LLM Shots: Best Fired at `System` or `User` Prompts?

Umut Halil
Huawei Research Ltd.
Edinburgh, United Kingdom
umut.halil@h-partners.com

Jin Huang
Huawei Research Ltd.
Edinburgh, United Kingdom
jin.huang2@h-partners.com

Damien Graux
Huawei Research Ltd.
Edinburgh, United Kingdom
damien.graux@huawei.com

Jeff Z. Pan
University of Edinburgh
Edinburgh, United Kingdom
j.z.pan@ed.ac.uk

Abstract

Over the past few years, the range of use cases involving Large Language Models (LLMs) has grown dramatically. Alongside this growth, many techniques have been established by the community to boost LLM performance. Among them, relying on the fact that LLMs excel at reproducing behaviors, practitioners have been charging their prompts with examples (*a.k.a.* shots) to guide or orientate the LLMs towards the correct directions given their main instruction. More recently, LLMs have evolved, allowing users to define overall roles by offering two inputs: `system` and `user` messages, based on the assumption that -in a sense- `system` instructions would be dedicated to admin/designer of chatbot interfaces. In such a setting: what is the best place to give example to the LLM so to improve its performances? In this study, we address this research question by systematically trying different *shooting* combinations with different popular benchmarks across a large set of LLMs. Our experiments show that it tends to be more beneficial to guide the LLMs through their `system` prompting mechanisms, leaving only the questions into their `user` messages.

CCS Concepts

• **Computing methodologies** → **Information extraction**; **Language resources**.

Keywords

Large Language Models (LLMs), Prompt engineering, Example-based prompting, System messages, User messages, Benchmarking, Shooting strategies, Performance optimization, Context-aware interactions

ACM Reference Format:
Umut Halil, Jin Huang, Damien Graux, and Jeff Z. Pan. 2025. LLM Shots: Best Fired at `System` or `User` Prompts?. In *Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3701716.3717814

WWW Companion '25, Sydney, NSW, Australia

ACM ISBN 979-8-4007-1331-6/2025/04
https://doi.org/10.1145/3701716.3717814

Tl;dr. Place your question in the `user` prompt and everything else in `system` prompt, including 2 or 4 *(0 for GPTs)* shots. 🍹

1 Introduction

In recent years, the development and application of Large Language Models (LLMs) have witnessed a substantial rise, marking a significant shift in the landscape of artificial intelligence and natural language processing. These models, built upon vast datasets and complex neural architectures, have shown remarkable capabilities in understanding and generating human-like text, thus finding applicability in a myriad of domains ranging from content creation and customer service to programming assistance and beyond. This surge in their utility is complemented by ongoing research into enhancing their performance, where the academic and industrial communities are heavily invested in devising refined techniques to leverage these models' potential to the fullest.

Among the strategies employed, one of the most prevalent involves augmenting the model's prompts with examples, often referred to as "shots", which serve to steer the LLMs toward more accurate outputs in response to given tasks. This approach capitalizes on the LLMs' proficiency in mimicking patterns and behaviors demonstrated within these examples, thereby fine-tuning their output to align with user expectations and the specific nuances of various tasks.

Furthermore, as LLMs continue to evolve, there has been a significant pivot towards more sophisticated interaction paradigms, such as the introduction of distinct input roles: `system` and `user` messages. This dual-input framework hypothesizes that `system` messages can serve a meta-level function, allowing developers and administrators to encode overarching directives and roles, which, in turn, could shape the broader conversation dynamics initiated by the `user`. By delineating these roles, practitioners can potentially craft more coherent and context-aware interactions between the LLMs and end-users.

Amid these developments, a critical question emerges: How should examples—or "shots"—be strategically positioned within this `system`|`user` message framework to maximize LLM performance? Our study addresses this very question by conducting a comprehensive analysis that explores various "shooting" strategies across a spectrum of popular benchmarks and a wide array of LLMs. Through methodical experimentation and evaluation, we reveal the

efficacy of different prompting configurations, ultimately demonstrating that embedding examples within the `system` prompts generally yields superior performance outcomes. This insight highlights the potential for optimizing LLM guidance through well-considered prompt structuring, thus offering a valuable perspective for both AI practitioners and researchers looking to enhance the capabilities of these models. To the best of our knowledge, this is the first time that positioning shots in a `sys|user` prompting environment is the focus of a study.

The rest of this article is structured as follows. We will first describe the specificity of the couple `sys|user` through the lens of in-context learning (Section 2), before diving into the experimental details (Section 3). Finally, before concluding, we will review some related efforts (Section 4).

2 The *shooting* approach in `sys|user` setting

Context of in-context learning. Large Language Models have demonstrated good capabilities when it comes to returning or answering NL questions or reformatting instructions. Nevertheless, they can "misunderstand" and hallucinate, sometimes because their reasoning space was not aligned with the users' expectation. To mitigate this drawback, one approach lays in guiding them with examples. As highlighted in [2, 4, 12], LLM performances often[1] increase when examples are provided, either to show LLM the expected return structure or to provide it an indication of the expected domain of knowledge it should utilise [9, 22, 24]. Practically, practitioners tell the LLMs the instruction together with few examples before asking for their queries, letting then the model produce the next tokens accordingly. For instance, in the following example, three *examples* of "sentiment analysis" are given with the *instruction* to drive the LLM within the space of expected result formats and labels.

Few-shot prompting (1 instruction + 3 shots + the query)

Associate sentiment for the given sentence. For instance:
"The movie was good" > positive
"The movie was quite bad" > negative
"I like the movie, but the ending was lacking" > neutral
"I LOVED the movie" > { . . . LLMs to generate tokens. . . }

More generally, depending on the considered LLM and the task/-dataset involved, the "best" number and selection of shots will vary [6] [2]. For instance, with the "sentiment analysis" task depicted in the above box, in order to get the best results it would be best to have examples with varying sentiments, in order to tell the LLM the space of expected results. Having a distribution of examples biased towards irrelevance when the number of provided examples is low may hurt performance by introducing semantic contamination [19] to the output generation. Finally, adding variability to this setting, the *instruction*, which corresponds to the overall task description, can also be tuned[3].

Creation of `SYS|USER`*-prompt ecosystems.* Originally large language models were queried (*a.k.a.* prompted [14]) through natural language instructions, relying solely on their capability of next-token-prediction to come and build up their answer. However following InstructGPT [16], their querying interfaces / modalities have evolved and major actors now propose more complex paradigm where roles or *personæ* may be instructed. In practice, pairs of `system` (*a.k.a.* `developer`) and `user` prompts can be passed to an LLM where the former should be used to specify specific behaviors or roles that the LLM must consider while answering the latter prompt. Originally, the `system` has been thought so that chatbot-designers could give specific rules while end-users would only have access to the `user` one for interacting with the model. Such an approach presents the advantage of having one single LLM which could then be tuned for plethora of use-cases, then in practical usage an end user can attempt to guide the LLM for a more use-case-specific response through prompting techniques like in-context learning (discussed later in Section 4). Considering the previous example, a potential `system` prompt could be added like so:

SYS|USER setting (a *general* role + a *conventional* prompt)

You are a movie expert and critique writer.

Associate sentiment for the given sentence. For instance:
"The movie was good" > positive
"The movie was quite bad" > negative
"I like the movie, but the ending was lacking" > neutral
"I LOVED the movie" > { . . . LLMs to generate tokens. . . }

Such an architecture has been made possible thanks to LLM post-training methods. If the technical (implementation) details for commercial models remain unclear[4], some openly available LLMs training steps have been described. For instance, `system` messages (for Multi-Turn Consistency) in Llama 2 [21] have been set up *via* the Ghost Attention method, which tweaks their post-training fine-tuning data in order to help the attention focus in a multi-stage process. Similarly, Mistral 7B [10] introduced a `system` prompt to guide the model to generate answers within specified guardrails, using the same method as for Llama 2.

Efficient prompt engineering. Based on the aforementioned example, assuming no specific *persona* would be given as `system` prompt[5], one may wonder where to position which parts, *e.g.* the instruction and the examples in the `system` and the final query in the `user`. More generally, considering a few-shot setting as above, we would like to find the sweet-spot –if any– in the sense that examples could be put at both places. Indeed, these prompts could be seen as blocks where the position of the limit between the two types of inputs could move:

$$\left[instruction \ \ldots \ shot_i \ \overset{\texttt{sys}\leftarrow}{\|}\overset{\rightarrow\texttt{user}}{} \ shot_{i+1} \ \ldots \ query \right]$$

[1] Some tasks do not benefit from such strategy, see *e.g.* [4, 19] for examples.
[2] Such a topic is not the focus of the current study, see *e.g.* [5] for details.
[3] The instruction prompt-engineering is out of this study's scope.

[4] Maybe OpenAI and its GPT LLM suit implements `system` prompts by concatenating them everytime before each `user` prompt.
[5] Apart, maybe, from the famous "You're a useful assistant" popularized by OpenAI.

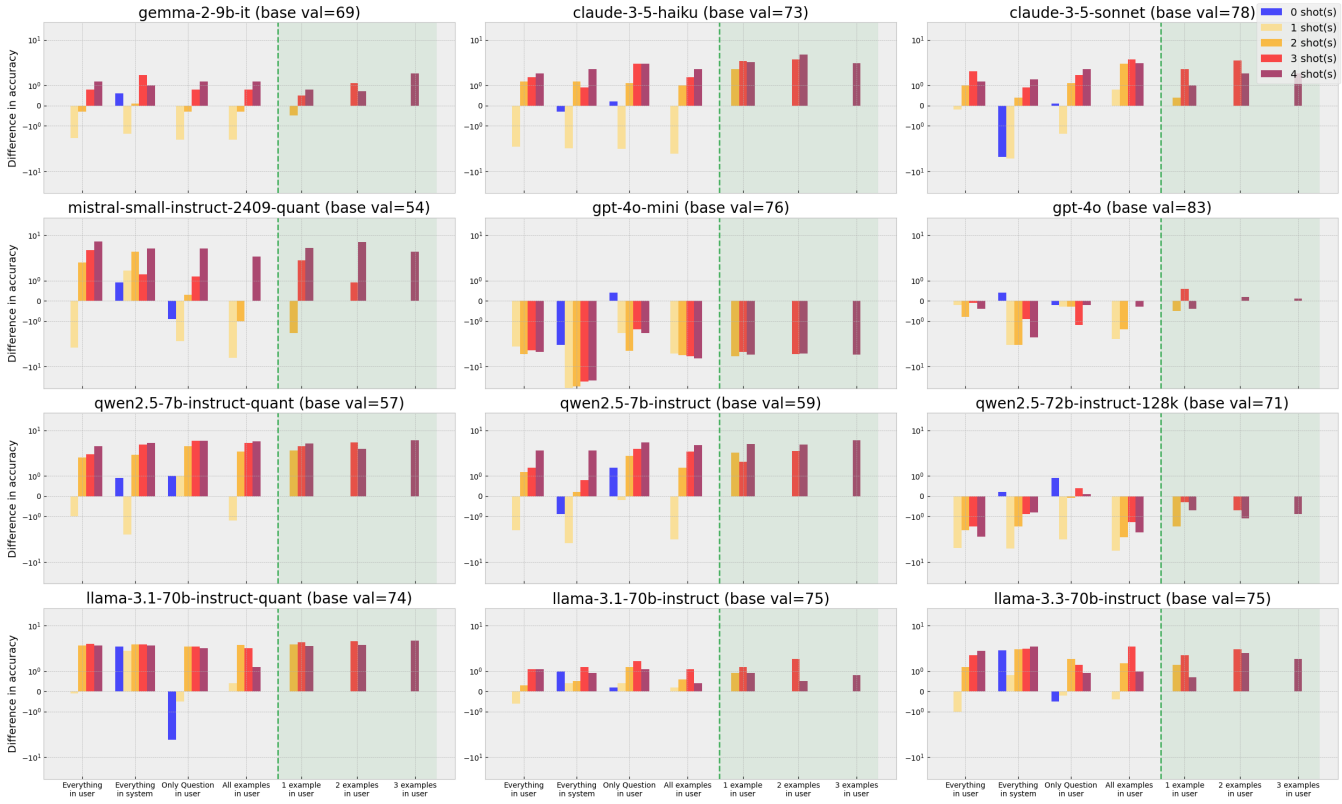

Figure 1: BoolQ [7].

As the respective architectures and interactions of `system` and `user` are not publicly advertised by LLM designers, the safest approach we could set up corresponds to a benchmark-driven method where LLMs answers on different configurations would be monitored, as presented in the following Section.

3 Experiments: Towards the best *shot* recipe

In this Section, we present the results of our evaluations of **25** distinct limit $^{\texttt{sys}\leftarrow}_{\texttt{user}\rightarrow}$ configurations on 4 datasets across 12 large language models.

Protocol. We consider scenarios ranging from 0- to 4-shot while keeping *instruction*, *shot* and *query* in this order to maintain the semantic of the overall prompt which we would have in non-`sys|user` prompting environments. For each dataset, we selected up to four examples, which were used consistently across all LLMs. This approach ensured that any observed performance differences were attributable to the inherent capabilities of the models, rather than variations in the input examples. Each benchmark was evaluated using 1,000 queries. The decision to limit the benchmarks to 1,000 queries and up to 4-shot examples was primarily driven by computational and time constraints. However, we believe this is still sufficient to highlight the differences between shot configurations.

To the best of our knowledge (see Section 4 for more details), this study is the first to systematically explore the best shot-configurations across a wide range of popular and up-to-date LLMs.

3.1 Benchmarks and LLMs

Datasets. To perform our review, we selected four datasets based on their popularity and on the fact that they have been part of in-context learning state-of-the-art studies where authors were assessing the capabilities of LLMs to perform few-shot prompting, (usually putting all the examples at the same place).

- **BoolQ** [7]: A dataset of naturally occurring yes/no questions, demonstrating that they often involve complex entailment reasoning, and finds that transferring knowledge from entailment data is particularly effective. *[Ex: Are house tax and property tax the same?]*
- **MMLU-Pro** [23]: An enhanced benchmark for language models featuring more challenging reasoning-focused questions and expanded choice options, which reduces model accuracy significantly compared to MMLU, while demonstrating better stability under prompt variations and highlighting the effectiveness of Chain of Thought reasoning for complex questions. *[Ex: Let A be the set of all ordered pairs of integers (m, n) such that $7m + 12n = 22$. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$?]*
- **PIQA** [3]: A dataset designed to benchmark physical commonsense reasoning by presenting questions that challenge AI models to choose between options involving everyday physical scenarios, revealing that while the task is straightforward for humans (95% accuracy), it remains difficult for pre-trained models. *[Ex:*

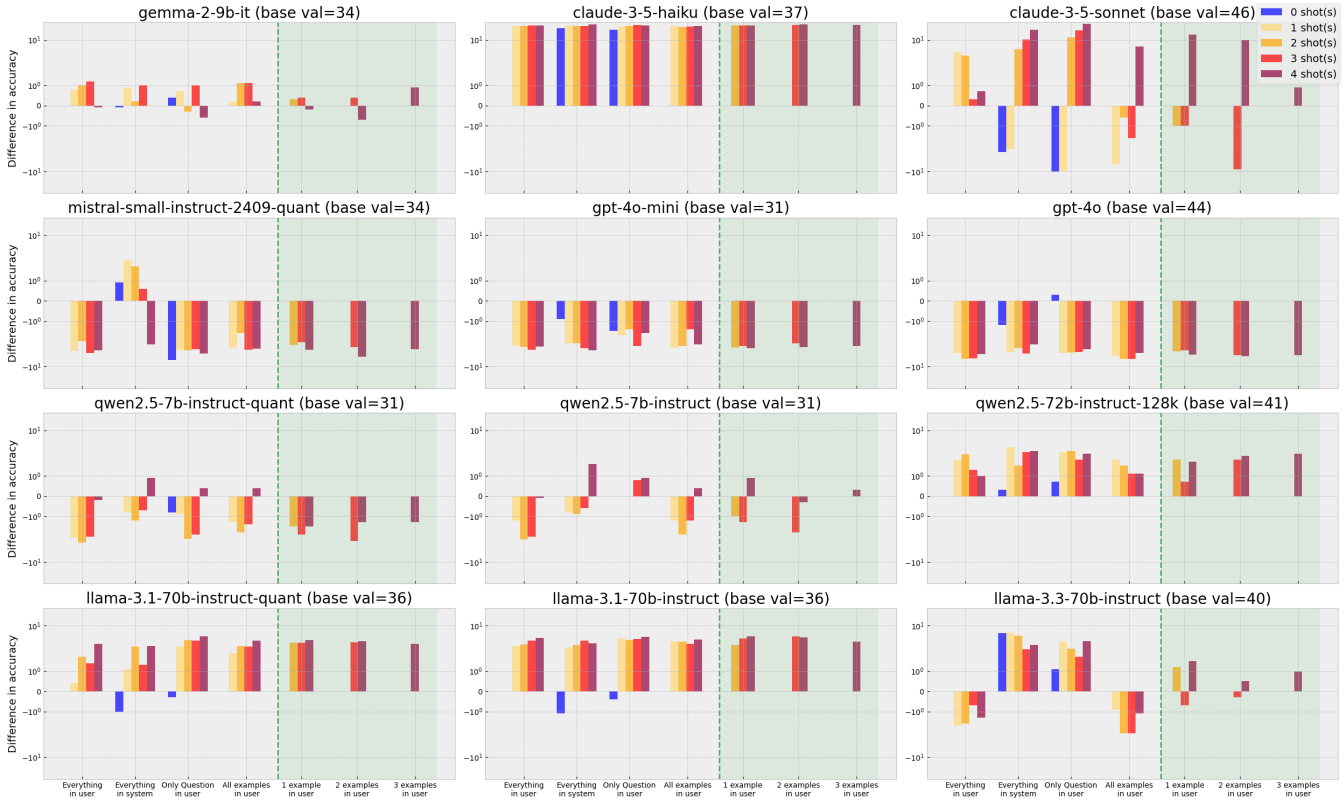

Figure 2: MMLU-Pro [23].

To apply eyeshadow without a brush, should I use a cotton swab or a toothpick?]

- **SQuAD-v2** [18]: Based on the first iteration of SQuAD [17][6], the v2 introduces a novel challenge by incorporating over 50 000 adversarially written unanswerable questions into the existing dataset, requiring systems not only to answer questions when possible but also to recognize and abstain from answering when no answer is supported by the context. *[Ex: After cellulose, what component is most plentiful in wood? Paragraph: Aside from water, wood has three main components…]*

Considered LLMs. To review the preferences of language models when it comes to positioning examples for in-context learning, we utilised LLMs from several leading organisations, ensuring that both general-purpose and specialist models (*i.e.* chatting, or instruction-following modes) are considered. Our set of models includes LLMs from **OpenAI** (*gpt-4o-mini* and *gpt-4o*), **Anthropic** (*claude-3-5-haiku* and *claude-3-5-sonnet*), **Google** (*gemma-2-9b-it*), **Meta** (*llama-3.1-70b-instruct-quant*, *llama-3.1-70b-instruct*, and *llama-3.3-70b-instruct*), **Mistral** (*mistral-small-instruct-2409-quant*), and **Alibaba** (*qwen2.5-7b-instruct-quant*, *qwen2.5-7b-instruct* and *qwen2.5-72b-instruct-128k*). Overall, this set involves members of **6** distinct providers, including commercial and open LLMs. In addition, this set allows us to compare behaviors and performances across different parameter numbers and specialities.

3.2 Results

On Figures 1, 2, 3, and 4, we represent the relative accuracy/F1 variations for the four datasets of the different configurations around the *base case*. As we want to evaluate the performance differences when varying the number of examples and their locations within the pair `system`|`user`, we consider as *base* the case where there are no examples and where only a `user` prompt field is available, *i.e.* "0-shot everything in `user`". We also use symmetrical log scaling in the y axis to better illustrate the relative differences. Practically, each figure gathers 12 snippets: one for each model (indicating the base score in their titles) where the number of shots (from 0 to 4) are represented using different colors and where we display relative performances by group of similar configurations. These configurations are as follows:

(1) Everything in `user` – This is the baseline use-case, in 0-shot.
(2) Everything in `system` – Having everything in `system` is an edge-case which has never been intended to, by LLM designers.
(3) Only question in `user` – Having only the question in the `user` prompt and all the rest in the `system` (*i.e.* instruction and the k examples) would correspond to a use-case where final users only have access to a solution whose administrator would have set it

[6] A reading comprehension dataset with 100k+ crowd-generated questions based on Wikipedia articles, requiring models to extract answers directly from the text.

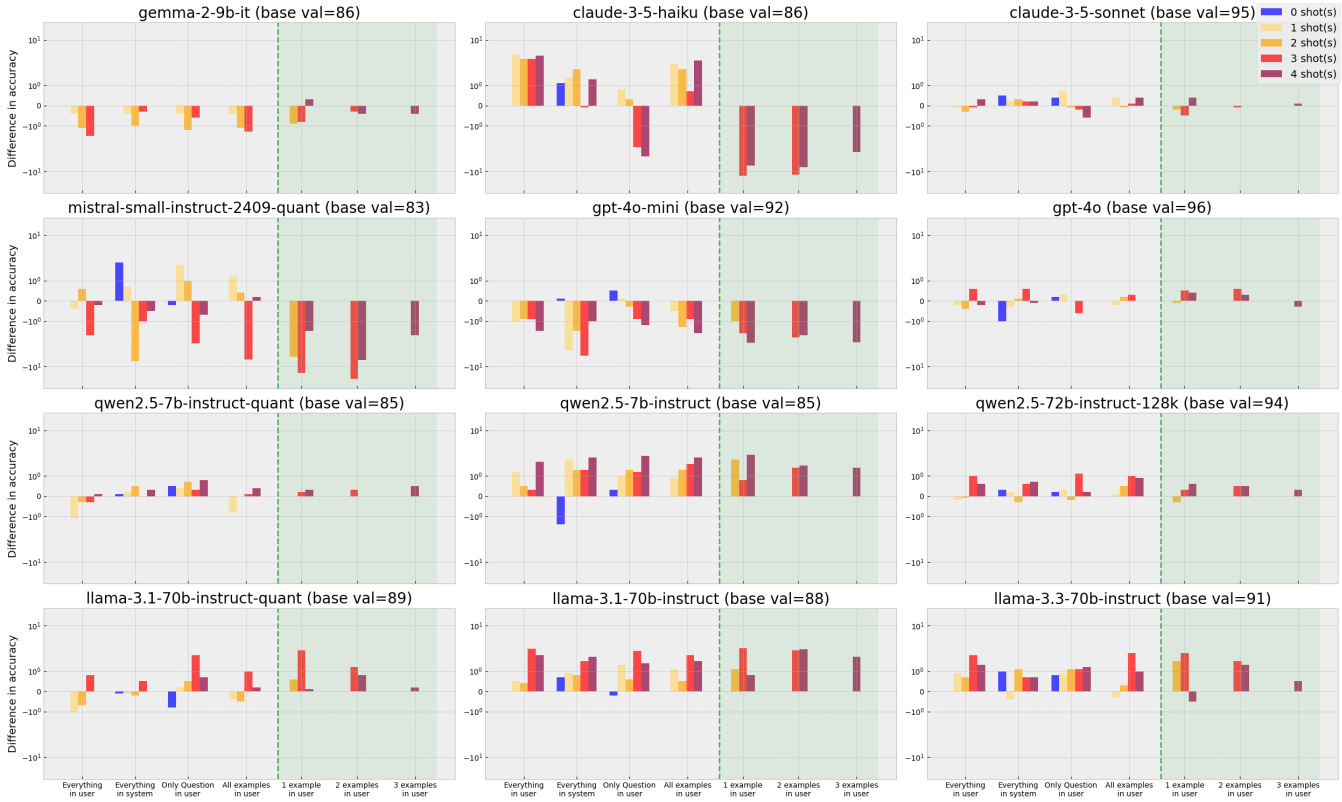

Figure 3: PIQA [3].

up *via* a complex and well-though `system` prompt, embedding potentially examples.

(4) All examples in `user` – This would correspond to a chatbot having only the instruction in the `system` and where the practitioners would come with their own set of examples.

(5) Mix of the examples between `system` and `user` (see the green background on snippets) – Depending on the number of shots, we have been able to leave some examples in the `system` while putting over in the `user`.

Practically, we always use the same set of examples for each dataset, *e.g.* shot#2 remains the same in the 2-,3- and 4-shot settings.

BoolQ (Figure 1). For this dataset, no matter the configurations, $^2/_3$ of the models show improvements over the baseline. It is worth noting that OpenAI models[7] and Qwen-72B preferred the `user`-0-shot which would suggest that their pre-training let them remember perfectly BoolQ and any extra-shots is therefore noisy to them. In term of *sweetspotness*, it seems that with yes/no datasets it's better to have several shots showing the plurality of answer, so not to lock LLMs in returning always the same value. When it comes to shot-distribution, best results tend to be obtained when all the examples are in the `system` and the query in `user` or when one or two examples accompany the question in the `user`.

MMLU-Pro (Figure 2). In this multiple-choice-answer case, we notice that none of the base scores are above 46, and rather ranger within [31, 46], this means that the dataset is challenging and not remembered from the models' pre-trainings. For this MMLU-Pro, more than with the other ones, the choice of the examples seem to influence a lot the results, such a behaviour being visible when when adding new shots return the situations from positive to negative difference in accuracy. Finally, we can see that the configurations which consistently provide improvements are: "everything in `system`" and "only question in `user`".

PIQA (Figure 3). When it comes to common-sense reasoning, apart for Gemma and GPT-4o-mini, it is clear that involving examples guides the LLMs and improves their performances. That being stated, not all the configurations lead to improvement across all models. Similarly, as with the previous datasets, there are no configurations which consistently improve all the results. Nonetheless, it seems that having all the examples in `system` usually helps, sometimes with the query too. Finally, it is worth noting that the variations with PIQA are usually of less than 5 points, this due to the fact that the base scores are above 80%, often above 90%.

SQuAD-v2 (Figure 4). This dataset is the one for which the variations between the configurations are the smallest, see Figure 5 and

[7] Moreover, GPT-4o-mini decided to answer *False* for everything, despite having some *True* in few-shot examples for `system`-message-only cases…

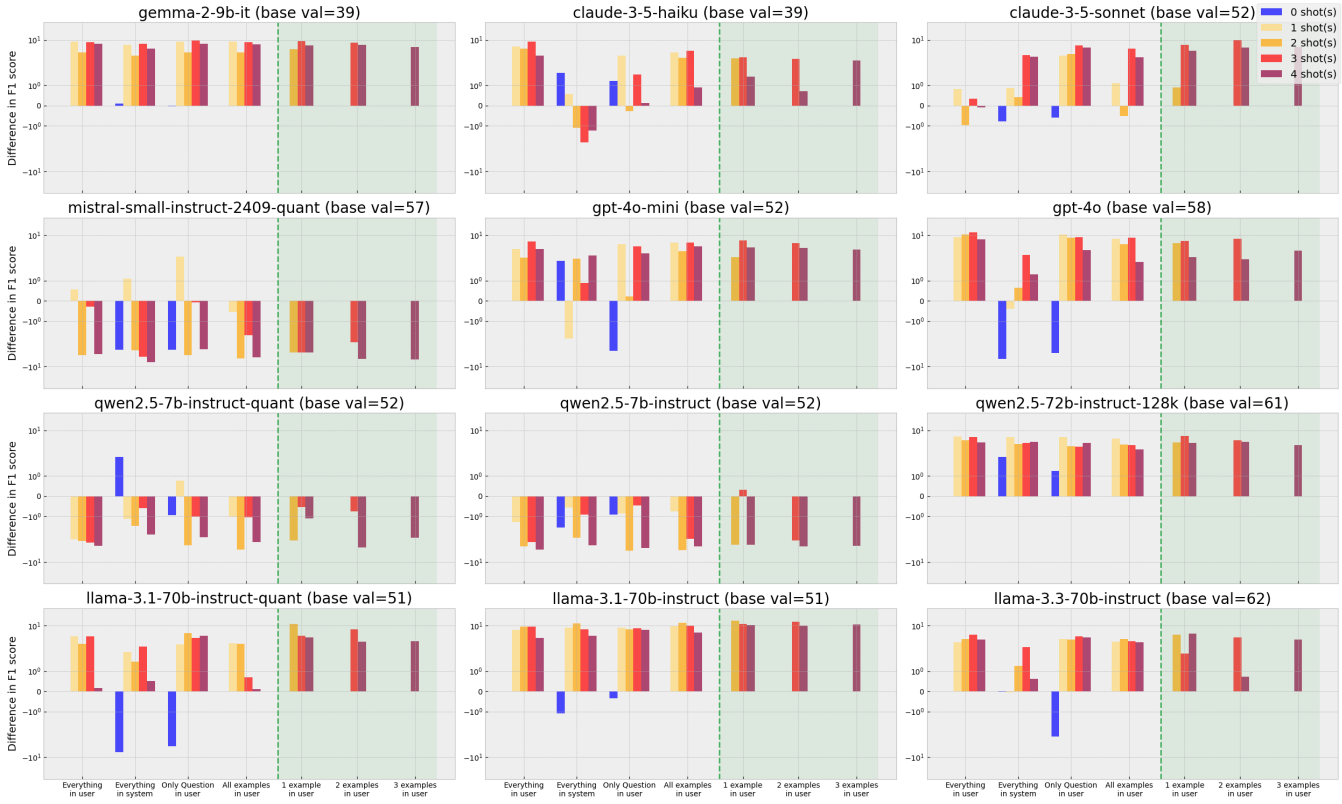

Figure 4: SQuAD-v2 [18].

the amount of blue cells (more details in Section 3.3). As a consequence, if shots are helping, then on average the all configurations tend to offer improvements.

3.3 Analysis of sys-user split

The focus of our study examines the importance of prompt placement between `system` and `user`. To quantify these differences more concretely, we look to calculate p-values to assess whether `sys-user` split variations are statistically significant. We start with a null hypothesis assuming no differences in `sys-user` split impact and calculate its probability (p-value). For binary outcome datasets like BoolQ, PIQA, and MMLU Pro, we use Cochran's Q test to compare related groups with binary responses. For the SQuAD dataset, where performance is measured by F1 scores, we apply ANOVA to assess differences in mean scores across methods.

Our findings, illustrated in Figure 5, reveal that 40% of all experiments yield a p-value of less than 0.05. This threshold is commonly used to reject the null hypothesis, indicating a statistically significant difference in the impact of `sys-user` prompt placement. This finding highlights that the use of the correct `sys-user` split can enhance prompt effectiveness, and yet the area remains largely underexplored in the current body of literature. This oversight suggests a promising avenue for future research, where a deeper understanding and systematic exploration of `sys-user` splits could lead to more effective prompt engineering methodologies.

Analyzing model performance, we observe that the Qwen suite demonstrates robustness to `sys-user` prompt variations on datasets like SQuAD but shows vulnerability on BoolQ. Conversely, the Gemma2-9b-it model exhibits consistent stability across all benchmarks. We are unsure as to what causes these patterns however we suspect it likely stems from how the models have been post-trained.

3.4 Discussion towards finding a sweet spot

We explored variations in both shot count and `system-user` split configurations. Regarding shot count, our findings indicate that the most effective prompting strategy typically involves 2-3-4 shots. This observation aligns with previous research on large language models, which suggests that in-context learning is beneficial [4].

From observing the results for `system-user` split variations we see the "Only question in user" configuration is slightly more favorable than the rest, as it exhibits the least variation across different shot counts and often ranks as the top-performing setup. To show this, we present in Table 1 the normalised scores of each prompt configuration. However, this conclusion should be approached with caution as we also see many cases where this is not optimal. The more prudent takeaway is that with the current state-of-the-art models, it is worthwhile to experiment on a case-by-case basis to determine the optimal split for enhancing performance.

Regarding the comparison between quantized and unquantized models, our analysis reveals negligible differences. The base_val

	0 shot	1 shot	2 shot	3 shot	4 shot
Everything in user	4808	4815	4798	4809	4778
Everything in system	**4830**	4749	4768	4765	4778
Only Question in user	4763	**4848**	**4839**	**4872**	**4869**
All examples in user	—	4788	4775	4788	4781
1 example in user	—	—	4821	4791	4818
2 examples in user	—	—	—	4776	4779
3 examples in user	—	—	—	—	4798
Standard Deviation	34.32	42.08	29.97	38.08	33.92

Table 1: Normalised scores of each prompt configuration.

scores are nearly identical across all benchmarks, and the overall trend in model performance remains consistent across various prompting methods.

Overall, it seems that the safest approach to design the best prompting strategy is to: First assess whether few-shot prompting would help, if not then better to 0-shot in `user` (see **blue** bars in Figures 1, 2, 3, and 4 which do not show consistent behaviors). Second, in case of few-shot scenario, it seems that either 2 or 4 shots work better and preferably in the `system` prompt (lowest standard deviations as depicted in Table 1). The question of where having the query is more opened but statistically, placing it in `user` seems better (see Table 1).

4 Related Work

As shown by [4], language models are few-shot learners. Their claims hold with other sub-domains, such as *health* [15], *programming* [11, 26] and *embodied task planning* [20].

Due to the limiting context length of previous LLMs, earlier studies only scaled the number of few-shot examples for in-context learning (ICL) up to 100 [4], however two recent studies [2, 12] (many-shot ICL) have scaled the number of shots into the thousands.

The former study of the two [12] showed that randomly selected few-shot examples can improve inference performance on LLMs even without additional fine-tuning and that larger models containing more knowledge tend to benefit more from the provided few-shot examples.

The latter study [2] further experimented with the hypothesis that few-shot examples help to generate more task-specific vectors from the input queries [9] that are more likely to activate latent concepts that LLMs have acquired during pre-training [22, 24], more specifically, they showed that by simply providing example queries without corresponding solutions (unsupervised ICL) as shots, with enough examples the resultant performance boost can be comparable to human-written ones in certain cases.

Results from these two studies along with the observation that LLMs tend to pick up information mainly at the start and end of a long prompt [13] would strongly suggest that few-shot (or many-shot) ICL benefit more from the more task-specific vectors of input queries formed by concatenating the actual query with few-shot examples of the same task than the information provided in those examples.

While these previous works have shown that ICL examples that are relevant to the task but not necessarily providing required information to answer a specific query can help with answering

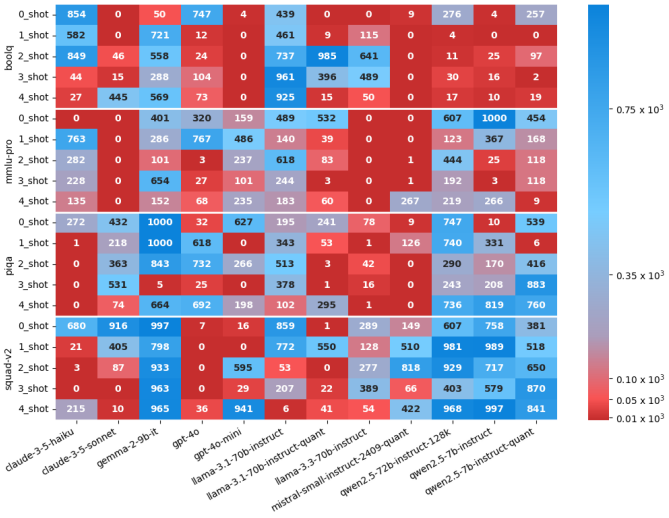

Figure 5: Statistical significance heatmap (p-value x10^3).

the query, it remains unclear how prompt templates (content surrounding the actual query and few-shot examples) would affect the performance of LLMs.

ExpertPrompting [25] experimented with the role-playing persona section of `system` prompts included at the start of a conversation (*e.g.* "You are a helpful assistant…") and showed improved performance by providing a suitable expert identity as persona as well as high-level descriptions of the ability this expert is expected to possess. Another research [27] experimented with a large number of personas without descriptions of expected abilities of the personas and concluded that the optimal persona to assign are query-specific with most persona selection strategies performing similarly to random guesses. These conclusions aligned with the previous hypothesis that having more task-specific input vectors would activate the learnt latent concepts better, yielding more accurate responses.

The most relevant study to our work [8] showed that with the same included information, the structure and syntax of the initial system messages have a significant effect on LLMs' performance, with no prompt template that excels universally. However, there does not seem to exist gold standard for the location to put in few-shot examples for ICL, and all previous works simply defaulted to providing the few-shot examples in location they seem fit without evidences to support these decisions, for example [8] put few-shot examples only within `system` prompts while PromptWizard [1] put them exclusively in `user`.

5 Conclusion

While most models now support two input formats with both `system` and `user` instructions, there has been little research on how best to utilize this split to optimize performance. In this study, we conduct experiments on four diverse datasets to observe how the latest models react to the placement of instructions, examples, and questions between the system and user prompts. Our findings reveal a noticeable difference in LLM performance, with 40% of

cases showing a statistically significant change simply by altering the placement. Across our experiments, the prompt configuration that generally yields the best results involves placing only the question in the `user` prompt, while the instructions and examples are included in the `system` prompt.

While our work encompasses diverse benchmarks and model selections, we maintained a simple prompt and instruction format to ensure experimental consistency and minimize the introduction of extraneous variables. However, real-world applications often involve models that are utilized with user-defined expert personas, given access to tools, or required to create structured outputs. We did not explore how these variables could impact our results; however, this could be explored as a future direction. Additionally, we believe it would be worthwhile to investigate whether prompt optimizers can benefit from adjusting the system-user split on a case-by-case basis, as we observed considerable variability depending on the use case, despite identifying a preferred split.

References

[1] Eshaan Agarwal, Joykirat Singh, Vivek Dani, Raghav Magazine, Tanuja Ganu, and Akshay Nambi. 2024. PromptWizard: Task-Aware Prompt Optimization Framework. arXiv:2405.18369 [cs.CL] https://arxiv.org/abs/2405.18369

[2] Rishabh Agarwal, Avi Singh, Lei M. Zhang, Bernd Bohnet, Luis Rosias, Stephanie Chan, Biao Zhang, Ankesh Anand, Zaheer Abbas, Azade Nova, John D. Co-Reyes, Eric Chu, Feryal Behbahani, Aleksandra Faust, and Hugo Larochelle. 2024. Many-Shot In-Context Learning. arXiv:2404.11018

[3] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 7432–7439.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[5] Manish Chandra, Debasis Ganguly, and Iadh Ounis. 2024. One size doesn't fit all: Predicting the number of examples for in-context learning. (2024). https://arxiv.org/abs/2403.06402

[6] Manish Chandra, Debasis Ganguly, and Iadh Ounis. 2025. One size doesn't fit all: Predicting the Number of Examples for In-Context Learning. doi:10.48550/arXiv.2403.06402 arXiv:2403.06402.

[7] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044* (2019).

[8] Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does Prompt Formatting Have Any Impact on LLM Performance? arXiv:2411.10541 [cs.CL] https://arxiv.org/abs/2411.10541

[9] Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-Context Learning Creates Task Vectors. arXiv:2310.15916 [cs.CL] https://arxiv.org/abs/2310.15916

[10] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv:2310.06825* (2023).

[11] Sungmin Kang, Juyeon Yoon, and Shin Yoo. 2023. Large Language Models are Few-shot Testers: Exploring LLM-based General Bug Reproduction. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. 2312–2323. doi:10.1109/ICSE48619.2023.00194

[12] Mukai Li, Shansan Gong, Jiangtao Feng, Yiheng Xu, Jun Zhang, Zhiyong Wu, and Lingpeng Kong. 2023. In-Context Learning with Many Demonstration Examples. arXiv:2302.04931 [cs.CL] https://arxiv.org/abs/2302.04931

[13] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2023. Lost in the Middle: How Language Models Use Long Contexts. arXiv:2307.03172 [cs.CL] https://arxiv.org/abs/2307.03172

[14] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *Comput. Surveys* 55, 9 (2023), 1–35.

[15] Xin Liu, Daniel McDuff, Geza Kovacs, Isaac Galatzer-Levy, Jacob Sunshine, Jiening Zhan, Ming-Zher Poh, Shun Liao, Paolo Di Achille, and Shwetak Patel. 2023. Large Language Models are Few-Shot Health Learners. arXiv:2305.15525

[16] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155

[17] P Rajpurkar. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[18] Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822* (2018).

[19] Laria Reynolds and Kyle McDonell. 2021. Prompt Programming for Large Language Models: Beyond the Few-Shot Paradigm. arXiv:2102.07350 [cs.CL] https://arxiv.org/abs/2102.07350

[20] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. 2023. LLM-Planner: Few-Shot Grounded Planning for Embodied Agents with Large Language Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 2998–3009.

[21] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[22] Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large Language Models Are Latent Variable Models: Explaining and Finding Good Demonstrations for In-Context Learning. arXiv:2301.11916 [cs.CL] https://arxiv.org/abs/2301.11916

[23] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574* (2024).

[24] Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An Explanation of In-context Learning as Implicit Bayesian Inference. arXiv:2111.02080 [cs.CL] https://arxiv.org/abs/2111.02080

[25] Benfeng Xu, An Yang, Junyang Lin, Quan Wang, Chang Zhou, Yongdong Zhang, and Zhendong Mao. 2023. ExpertPrompting: Instructing Large Language Models to be Distinguished Experts. arXiv:2305.14688 https://arxiv.org/abs/2305.14688

[26] Derek Xu, Tong Xie, Botao Xia, Haoyu Li, Yunsheng Bai, Yizhou Sun, and Wei Wang. 2024. Does Few-Shot Learning Help LLM Performance in Code Synthesis? arXiv:2412.02906 [cs.SE] https://arxiv.org/abs/2412.02906

[27] Mingqian Zheng, Jiaxin Pei, Lajanugen Logeswaran, Moontae Lee, and David Jurgens. 2024. When "A Helpful Assistant" Is Not Really Helpful: Personas in System Prompts Do Not Improve Performances of Large Language Models. arXiv:2311.10054 [cs.CL] https://arxiv.org/abs/2311.10054

A Benchmarking Details

In order to facilitate reproducibility, we provide in this Appendix our instructions and examples which were used for the experiments presented in Section 3. Typically, as mentioned in Section 2, the general structure of our prompts remains the same across all the configurations (and for the four tested benchmarks [3, 7, 18, 23]), we fix the instruction and the example set and "just" vary the number of examples and the frontier between `system` and `user` prompts:

$$\left[instruction \ \ldots \ shot_i \ \overset{\texttt{sys}\leftarrow}{\|}\overset{\rightarrow\texttt{user}}{} \ shot_{i+1} \ \ldots \ query\right]$$

A.1 Instructions for each dataset

```
boolq_instruction = 'Answer the current question by only
    returning True or False.'
mmlu_pro_instruction = 'Answer the current question by
    only returning the letter corresponding to the
    correct choice.'
piqa_instruction = 'Given a question and 2 possible
    solutions, pick the most appropiate solution, of
    which exactly one is correct. When providing your
    answer return only the letter corresponding to the
    correct choice.'
squad_instruction = 'Given a question and paragraph,
    determine if the paragraph contains the answer to
    the question. If the answer IS NOT contained in the
    paragraph simply return False. If the answer IS
    contained in the paragraph, return the answer to
    the question by quoting it from the paragraph.'
```

A.2 The Selected 4 Examples for each dataset

PIQA [3]

Q: To ensure the Vertical Strawberry Planter drains. Options:
- (A): Drill holes into the bottom of the base.
- (B): Cut trenches into the bottom of the base.

Ans: A
Q: make crisp french fries. Options:
- (A): after cutting, let the raw potatoes stand in cold water for at least 30 minutes before frying
- (B): let the fries sit in the freezer for six hours before frying, and remove when they are just beginning to turn colors.

Ans: A
Q: How to keep thread from showing too much in a sewing project? Options:
- (A): To keep your thread from showing too much in your sewing project, try to keep the thread the opposite color as the fabric so if it does show through, it wouldn't be as noticeable.
- (B): To keep your thread from showing too much in your sewing project, try to keep the thread as close to the same color as the fabric so if it does show through, it wouldn't be as noticeable.

Ans: B
Q: how to make sausage-apple biscuit Options:
- (A): Spread a split buttermilk biscuit with apple butter and sandwich with a cooked sausage patty.
- (B): Spread a split buttermilk biscuit with applesauce and chocolate sauce and sandwich with a cooked sausage patty.

Ans: A

MMLU-Pro [23]

Q: The symmetric group S_n has $factorial\{n\}$ elements, hence it is not true that S_{10} has 10 elements. Find the characteristic of the ring 2Z. Options:
- (A): 0
- (B): 30
- (C): 3
- (D): 10
- (E): 12
- (F): 50
- (G): 2
- (H): 100
- (I): 20
- (J): 5

Ans: A
Q: Let V be the set of all real polynomials p(x). Let transformations T, S be defined on V by T:p(x) -> xp(x) and S:p(x) -> p'(x) = d/dx p(x), and interpret (ST)(p(x)) as S(T(p(x))). Which of the following is true? Options:
- (A): ST + TS is the identity map of V onto itself.
- (B): TS = 0
- (C): ST = 1
- (D): ST - TS = 0
- (E): ST = T
- (F): ST = 0
- (G): ST = TS
- (H): ST - TS is the identity map of V onto itself.
- (I): TS = T
- (J): ST = S

Ans: H
Q: Let A be the set of all ordered pairs of integers (m, n) such that 7m + 12n = 22. What is the greatest negative number in the set $B = \{m + n : (m, n) \in A\}$? Options:
- (A): -5
- (B): 0
- (C): -3
- (D): -7
- (E): -4
- (F): -6
- (G): -1
- (H): -2
- (I): -9
- (J): N/A

Ans: E
Q: A tank initially contains a salt solution of 3 grams of salt dissolved in 100 liters of water. A salt solution containing 0.02 grams of salt per liter of water is sprayed into the tank at a rate of 4 liters per minute. The sprayed solution is continually mixed with the salt solution in the tank, and the mixture flows out of the tank at a rate of 4 liters per minute. If the mixing is instantaneous, how many grams of salt are in the tank after 100 minutes have elapsed? Options:
- (A): 3 + e^-2
- (B): 2 - e^-4
- (C): 2 - e^-2
- (D): 3 + e^-4
- (E): 2 + e^-3
- (F): 2 - e^-3
- (G): 3 - e^-2
- (H): 2 + e^-2
- (I): 2 + e^-4
- (J): 2

Ans: I

BoolQ [7]

Q: do iran and afghanistan speak the same language
Ans: True
Q: do good samaritan laws protect those who help at an accident
Ans: True
Q: is windows movie maker part of windows essentials
Ans: True
Q: is confectionary sugar the same as powdered sugar
Ans: True

SQuAD-v2 [18]

Q: After cellulose, what component is most plentiful in wood?
Paragraph: Aside from water, wood has three main components. Cellulose, a crystalline polymer derived from glucose, constitutes about 41-43%. Next in abundance is hemicellulose, which is around 20% in deciduous trees but near 30% in conifers. It is mainly five-carbon sugars that are linked in an irregular manner, in contrast to the cellulose. Lignin is the third component at around 27% in coniferous wood vs. 23% in deciduous trees. Lignin confers the hydrophobic properties reflecting the fact that it is based on aromatic rings. These three components are interwoven, and direct covalent linkages exist between the lignin and the hemicellulose. A major focus of the paper industry is the separation of the lignin from the cellulose, from which paper is made.
Ans: hemicellulose
Q: Which country have the Haredi and the Hasidic Jewry disowned?
Paragraph: On the other hand, Orthodox Jews subscribing to Modern Orthodoxy in its American and UK incarnations, tend to be far more right-wing than both non-orthodox and other orthodox Jews. While the overwhelming majority of non-Orthodox American Jews are on average strongly liberal and supporters of the Democratic Party, the Modern Orthodox subgroup of Orthodox Judaism tends to be far more conservative, with roughly half describing themselves as political conservatives, and are mostly Republican Party supporters. Modern Orthodox Jews, compared to both the non-Orthodox American Jewry and the Haredi and Hasidic Jewry, also tend to have a stronger connection to Israel due to their attachment to Zionism.
Ans: False
Q: How much of the Bronx's vote in 1916 did Hughes get?
Paragraph: Since then, the Bronx has always supported the Democratic Party's nominee for President, starting with a vote of 2-1 for the unsuccessful Al Smith in 1928, followed by four 2-1 votes for the successful Franklin D. Roosevelt. (Both had been Governors of New York, but Republican former Gov. Thomas E. Dewey won only 28% of the Bronx's vote in 1948 against 55% for Pres. Harry Truman, the winning Democrat, and 17% for Henry A. Wallace of the Progressives. It was only 32 years earlier, by contrast, that another Republican former Governor who narrowly lost the Presidency, Charles Evans Hughes, had won 42.6% of the Bronx's 1916 vote against Democratic President Woodrow Wilson's 49.8% and Socialist candidate Allan Benson's 7.3%.)
Ans: 42.6%
Q: Who is the UK an overseas territory of?
Paragraph: The 1961 volcanic eruption destroyed the Tristan da Cunha canned crawfish factory, which was rebuilt a short time later. The crawfish catchers and processors work for the South African company Ovenstone, which has an exclusive contract to sell crawfish to the United States and Japan. Even though Tristan da Cunha is a UK overseas territory, it is not permitted direct access to European Union markets. Recent[clarification needed] economic conditions have meant that the islanders have had to draw from their reserves. The islands' financial problems may cause delays in updating communication equipment and improving education on the island. The fire of 13 February 2008 (see History) resulted in major temporary economic disruption.
Ans: False