In-place Implicit In-context Learning by Neuron Amplification

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved remarkable performance through in-context learning (ICL) with demonstrations, yet these methods incur significant GPU memory and computational costs. Therefore, we consider a cost-efficient approach to implement implicit ICL such that demonstrations do not occupy space in the context. In this paper, we propose an in-place method for implicit ICL by identifying and amplifying specific neurons within the feed-forward networks of LLMs. The proposed method transfers few-shot learning capabilities to zero-shot settings through neuron perturbation. Despite the model taking zero-shot inputs, our method leads to performance approaching few-shot learning, while requiring no additional computation or memory costs. Experimental results across instruction-following and problem-solving tasks demonstrate that our approach enables implicit ICL.

1 Introduction and Related Work

Large Language Models (LLMs) have revolutionized Natural Language Processing (NLP) by achieving remarkable performance across a wide range of tasks, such as Question Answering (Zhu et al., 2021), Information Extraction (Xu et al., 2024), and Text Classification (Li et al., 2022). Their success can be attributed to in-context learning (ICL), which enables the LLMs to adapt to specific tasks by effectively leveraging contextual information (Dong et al., 2024).

ICL is an active field of research, delineated principally into two categories: (1) Training-time approaches, such as MetaICL (Min et al., 2022a) and ICL Markup (Brunet et al., 2023), use meta-learning to enhance LLMs for ICL inference. (2) Inference-time research focuses on optimizing prompts, particularly the organization of demonstrations including question-answer pairs. (Zhao et al., 2021; Lu et al., 2022). Additionally,

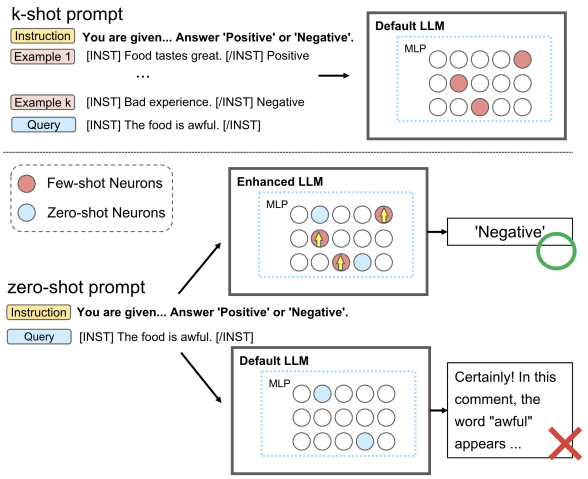

Figure 1: **Step 1 (Top):** The LLM processes few-shot prompts, identifying a set of salient neurons (red). **Step 2 (Bottom):** The examples are removed, and the salient neurons are amplified (yellow arrows). Salient neurons enable the model to generate correct outputs even without explicit examples. Conversely, the default LLM activates trivial neurons (blue), leading to unintended outputs.

Chain of Thought (CoT) largely improves LLM reasoning by refining instructions (Wei et al., 2022).

Although ICL approaches deliver excellent outcomes, the lengthy demonstrations raise GPU memory and computational costs. Therefore, *how* to reduce the costs of ICL is important. Existing context compression approaches, including implicit CoT (Deng et al., 2024) and COCONUT (Hao et al., 2024), have certain drawbacks: (1) Training is costly and lacks flexibility for diverse tasks; (2) The training objective is vague, as it's unclear whether the model learns reasoning or merely memorizes information. Therefore, we advocate for an in-place method that enhances LLM performance without extra parameters and computational costs.

Currently, extensive research focuses on the neurons within the feed-forward networks (FFN)

of LLMs. It is widely accepted that FFN layers make up the majority of a Transformer model's parameters, and they function as key-value memory systems (Geva et al., 2021). Building upon this, research has identified various kinds of neurons with unique characteristics: Knowledge Neurons play a key role in representing and expressing specific factual information (Dai et al., 2022); Skill Neurons demonstrate the capability to handle specific tasks. (Wang et al., 2022a). Motivated by research on specialized neurons, we investigate perturbing neurons to facilitate implicit in-context learning in LLMs.

In this paper, we propose a method that achieves implicit ICL by identifying and amplifying specific neurons within FFNs, as shown in Figure 1. Specifically, we first input *few-shot* prompts for a task into an LLM. Next, we identify salient neurons sensitive to the demonstrations. Then, we amplify these neurons by increasing their activation values. Finally, we feed the enhanced LLM with *zero-shot* prompts for the same task. Experimental results on instruction-following and problem-solving benchmarks demonstrate that our approach enables implicit ICL. Our contributions are as follows: (1) We introduce a novel in-place method that reduces the cost of learning from demonstrations; (2) we provide empirical results on benchmark NLP tasks that show the feasibility of our approach.

2 Method

Our method involves two key steps: identifying the neurons most sensitive to the few-shot demonstrations (§2.2), and subsequently amplifying these neurons for zero-shot inference (§2.3).

2.1 Background

Neurons in FFN. Our research focuses on LLaMA-2 (Touvron et al., 2023). Following Dai et al. (2022), we inspect FFN intermediate neurons which directly affect the output of the activation function. The effect of FFN can be represented as the following:

$$\mathrm{FFN}(\mathbf{H}) = (f(\mathbf{H}\mathbf{W}_{\mathbf{gate}}) \circ \mathbf{H}\mathbf{W}_{\mathbf{up}})\mathbf{W}_{\mathbf{down}}, \quad (1)$$

where $\mathbf{H}$ is the hidden state output from the attention module; $\mathbf{W}_{\mathbf{gate}}$, $\mathbf{W}_{\mathbf{up}}$, and $\mathbf{W}_{\mathbf{down}}$ are parameter matrices; f is the activation function; $\circ$ denotes the element-wise multiplication.

Special Tokens. Recent LLMs usually apply chat templates to structure input prompts. Particularly for LLaMA-2, each question is enclosed within a pair of tokens that delineate its boundaries (see Appendix A). Such special tokens help trigger ICL ability during inference (Brunet et al., 2023).

2.2 Identifying Salient Neurons for ICL

Inspired by previous studies on neurons, we aim to identify neurons whose activation strongly correlates to the demonstrations in the context of a task. In this step, our identification method is a variant of CGVST (Song et al., 2024). Given N special words in the chat template:

$$W = \{w_1, w_2, ..., w_N\}, \quad (2)$$

the original approach utilizes the whole set. In contrast, ours simply takes the final special word w_N, which is also the final word in the input sequence (see Appendix A). This word is further split into n tokens, forming our special token set:

$$S = \{x_{T-n+1}, x_{T-n+2}, ..., x_T\} \subset X, \quad (3)$$

where each x is a special token and X denotes an input sequence of length T.

We continue by putting the input sequence X into the LLM. In a forward pass, the model compute a loss function using the log-likelihood of predicting these special tokens S:

$$\mathcal{L} = -\sum_{t \in \{t | x_t \in S\}} \log P(x_t \mid x_1, x_2, ..., x_{t-1}). \quad (4)$$

Then, in a backward pass, we accumulate the gradient variance of the intermediate neurons by taking the derivative of the loss function with respect to the W_{gate} parameter.

The outcome is a matrix of size $l \times d \times n$, where l is the total number of layers in the model, d is the size of the hidden state and n is the intermediate size. We further compress the matrix into $\delta' \in \mathbb{R}^{l \times n}$. Consequently, the position of a neuron in the intermediate layer corresponds to its respective index in the matrix.

In each cell of the matrix δ', the value represents how much a neuron's gradient varies when the model processes few-shot prompts. We select the largest p% of cells as the salient neurons for ICL. Their indices are stored for future use.

2.3 Neuron Amplification

In the previous step, we have used an LLM for inference and have obtained locations of its salient neurons. Now, we input the same data into the model, but this time without demonstrations. Inside the LLM, we enhance the influence of salient neurons by multiplying their activation values:

$$\text{FFN}(\mathbf{H}) = (\mathbf{A} \circ f(\mathbf{H}\mathbf{W}_{\mathbf{gate}}) \circ \mathbf{H}\mathbf{W}_{\mathbf{up}})\mathbf{W}_{\mathbf{down}}, \quad (5)$$

where matrix $\mathbf{A}$ acts as a mask which enlarge the activation values by a factor of α such that $\alpha > 1$ for the salient neurons *only*. It keeps the activation values unchanged otherwise.

Two hyperparameters, α and p, greatly impact the performance of the proposed pipeline. To search for the best outcomes, we further optimize the two variables using this grid search algorithm:

Algorithm 1 Optimize α and p

```
Initialize temp ← 0, α̂ ← 0, p̂ ← 0.
for each p in {0.05n}_{n=1}^{10} do
  for each α in {0.1n + 1}_{n=1}^{15} do
    Amplify LLM with p% of salient neurons
    Generate outputs and Evaluate acc
    if acc > temp then
      Update temp, α̂, and p̂.
Output: α̂, p̂
```

3 Experiments

To examine the feasibility of LLMs performing implicit ICL with the assistance of amplified neurons, we conduct experiments on multiple-choice tasks and several general NLP tasks. We use 5-shot prompts to identify salient neurons by default, and we always evaluate the enhanced LLM with zero-shot prompts.

3.1 Datasets

We cherry-pick six specific tasks from the Super-NaturalInstructions dataset to evaluate the instruction-following capabilities of our enhanced model (Wang et al., 2022b). These include *tweetqa classification* (TC), *overruling legal classification* (LC), *creak commonsense inference* (CI), *yelp polarity classification* (PC), *summarization* (S), and *drug extraction ade* (DE).

We also evaluate our approach on all 57 tasks in MMLU to assess its effectiveness in problem-solving. (Hendrycks et al., 2020).

3.2 Implementation details

Model: We use the LLaMA-2-chat-7B model for all our experiments on a platform with two 24GB NVIDIA GPUs.

Metrics: Our major evaluation metric is Exact Match accuracy for a single run. To provide a comprehensive comparison, a "Total Diff." score is also calculated to find the gap between baselines and zero-shot prompting. Formally, given a row r and column c in Table 1:

$$\text{Total\_Diff.}(r) = \sum_{c=1}^{7}(m_{r,c} - m_{2,c}). \quad (6)$$

Dataset Splits: In our experiments, we divide the dataset into training and test sets (see Appendix B for more details). The training set is used to identify salient neurons, while the test set is used to evaluate the enhanced model. Notably, every training set also acts as the validation set for optimization of the hyperparameters.

Label Extractor: In all experiments, we instruct the LLM to generate the answer directly. After receiving the responses, we process them using a label extractor. The extractor accepts the answer only if it follows the correct format as described in the prompt. The extracted label is then compared with the ground truth to compute the accuracy.

3.3 Baselines

We compare our identification method with the following baselines, while the amplification method remains the same for all: (1) *Random*: We randomly sample 25% of the neurons from the intermediate layer, and amplify them with a fixed factor of 1.7. (2) *LAPE* (Tang et al., 2024): We calculate the LAPE score. For each neuron, it estimates a activation probability which is later used to calculate an entropy score. Even though this method's original purpose is to find language-specific neurons, it serves as a good reference in our problem setting due to its focus on activation values. (3) *CGVST* (Song et al., 2024): We compute the original CGVST with all available special tokens.

4 Results and Analysis

In this section, we report the results of our main experiments (§4.1) and analysis (§4.2).

4.1 Main Results

We show the results of the main experiments in Table 1. Experimentally, our method successfully

Method	Dataset							
	T.C.	L.C.	C.I.	P.C.	S.	D.E.	MMLU	Diff.
5-shot	83.9	75.8	81.1	89.0	83.4	89.8	43.6	494.2
Zero-shot	0.1	0.4	13.0	2.0	0.2	1.4	35.2	0.0
Random	1.2	1.2	13.2	0.1	4.5	0.6	25.1	-6.4
LAPE	23.2	1.4	11.3	**40.5**	1.4	4.9	35.8	66.1
CGVST	0.5	**38.2**	**19.0**	13.7	**14.9**	2.8	34.8	71.6
ours	**27.2**	14.2	8.6	17.6	6.4	**43.3**	**35.9**	**100.7**

Table 1: Exact Match Accuracy (%) on seven tasks (TC, LC, CI, PC, S, DE, MMLU) for four methods based on neuron amplification (including ours). The "Diff." column represents the Total Diff. scores. 5-shot and zero-shot results are presented as upper and lower bounds of performance.

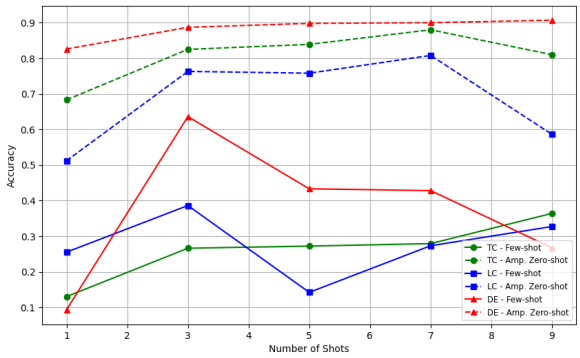

Figure 2: Performance trends based on the number of shots, measured across three tasks (TC, LC, and DE) using **Few-shot learning** (dashed lines) and **our method** (solid lines)

implement implicit ICL, as shown by consistent improvement over *Random* and *Zero-shot*. It is also the most stable one among its neuron-based counterparts, as indicated by a Total Diff. of 100.7. Several key findings are described below.

Neuron Amplification is a feasible way to implicit ICL. In all three neuron-based methods, we observe a noticeable amount of performance gain compared to *Zero-shot*, even when the input is the same set of prompts in both scenarios. This demonstrates that our proposed 2-step pipeline is generally effective for implicit ICL, regardless of the specific identification method.

The final special token is a better representative of context. By comparing our approach with the original CGVST in terms of Total Diff., we find that our method produces more consistent results. We hypothesize that this improvement can be attributed to concentration on the final special token. As the prefix of the answer tokens, our special token of choice participates in the sequence copying process of induction heads, which promote correct label words that appeared earlier in the context (Olsson et al., 2022).

Improved accuracies comes from correct format. As suggested by our qualitative analysis of several outputs (see Appendix C), traditional zero-shot generation can produce responses that contain the correct answer but completely ignore the format requirement. As a result, the label extractor rejects such outputs. Without demonstrations, the default model rarely respect the format. In contrast, the enhanced model responds with answers in the correct format more often. We deduce that the identified neurons store the memories of task-specific labels, and amplifying these neurons serves as a hint on the label space (Min et al., 2022b).

No learning is involved in neuron amplification. Our method does not involve any parameter updates. The neuron perturbation technique is solely responsible for the change in behavior of the LLM. We hypothesize that this occurs because the model being tested already stores memories of textual patterns closely related to the current task. The amplification simply informs the model about which memories to activate for generation.

4.2 Analysis

In this section, we explore how the performance of our method varies with the number of shots. The experiments are conducted for to 1, 3, 5, 7, and 9 shots. The outcome of regular few-shot prompting is also presented as a reference. Based on the results shown in Figure 2, we make the following observations:

Regardless of the number of shots, our method consistently outperforms vanilla zero-shot learning which never exceeds 10%. This demonstrates its effectiveness across various context lengths. However, our method does not improve with the increasing number of examples. This discrepancy arise because even original few-shot performance does not follow the expected scaling law.

5 Conclusion

In this paper, we propose a novel approach to the implicit in-context learning of LLMs. Through experiments and analysis, we demonstrate the feasibility of our method and provide insights into the fundamental learning processes underlying ICL. We hope our work contributes to a deeper understanding of LLMs.

Limitations

This study has several limitations. First, we did not explore larger sizes of LLaMA-2, which may have led to improved performance due to increased model capacity. Second, we focused solely on the LLaMA architecture and did not experiment with alternative models, such as GPT or Mistral, which could offer different strengths. Finally, we did not incorporate Chain of Thought reasoning, which has been shown to enhance complex reasoning tasks. Future work should investigate these aspects to better understand their impact on performance and robustness.

Ethinics Statement

This research, "In-place Implicit In-context Learning by Neuron Amplification," was conducted with strict adherence to ethical principles. It raises no ethical concerns or potential risks, as it utilizes only open-source data and a large language model compliant with their declared licenses and intended use. None of these contains any information that names or uniquely identifies individual people or offensive content. All participants were fully informed of the study's procedures and provided their informed consent.

References

Marc-Etienne Brunet, Ashton Anderson, and Richard Zemel. 2023. Icl markup: Structuring in-context learning using soft-token tags. *arXiv preprint arXiv:2312.07405*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *arXiv preprint arXiv:2405.14838*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S Yu, and Lifang He. 2022. A survey on text classification: From traditional to deep learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(2):1–41.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

Ran Song, Shizhu He, Shuting Jiang, Yantuan Xian, Shengxiang Gao, Kang Liu, and Zhengtao Yu. 2024. Does large language model contain task-specific neurons? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7101–7113, Miami, Florida, USA. Association for Computational Linguistics.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

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*.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022a. Finding skill neurons in pre-trained transformer-based language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11132–11152.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022b. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A Choice of Special Tokens

This section shows how our identification method is different from CGVST. The 5-shot prompt rendered by the default chat template of LLaMA-2 is shown in Figure 3.

B Details on Datasets

The MMLU dataset consists of 57 tasks in 4 major groups: "Social Science," "STEM," "Humanities," and "Others." We divide the data into two equal halves. The first half of the tasks form 4 combined training set, while the remaining half serves as the test set for each task. Similarly, the original data for few-shot learning is also combined and resampled to produce k examples for each category. During the neuron identification phase, we obtain a set of salient neurons for each category by processing its training set with the model. In the subsequent neuron amplification phase, we evaluate the model on the test set of each task, using amplified neurons specific to the category the task's corresponding category. Overall, we split the MMLU dataset into 4 training set and 57 test set. as examples, and the remaining half is the test set.

Category	Training Set Size	Subcategory #
STEM	1507	18
Social Sciences	1536	12
Humanities	2349	13
Others	1618	14

Table 2: Statistics of MMLU Dataset Splits.

The Natural-Instructions dataset is split in a normal way. For each task, the first half of data is the training set. k instances are chosen as the few-shot examples.

C Qualitative Analysis

In Figure 4, we show three cases the default LLM generate the correct answer in its output sequence, but fails to follow the instructed format. The model generates extra tokens other than the label, so the prediction is evaluated as incorrect by the label extractor.

5-shot Prompt:

<s> [INST] <<SYS>>
In this task, you will be given sentences in which your task is to recognize the name of the drug or medicine. Drugs are substances that change a person's mental or physical state. They can affect how your brain works, how you feel and behave, your understanding, and your senses. Although there might be several correct answers, you need to write one of them.
<</SYS>>

We report a case of torsade de pointes following a single oral dose of amiodarone (1400 mg or 30 mg kg-1) administered after short intravenous loading for prevention of paroxysmal atrial flutter. [/INST] amiodarone [INST] OBJECTIVE: To describe a probable case of transient global amnesia caused by propafenone. [/INST] propafenone [INST] Development of persistent late onset asthma following treatment with captopril. [/INST] captopril [INST] Disulfiram-induced hepatitis. [/INST] Disulfiram [INST] MATERIALS AND METHODS: We present two cases of significant morbidity related to primary and secondary perforation of the bladder following two installations of epirubicin. [/INST] epirubicin [INST] Subsequently, he developed hyperglycemia (fasting blood glucose 138 mg/dL) that resolved when olanzapine was stopped and recurred (fasting blood glucose 150 mg/dL) after 2 days of rechallenge with olanzapine 2.5 mg twice daily. [/INST]

Figure 3: The default chat template for LLaMA-2. The special words highlighted in yellow are the special tokens used in CGVST. Our method uses only the final special word "[/INST]", which is both highlighted in yellow and colored in red.

Prompt:

"<s> [INST] <<SYS>>\nIn this task, you are given a context tweet, a question and corresponding answer of given question. Your task is to classify given passage into two categories: (1) \"yes\" if the given context is useful in answering the question, and (2) \"no\" if the given context is not useful.\n<</SYS>> \n\n Context: just scored an exclusive w/ Sen. Cruz... he denies he was whipping against Boehner, defends mtgs w/ House Rs— Robert Costa (@costareports) July 31, 2014 Question: who is being interviewed? Answer: senator cruz [/INST]"

Output: " Based on the context provided, I would classify the passage as:\n\nYes, the"

Label: "yes"

Prompt:
"<s> [INST] <<SYS>>\nIn this task, you will be given sentences in which your task is to recognize the name of the drug or medicine. Drugs are substances that change a person's mental or physical state. They can affect how your brain works, how you feel and behave, your understanding, and your senses. Although there might be several correct answers, you need to write one of them.\n<</SYS>>\n\nSubsequently, he developed hyperglycemia (fasting blood glucose 138 mg/dL) that resolved when olanzapine was stopped and recurred (fasting blood glucose 150 mg/dL) after 2 days of rechallenge with olanzapine 2.5 mg twice daily. [/INST]"

Output: " The drug mentioned in the sentence is \"olanzapine\"."

Label: "olanzapine"

Prompt:
"<s> [INST] <<SYS>>\nIn this task, you will be shown a sentence, and you should determine whether it is overruling or non-overruling. In law, an overruling sentence is a statement that nullifies a previous case decision as a precedent by a constitutionally valid statute or a decision by the same or higher ranking court which establishes a different rule on the point of law involved. Classify your answers into overruling or non-overruling\n<</SYS>>\n\nnor did ms. shahin offer any authority interpreting rule 59(e), or any legal argument to persuade this court that the motion for reconsideration was incorrectly decided. [/INST]"

Output: " Non-overruling.\n\nThe sentence you provided does not nullify or change any"

Label: "non-overruling"

Figure 4: Several outputs of the default LLM for zero-shot prompts. The words matching the labels are highlighted in yellow.