# Clusters Emerge in Transformer-based Causal Language Models

**Xinbo Wu**
University of Illinois Urbana-Champaign
xinbowu2@illinois.edu

**Lav R. Varshney**
University of Illinois Urbana-Champaign
varshney@illinois.edu

## Abstract

Even though large language models (LLMs) have demonstrated remarkable capability in solving various natural language tasks, the capability of an LLM to follow human instructions is still an area of active development. Recent works (Ouyang et al., 2022; Rafailov et al., 2023; Zhang et al., 2023) have shown great improvements in instruction-following capability through additional training for instruction-following tasks. However, the mechanisms responsible for effective instruction-following capabilities remain inadequately understood. Here, we introduce a simplified instruction-following task and use synthetic datasets to analyze a Transformer-based causal language model. Our findings suggest that the model learns task-specific information by clustering data within its hidden space, with this clustering process evolving dynamically during learning.

## 1 Introduction

In recent years, large language models (LLMs) have achieved remarkable capabilities in natural language processing and artificial intelligence more generally (Brown et al., 2020; OpenAI, 2023; Touvron et al., 2023). However, a significant challenge with LLMs is the misalignment between their training objectives and users' intentions. Techniques such as reinforcement learning from human feedback (Ouyang et al., 2022), direct preference optimization (Rafailov et al., 2023), and instruction tuning (Zhang et al., 2023) have been proposed to further train LLMs for instruction following, yielding seemingly great instruction-following capabilities.

Yet, the mechanisms underlying these successful instruction-following capabilities are not well-understood and require specific analysis. We devise a simplified instruction-following task with a synthetic dataset that we fully control but that reflects some key properties of natural language data. We aim to perform analysis on a Transformer-based causal language model (CLM) trained for a simplified instruction-following task to study its inductive biases.

More specifically, the ability to correctly recognize a learned task may be needed to successfully execute it. We aim to investigate how task-specific information is encoded into the representation space of Transformer-based CLMs trained for instruction-following. One intuitive hypothesis is that hidden states corresponding to the same task are arranged close together to form a task-specific cluster, reminiscent of functional modules and topographic maps that neuroscientists have discovered in the brain (Knudsen et al., 1987; Chklovskii and Koulakov, 2004). Section 3 provides experimental evidence supporting this hypothesis.

## 2 Method

We define a task as a function $f : \mathcal{X} \rightarrow \mathcal{Y}$, where each input-output pair $(x, y)$ is a mapping. An instruction-following task involves predicting an output $y$ given an instruction $I$ and an input $x$. For example, in the task "given a location, state its continent. New York City," the input is "New York City," and the output is "North America." Here, the instruction helps identify the specific task function $f$. For simplicity, we assume inputs and instructions are separate. A task instance is represented as a sequence $[I; x; y]$, with each part as a text sequence. The instruction-following task is then treated as a language modeling task, where the model predicts the next token in the sequence.

To make analysis easier, we simplify the instruction-following task. We assume the input and output alphabets $\mathcal{X}$ and $\mathcal{Y}$ are discrete and represented by single tokens. We create a simplified task function by randomly sampling a finite set of input-output pairs, where each input is uniquely associated with an output. Different task functions can share the same input set but yield different outputs, making it crucial for the model to correctly identify the task. In our study, we randomly sample a regular expression for each task, which is considered a simple grammar rule. We then sample instructions represented as sequences of symbols based on the regular expression.

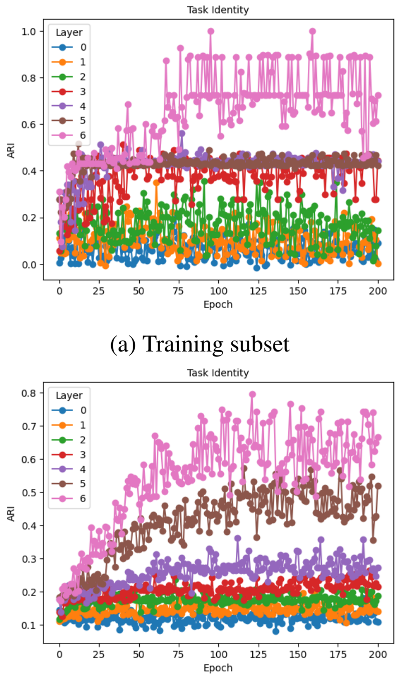
## 3 Experiments

We construct a synthetic instruction-following instruction dataset based on the guidelines outlined in Section 2. This dataset is then divided into training and validation sets. For computational efficiency in subsequent clustering analysis, we randomly sample a number of instances from a subset of tasks to form the validation set and a training subset for intermediate evaluations. Given full control over the data generation process, we record a task identity for each data instance. We construct 50 tasks in total and each task has 152 variants of instructions on average.

We train a six-layer Transformer model following the GPT-2 architecture (Radford et al., 2019). This model is optimized using an AdamW optimizer (Loshchilov and Hutter, 2017) and employs a cosine annealing learning rate schedule. The task accuracy is measured by the percentage of correct outputs, which is treated as the measurement of task performance.

We gather hidden states of the input tokens from various data instances. Next, we use the popular KMeans clustering algorithm to uncover clusters within the data. We optionally pre-process them using t-SNE dimension reduction (Van der Maaten and Hinton, 2008) if it benefits the subsequent clustering performance. We conduct extrinsic clustering evaluation on the clustering results, using task identities as labels. Our analysis reports results on the training subset and validation set, employing the Adjusted Rand Index (ARI) as a metric.

As shown in Figure 1, on both training and validation splits, there exists a strong trend of improvement of the clustering performance based on task identities throughout the training process until saturating at some high values. Moreover, clustering performance tends to improve in higher layers of the Transformer model, with the 0th layer serving as a baseline solely based on input word embedding. Notably, the baseline does not undergo much change during the training process compared to the clustering performances of other layers. It is important to note that task identities are concealed from the training process, and the Transformer models perform clustering during training without explicit supervision. Moreover, we designed the simplified task to have many tasks share the same inputs by using a small task-related vocabulary such that the model will not be able to identify a task solely from the inputs. Additionally, similar clustering phenomena are observed on the validation sets, indicating that the clustering effect generalizes to unseen instances as well. These results not only provide compelling evidence supporting the existence of task-specific clusters but also show that the clusters evolve throughout the training process instead of appearing spontaneously.



(a) Training subset

(b) Validation set

Figure 1: Clustering analysis on both of training subset (a) and validation set (b) across different layers throughout the training process. Each dot represents a data point.

## 4 Conclusion

In this work, we introduce a simplified instruction-following task and construct synthetic datasets to analyze a Transformer-based CLM model. From the simplified setting, we provide experimental evidence supporting the notion that the model encodes task-specific information through clustering in its hidden space, and demonstrate that this clustering evolves continuously during the learning process.

## References

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*, 33:1877–1901.

Dmitri B. Chklovskii and Alexei A. Koulakov. 2004. Maps in the brain: What can we learn from them? *Annual Review of Neuroscience*, 27:369–392.

Eric I. Knudsen, Sascha du Lac, and Steven D. Esterly. 1987. Computational maps in the brain. *Annual Review of Neuroscience*, 10:41–65.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

OpenAI. 2023. GPT-4 technical report. *arXiv:2304.01852*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and efficient foundation language models. *arXiv:2302.13971*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.