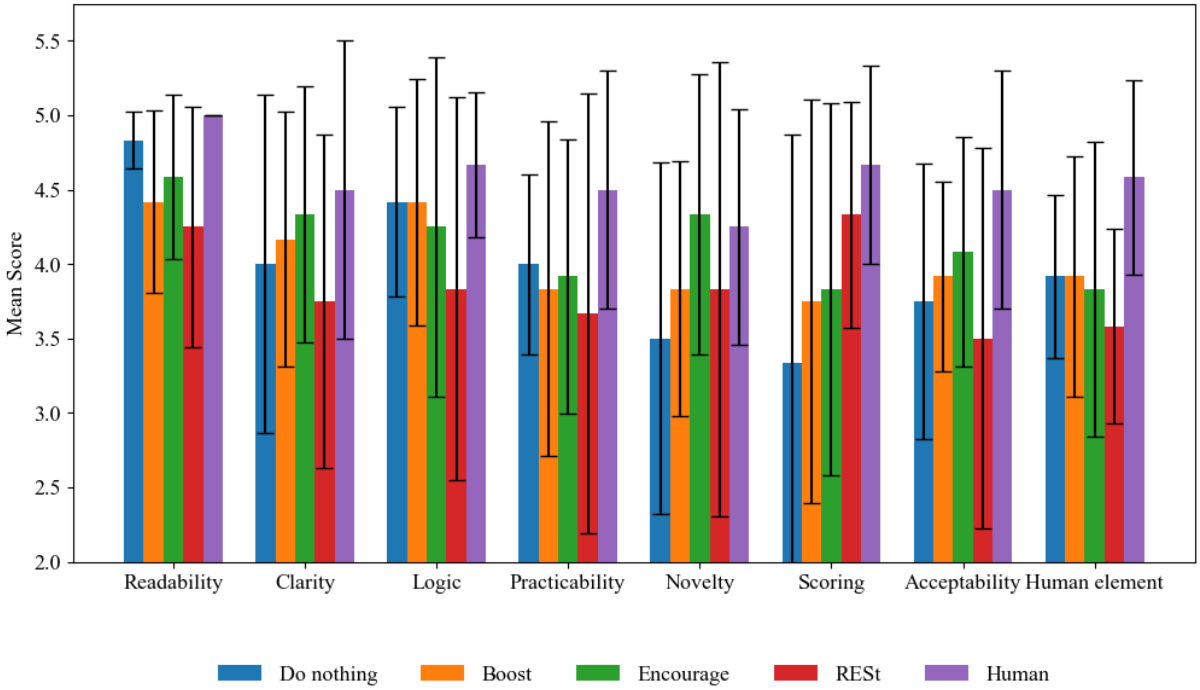

Figure 9: Experts' judgment of the quality of the problems across 4 generation methods compared with human-written problems.

Category	α	CI	AMD
Readability	-0.01	-0.21–0.16	0.95
Clarity	0.21	-0.13–0.49	1.40
Logic	0.15	-0.13–0.40	1.30
Practicability	0.24	-0.09–0.54	1.35
Novelty	0.36	0.04–0.60	1.35
Scoring	0.31	-0.05–0.59	1.35
Acceptability	-0.03	-0.26–0.22	2.00
Human element	-0.27	-0.38– -0.14	2.30

Table 2: Inter-Annotator Agreement through Krippendorff's Alpha (α), its 95% confidence intervals (CI), and average maximum disagreement (AMD) for each category.

4.5 Quality results

Before analyzing the results, we calculate Inter-Annotator Agreement (IAA) between the 3 experts (Table 2). We calculate it using Krippendorff's Alpha using the Krippendorff Python package (Castro, 2017) with the level of measurement set to ordinal, together with 95% confidence intervals, as recommended by van der Lee et al. (2019). Additionally, we calculate average maximum disagreement (AMD) for each category to better illustrate what the disparities are. The expert survey shows moderate agreement in certain categories, while none in others. The categories the experts agreed on the most are novelty, scoring, practicability and clarity, with the best alpha value being 0.36. On the surface, this constitutes poor IAA, but it is im-

portant to note that the task was highly subjective. It is expected that evaluations with few annotators have lower IAA scores, as well as larger confidence intervals given the variable nature of human language (van der Lee et al., 2019; Amidei et al., 2018). However, it is worth noting that in terms of AMD, experts seem to agree on readability the most, with an average AMD of 0.95. The AMD scores seem overall acceptable, with the exception of two categories: acceptability and human element, where disagreement is high under both metrics.

Despite a lack of clear agreement, there is still insight to be gained from the quality survey, visualized in Figure 9. Interestingly, human-written problems are unanimously rated at 5 out of 5 points by all experts. While the sizable error bars make it difficult to state clear patterns, human-written problems seem to have a lead over AI-generated problems in all categories but one – novelty, where Encourage performs the best. This category sees the most agreement among the experts, but contains the longest margins of error, particularly for the RESt method. This indicates that problems in that group are highly different from each other, most of them being either an exceptional success or a notable underperformance. However, the general consensus seems to be that, across most categories, RESt performs slightly worse than other generation methods in terms of the quality of its output.

5 Discussion

5.1 Practical applications

RESt could be a crucial technique for any environment where the goal is to generate pieces of text that are different from each other. In education, it can be used to create large sets of exercises or homeworks for students to practice, such as in Jordan et al. (2024). In research, it enables scientists to gather a rich and varied dataset with minimal intervention, without having to resort to methods such as feedback loops or asking for multiple answers in a single prompt to prevent repetitions. In the industry, there is a multitude of use cases for diverse generation. These range from physical products like collection cards, to digital assets like unique customization options on a website, to tools used by project teams to brainstorm ideas and guide their discussion. There is also potential in using RESt for inspiration, both by real artists and LLMs themselves. As Mehrotra et al. (2024) show, AI-written stories are much more immersive when associative thinking strategies are employed. Enhancing those strategies with true randomness could lead to even better results.

5.2 RESt as a supplement

In this paper we proved that RESt is capable of generating a number of different concepts without any intervention, but not all of those concepts will meet the expectations of the user. It is therefore worth exploring this approach not only as a standalone method, but also as a complement to others. Some examples include Tree-of-Thoughts (Yao et al., 2023), Multi-Agent Debate (Liang et al., 2024), or other prompting strategies where the aim is to improve the quality of the output (Sahoo et al., 2024).

6 Conclusions

While simple at its core, RESt is a powerful prompting technique for obtaining diverse outputs in mass quantities, with a slight decrease in quality. Further research should focus on establishing whether that quality drop is statistically significant. Additionally, other experiments with RESt should be performed – specifically, single concept generation, where the LLM is asked to produce one word. This approach seems easier to reliably evaluate in terms of diversity, because text understanding is no longer necessary. Lastly, increasing creativity parameters elevated the Boost method above Do nothing. It would be useful to test if adjusting those values could similarly lead to RESt producing better and more diverse results.

Limitations

We are confident that RESt is exceptionally promising. However, due to the complex nature of the metrics involved, it present challenges in objective measurement, which are discussed further in Appendix C. Furthermore, as the expert annotation shows, the quality of the output is not as high as standard prompting methods like **Do nothing**.

Ethical considerations

The data for the quality of the outputs was obtained through a survey, with answers from 3 experts who have experience writing, solving and scoring such problems for Odyssey of the Mind Polska. Each expert was paid 100 PLN (around 25 USD) for the task.

References

Jacopo Amidei, Paul Piwek, and Alistair Willis. 2018. Rethinking the agreement in human evaluation tasks. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3318–3329, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Roger E. Beaty and Yoed N. Kenett. 2023. Associative thinking at the core of creativity. *Trends in Cognitive Sciences*, 27(7):671–683.

Santiago Castro. 2017. Fast Krippendorff: Fast computation of Krippendorff's alpha agreement measure.

Honghua Chen and Nai Ding. 2023. Probing the "Creativity" of Large Language Models: Can models produce divergent semantic association? In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12881–12888, Singapore. Association for Computational Linguistics.

David Cropley. 2023. Is artificial intelligence more creative than humans? : ChatGPT and the Divergent Association Task. *Learning Letters*, 2:13–13.

Giorgio Franceschelli and Mirco Musolesi. 2024. On the Creativity of Large Language Models. *Preprint*, arXiv:2304.00008.

Karan Girotra, Lennart Meincke, Christian Terwiesch, and Karl T. Ulrich. 2023. Using Large Language Models for Idea Generation in Innovation. *SSRN Electronic Journal*.

Kent F. Hubert, Kim N. Awa, and Darya L. Zabelina. 2024. The current state of artificial intelligence generative language models is more creative than humans on divergent thinking tasks. *Scientific Reports*, 14(1):3440.

Mollie Jordan, Kevin Ly, and Adalbert Gerald Soosai Raj. 2024. Need a Programming Exercise Generated in Your Native Language? ChatGPT's Got Your Back: Automatic Generation of Non-English Programming Exercises Using OpenAI GPT-3.5. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, pages 618–624, Portland OR USA. ACM.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. Encouraging Divergent Thinking in Large Language Models through Multi-Agent Debate. *Preprint*, arXiv:2305.19118.

Qiang Liu. 2024. Does GPT-4 Play Dice?

Pronita Mehrotra, Aishni Parab, and Sumit Gulwani. 2024. Enhancing Creativity in Large Language Models through Associative Thinking Strategies. *Preprint*, arXiv:2405.06715.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, A. J. Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button,

Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. 2024. GPT-4o System Card. *Preprint*, arXiv:2410.21276.

Gabrijela Perković, Antun Drobnjak, and Ivica Botički. 2024. Hallucinations in LLMs: Understanding and Addressing Challenges. In *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, pages 2084–2088, Opatija, Croatia. IEEE.

Olga M. Razumnikova. 2013. Divergent Versus Convergent Thinking. In Elias G. Carayannis, editor, *Encyclopedia of Creativity, Invention, Innovation and Entrepreneurship*, pages 546–552. Springer New York, New York, NY.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. *Preprint*, arXiv:2402.07927.

Claire Stevenson, Iris Smal, Matthijs Baas, Raoul Grasman, and Han van der Maas. 2022. Putting GPT-3's Creativity to the (Alternative Uses) Test. *Preprint*, arXiv:2206.08932.

Chris van der Lee, Albert Gatt, Emiel Van Miltenburg, Sander Wubben, and Emiel Krahmer. 2019. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 355–368, Tokyo, Japan. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. *Preprint*, arXiv:2305.10601.

Qihao Zhu and Jianxi Luo. 2022. Generative Transformers for Design Concept Generation. *Preprint*, arXiv:2211.03468.

A Example problems

Figure 10 contains one of the 20 problems from the human-written corpus. They all follow a similar structure – first, they contain meta information about the problem, including its title and procedures about asking judges for the time. Then, they describe the setting of the competition. Afterwards they explain the core of the problem, and then proceed to specific rules. The final section of the instruction is the scoring, where the team learns how many points they can receive for each aspect of the problem. The list of materials is not present in real-world problems (because the materials are visible to the participants), but we chose to include them to have annotators evaluate how well suited the provided materials match are to the task in the practicability category. None of the problems contain any personal information. An example 5-shot RESt output can be found in Figure 11. The reasoning part was cut out from the diversity and

quality evaluations to prevent from easily distinguishing them from the rest, but we include it here for demonstration of the algorithm.

B Expert survey

The survey was conducted through Google Forms. Participants were sequentially presented with a problem, followed by an explanation of the evaluation categories they were asked to judge, and then a grid where they recorded their judgments. The exact questions the experts were asked can be found in Table 3. The problems were presented to each participant in the same order. All 3 experts are native speakers of Polish and fluent speakers of English.

C BERTopic clustering

We decided that the best minimum cluster size parameter for evaluating diversity would be 2 – it is the smallest possible cluster size, which provides us with a granular approach. That way, we separate clusters that can be separated, but nearly identical outputs stay classified together. However, interesting trends can be observed when changing that parameter to higher values (Figure 12). At 5, the **Boost** method outperforms RESt in 1-shot prompting, while at 10, all four methods perform similarly (except for **Boost** at 1-shot).

It is likely that at higher minimum cluster size values, BERTopic merges problems written in a similar style together, which can be evidenced by a sudden drop of human diversity between 4 and 5. At a minimum cluster size of 10, human-written problems have a diversity score of 0.05. Given that there is 20 of them, it means that they all become merged into a single cluster, despite being distinct from a human's perspective. We hypothesize that the same happens to other methods, and **Boost**'s relatively strong performance could be attributed to its high temperature and top_p parameters, which cause it to use a more varied vocabulary, thus resulting in perceived diversity. In our view, that makes it all the more impressive that RESt outperforms **Boost** at lower minimum cluster sizes.

Solve the hands-on problem titled "Lightning Structure."
All team members can participate in working on the task. The problem statement will first be read to you in its entirety, and then its key parts will be repeated. You will have two written copies of the instructions at your disposal, which you can use whenever you wish. Once the clock starts, you can also ask at any time how much time you have left. Good luck!
Instructions:
1. There is a square marked with tape on the floor, along with materials you can use. You are not allowed to use anything else. Scissors can be used for work but cannot be incorporated into the solution.
2. Your task is to construct one solid structure from the available materials, which should be as tall as possible. If a golf ball is placed on top of it, you will earn significantly more points for the height of the structure.
3. The structure can only touch the floor—and only within the designated square. It cannot be supported by anything or anyone.
4. You will have only 5 minutes to solve the problem. You will be informed when there are 2 minutes left, and then 1 minute left. You can talk to each other and ask the judges questions whenever you want—but the time measurement will not be paused for this.
5. The task will end when the time limit is reached or when you request an evaluation.
6. You will be scored as follows:
a. For every full 5 centimeters of the structure's height, you will receive 2 points.
b. If a golf ball is supported on top of the structure, the score for its height will be multiplied by 3.
c. You will also earn between 1 and 10 points for the creativity of your solution...
d. ...and between 1 and 15 points for teamwork.
List of available materials:
- 15 wooden toothpicks
- 7 drinking straws
- 7 cotton swabs
- 7 flexible, unbreakable paper clips
- 7 pipe cleaners
- 3 playing cards
- 3 small adhesive labels
- 2 rubber bands
- 2 sheets of A5 paper
- 1 paper cup
- 1 R20 battery
- 1 pair of scissors (can be used for work but cannot be incorporated into the solution)

Figure 10: Example problem from the human-written corpus.

The number 27 reminds me of the 27 letters in the Spanish alphabet. This, in turn, makes me think of the mechanical process of a printing press, where letters are arranged to form words and sentences. Drawing inspiration from this, let's create a task that involves arranging elements to convey a message.
Solve the manual problem titled "Message in Motion."
All team members can participate in working on the task. The problem's content will first be read to you in its entirety, and then its key parts will be repeated. You will have two written copies of the instructions at your disposal. You can refer to them whenever you wish. Once the clock starts, you can also ask at any time how much time you have left. Good luck!
Instructions:
1. You will have 6 minutes to complete the task. You will be informed when there are 2 minutes and then 1 minute remaining. You can talk among yourselves and ask the judges questions whenever you want, but the time measurement will not be paused for this.
2. On the table, there are various materials and a set of symbols. Your task is to arrange these symbols to form a coherent message that can be "read" by moving a pointer across them.
3. The symbols must be placed on a track that you will create using the available materials. The track should guide the pointer smoothly from the start to the end of the message.
4. You may not touch the symbols directly with your hands once they are placed on the track. You must use the pointer to adjust their positions if needed.
5. The task will end when the time limit is reached or when you request an evaluation.
6. You will be scored as follows:
a. You will receive 1 point for each symbol correctly placed in the sequence.
b. You will earn 5 bonus points if the entire message can be read smoothly without stopping the pointer.
c. You will also receive between 1 and 10 points for the creativity of your track design...
d. ...and between 1 and 10 points for teamwork.
List of available materials:
- 15 small cardboard squares (for symbols)
- 10 popsicle sticks
- 5 pipe cleaners
- 5 rubber bands
- 3 sheets of colored paper
- 3 adhesive labels
- 2 paper cups
- 1 marble (to be used as the pointer)
- 1 pair of scissors (can be used for work but not incorporated into the solution)

Figure 11: Example 5-shot RESt output. Purple text was cut out before evaluation.

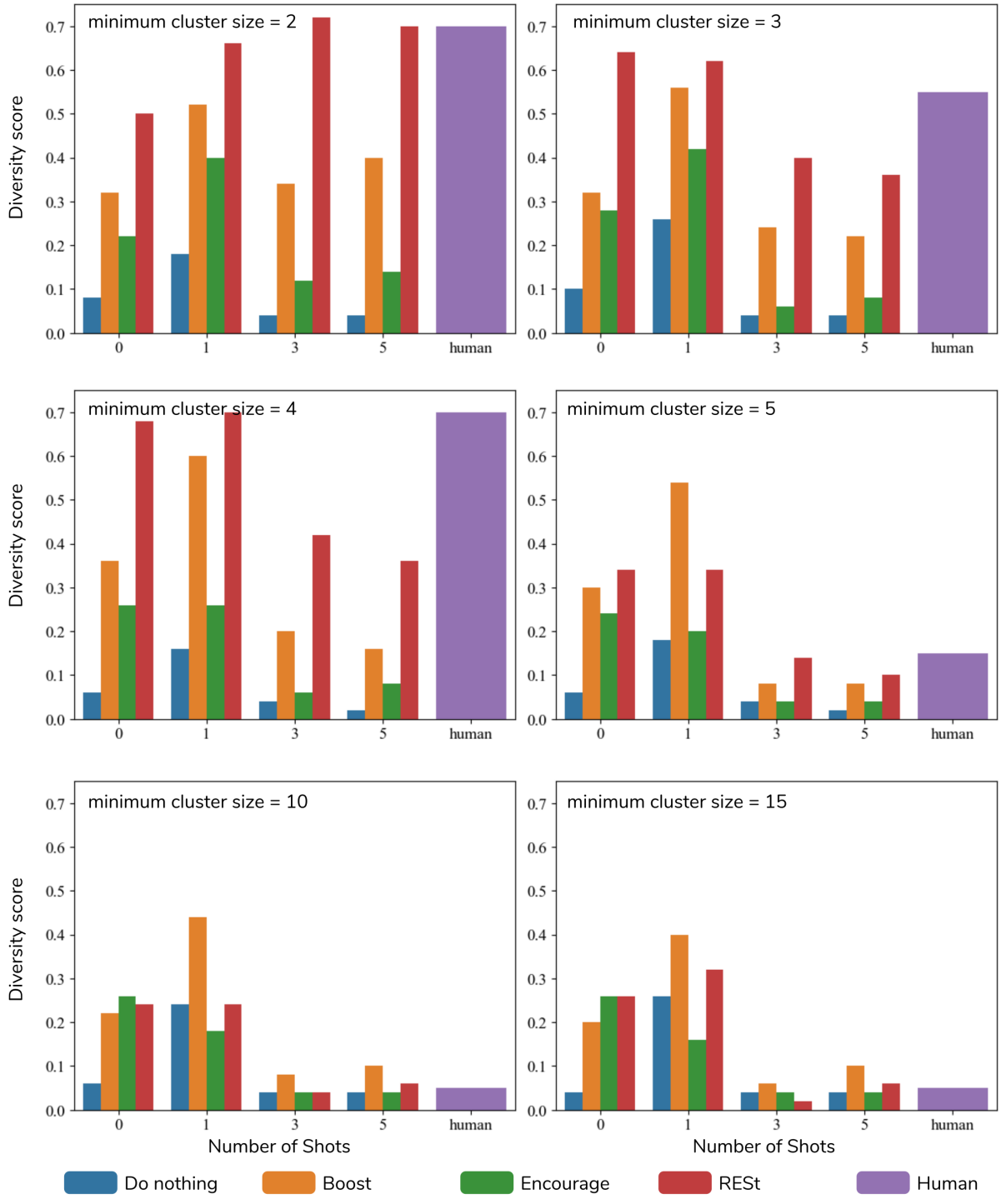

Figure 12: Diversity scores depending on minimum cluster size. PCA random_state parameter is 42.

Category	Question
Readability	Is the instruction understandable to you from the grammatical point of view? Does it contain typos? Do you find the wording strange or unnatural?
Clarity	Do you understand what the task is? Is the information in the instruction presented in an appropriate order?
Logic	Does everything make sense? Are there contradictions? Is there missing information? Is the instruction consistent?
Practicability	Is the task physically possible? Is it reasonable for a human or team of teenagers to do? Are the time limit and materials adequate?
Novelty	Is the problem unique? Would it be fun to solve? Have you seen a problem like this before?
Scoring	In the evaluation section, does the number of points awarded for each category make sense? Do you find that there should be a category that's missing? Is there a scored category that shouldn't be scored?
Acceptability	Would you accept this task at the Odyssey of the Mind competition? Does it follow the structure of a typical OotM problem? Is it too easy?
Human element	Does the task feel as if it was written by a human? (good if human, bad if AI)

Table 3: Categories judged by experts in the annotation.