From Language Modeling to Instruction Following: Understanding the Behavior Shift in LLMs after Instruction Tuning

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved remarkable success, where instruction tuning is the critical step in aligning LLMs with user intentions. In this work, we investigate how the instruction tuning adjusts pre-trained models with a focus on intrinsic changes. Specifically, we first develop several local and global explanation methods, including a gradient-based method for input-output attribution and techniques for interpreting patterns and concepts in self-attention and feed-forward layers. The impact of instruction tuning is then studied by comparing the explanations derived from the pre-trained and instruction-tuned models. This approach provides an internal perspective of the model shifts on a human-comprehensible level. Our findings reveal three significant impacts of instruction tuning: 1) It empowers LLMs to recognize the instruction parts from user prompts, and promotes the response generation constantly conditioned on user instructions. 2) It encourages the self-attention heads to capture more word-word relationships about instruction verbs. 3) It encourages the feed-forward networks to rotate their pre-trained knowledge toward user-oriented tasks. These insights contribute to a more comprehensive understanding of instruction tuning and lay the groundwork for future work that aims at interpreting and optimizing LLMs for various applications.

1 Introduction

The remarkable capability of Large Language Models (LLMs) to align with user intentions is well-recognized across various real-world applications, where they are expected to be helpful, honest, and harmless AI assistants (Ouyang et al., 2022; OpenAI, 2023). Central to these roles, being "helpful" is the most fundamental requisite, emphasizing that LLMs should help users to complete various tasks, known as the "instruction following" capability. Many studies (Raffel et al., 2020; Wang et al., 2022; Zhou et al., 2023) show that *instruction tuning*, also called supervised fine-tuning (Ouyang et al., 2022), is critical to acquire such capability, by fine-tuning pre-trained models on high-quality prompt-response pairs. However, the impact of instruction tuning on the helpfulness of language models remains inadequately understood, limiting the improvements toward better AI assistants.

In this work, we focus on exploring how instruction tuning influences pre-trained models. Specifically, how do instruction-tuned models utilize the instruction words to guide their generation in a way that differs from pre-trained models? Step further, how do self-attention heads and feed-forward networks contribute to this difference by adapting their pre-trained knowledge, respectively?

However, technically answering these questions by interpreting LLMs is non-trivial. For the first question, we aim to quantify the importance of prompt words to response words, known as attribution explanations. Existing work (Selvaraju et al., 2016; Sundararajan et al., 2017; Mu and Andreas, 2020) is proposed for the classification problems, which is not suitable for auto-regressive LLMs. For the second question, we seek to interpret both self-attention and feed-forward layers within LLMs. The straightforward method (Dar et al., 2022; Geva et al., 2021) of projecting weight vectors into the word embedding space and then selecting the most activated words as explanations is compromised by the polysemic nature of model weights (Arora et al., 2018; Scherlis et al., 2022), leading to unclear and not concise explanations. Other researchers switch to studying internal activations of the models, such as heatmap visualization (Vig, 2019), sparse auto-encoder decomposition (Bricken et al., 2023b; Cunningham et al., 2023), and knowledge probing (Belinkov et al., 2018; Jawahar et al., 2019), while they may yield biased explanations due to the potential bias in the chosen samples for collecting activations. Overall, existing explanation methods cannot be directly applied to auto-regressive LLMs.

To fill these gaps, we first develop a series of explanation methods as a *toolbox* to study LLMs, including a gradient-based method for prompt-response attributions and techniques to interpret the patterns and concepts in self-attention heads and feed-forward networks at a human-understandable level. We then investigate the impact of instruction tuning by comparing the explanations coming from the pre-trained and instruction-tuned models. This approach provides an internal perspective of exploring instruction tuning, distinguishing it from existing research that primarily focuses on comparing the performance of the model trained under different settings (Liang et al., 2023; Kung and Peng, 2023; Zhou et al., 2023; Kirk et al., 2023). We obtain three main findings of the impact of instruction tuning as follows:

- **Finding 1:** *It enables models to recognize instruction words in user prompts and drives the generation process to be consistently conditioned on these words.* We introduce a normalization strategy to make the traditional gradient-based methods suitable for attributing response words to prompt words. We observe that instruction words, such as "Fix grammar errors:", influence multiple response words across different positions, unlike other words that have a limited effect on the response (Sec. 4.1). Additionally, we leverage a density function to aggregate the overall importance of each individual prompt word. This importance density score is quantitatively shown to correlate strongly with the models' ability to follow instructions (Sec. 4.2).

- **Finding 2:** *It encourages self-attention heads to learn more word relations with instruction verbs than common verbs.* We suggest extracting word-word patterns under the local co-occurrence assumption to alleviate the polysemantic challenge in interpreting self-attention heads (Sec. 5.1). We notice a significant change in the word-word patterns within the same self-attention head after instruction tuning. Analysis shows that the word-word patterns associated with instruction verbs become more popular, especially in the bottom and middle layers, while patterns linked to commonly used verbs do not display a similar increase in popularity. This finding demonstrates that self-attention heads have a direct influence on understanding user instructions.

- **Finding 3:** *It adapts the pre-trained knowledge encoded by feed-forward networks into user-oriented tasks without changing their linguistic structures.* We propose interpreting the principal components of weight vectors to reach a "concept" level explanation of feed-forward networks (Sec. 5.2). Our analysis of these concepts spans two dimensions: user-oriented tasks[1] and linguistic levels[2] (Thomas, 2005). We find that the proportion of concepts that are suitable for specific tasks, such as writing, coding, and solving math problems, becomes significantly greater after instruction tuning. In contrast, the distribution of these concepts across different linguistic levels remains the same. This phenomenon shows that feed-forward networks adapt their pre-trained knowledge to downstream tasks by slightly rotating the basis of their representation space.

This study reveals that instruction words are crucial to instruction-tuned models because of their consistent impact on the generation process, and further emphasizes the distinctive contributions of self-attention mechanisms and feed-forward networks to this functionality. While our focus is on behavior shifts after instruction tuning, future research might also apply our toolbox to understand LLMs for various other purposes.

2 Related Work

Interpreting Language Models. Majority investigations in interpreting LLMs aimed to understand the decision-making processes of LLMs for a specific task or dataset, which involves feature attribution methods (Li et al., 2015; Vig, 2019; Kokalj et al., 2021), attention-based methods (Vig, 2019; Barkan et al., 2021), and sample-based methods (Kim et al., 2018; Wu et al., 2021). Recently, many researchers turned to understanding why LLMs can perform in-context learning (Xie et al., 2021; Olsson et al., 2022; Li et al., 2023; Wei et al., 2023; Varshney et al., 2023; Xiong et al., 2023; Duan et al., 2023). In parallel, some works delved into interpreting the internal components of LLMs, including the self-attention mechanism (Elhage et al., 2021; Sukhbaatar et al., 2019) and feed-forward networks (Press et al., 2019; Geva et al.,

[1] User-oriented tasks include "writing", "coding", "translation", and "solving math problem".

[2] Linguistic levels include "phonology", "morphology", "syntax", and "semantic".

2020; Voita et al., 2023; Petroni et al., 2019; Meng et al., 2022; Huang et al., 2023). Our work builds on these foundations, introducing novel interpretation methods tailored for modern LLMs.

Interpreting Instruction-tuned Models. Interpreting instruction tuning is still in the early stages of exploring unexpected phenomena. A notable example is the "lost-in-the-middle" effect identified by (Liu et al., 2023), which demonstrates that inserting contents in the middle of prompts often results in poor model performance.Similarly, (Zhou et al., 2023) showed that even only 1000 prompt-response pairs could significantly enhance the instruction-following capabilities of LLMs. Moreover, researchers (Liang et al., 2023; Kung and Peng, 2023; Zhou et al., 2023) find that instruction-tuned models just learn superficial patterns through instruction tuning. These observations motivate us to investigate the internal changes of instruction-tuned models, aiming to reach a comprehensive understanding that recognizes these diverse phenomena under a unified perspective.

3 Preliminary

3.1 Transformer Architecture

Considering $\mathcal{V}$ as a pre-defined vocabulary set, then X denotes an N-length prompting text and Y is a M-length response from a transformer-based language model f, where each individual token $x_n \in X$ or $y_m \in Y$ comes from $\mathcal{V}$. f is defined in a D-dimensional space, starting with an input word embedding $\mathbf{E}_i \in \mathbb{R}^{|\mathcal{V}| \times D}$ presenting input tokens in $\mathbf{X} \in \mathbb{R}^{N \times D}$. $\mathbf{X}$ goes through L transformer blocks, each containing a self-attention module and a feed-forward network. Every self-attention module includes H heads that operate in a space with D' dimensions. Each self-attention head captures word relations by $\mathbf{A}^h = \text{softmax}(\mathbf{X}\mathbf{W}_q^h(\mathbf{X}\mathbf{W}_k^h)^\top/\epsilon)$, where $\mathbf{W}_q^h, \mathbf{W}_k^h \in \mathbb{R}^{D \times D'}$ and ϵ is a constant. The aggregation of heads' outputs is $[\mathbf{A}^1\mathbf{X}\mathbf{W}_v^1; ...; \mathbf{A}^H\mathbf{X}\mathbf{W}_v^H]\mathbf{W}_o$. Each feed-forward network is defined as $\sigma(\mathbf{X}\mathbf{W}_u^\top)\mathbf{W}_p$, where σ refers to a non-linear function, and $\mathbf{W}_u, \mathbf{W}_p \in \mathbb{R}^{D'' \times D}$. Finally, the processed word embeddings dot product with the transpose of output word embeddings $\mathbf{E}_o \in \mathbb{R}^{|\mathcal{V}| \times D}$ for next word prediction.

3.2 General Experimental Settings

Language Models. We choose the LLaMA family (Touvron et al., 2023) as the focus for two reasons. Firstly, LLaMA is one of the most advanced publicly accessible pre-trained language model families. Secondly, LLaMA is the foundation for many instruction-tuned models, providing a vast array for further research. In this research, we mainly use the fully fine-tuned versions of Vicuna (Zheng et al., 2023) as the instruction-tuned model, using LLaMA (Touvron et al., 2023) as the corresponding pre-trained model[3]. We employ a greedy search (for reproduction) to generate up to 300 tokens for each input prompt.

Instruction Datasets. We collect user-oriented prompting texts from three publicly available datasets: Self-Instruct (Wang et al., 2022), LIMA (Zhou et al., 2023), and MT-Bench (Zheng et al., 2023). The Self-Instruct dataset includes 252 pairs of prompts and responses written by humans, used both for generating more pairs and as a test set. LIMA, mainly based on questions and answers from online platforms like Stack Exchange, has 1000 training pairs and 300 testing pairs. On the other hand, MT-Bench, intended only for machine evaluation, has 80 human-written pairs across eight categories but lacks a training set. Our analysis focuses on the test sets from these datasets.

4 Impact of User Prompts for Human Alignment

This section focuses on the differential treatment of user prompts by instruction-tuned models compared to pre-trained models. We introduce a gradient-based attribution approach in Sec. 4.1 to measure the importance of individual input words on specific output words. In Sec. 4.2, we compare the importance densities across various models to study their distinct in using user prompts.

4.1 Quantifying Prompt Influence on Generation Process

Method. We aim to measure the importance of each prompt token to each response token. In classification, input feature importance is typically measured by monitoring confidence changes upon its removal (Ribeiro et al., 2016; Feng et al., 2018). Treating text generation as a sequence of word classification tasks, the importance of an input token to an output token is gauged by examining confidence changes in output generation while the input token

[3]We implement these models with the code and checkpoints available from Huggingface library (Wolf et al., 2019). We use lmsys/vicuna-7b-delta-v1.1 for Vicuna.

is removed. Therefore, we define *importance* $I_{n,m}$ of input token x_n to output token y_m as:

$$I_{n,m} = p(y_m|Z_m) - p(y_m|Z_{m,/n}), \qquad (1)$$

where Z_m is the context to generate y_m by concatenating the inquire X and the first $m-1$ tokens of response Y, $Z_{m,/n}$ omits token x_n from Z_m, and $p(\cdot|\cdot)$ is the conditional probability computed by language model f. We accelerate Eq. (1) with the first-order approximation: $I_{n,m} \approx \frac{\partial f(y_m|Z_m)}{\partial \mathbf{E}_i[x_n]} \cdot \mathbf{E}_i[x_n]^\top$, where $\mathbf{E}_i[x_n]$ is the input word embedding of token x_n (check Appendix A for theoretical justification). The importance of input tokens cannot be compared across different output tokens due to its dependency on the confidence $f(y_m|Z_m)$. It's crucial to recognize that a word with a lower confidence doesn't necessarily imply it is a trivial word. Specifically, in language modeling, the likelihood of a word y given previous context x could be extended with Bayes' theorem as $p(y|x) \propto p(x|y) \cdot p(y)$. Here, semantic (non-trivial) words have a lower prior probability $p(y)$ since they are less common in the general corpus. In addition, models tend to estimate a lower conditional probability $p(x|y)$ since it is more challenging to predict such meaningful words unless they observe a very strong semantic relation. Consequently, models are typically more confident about common, less meaningful words, and less confident about semantically rich, rare words. Therefore, we propose to rescale the importance scores derived from the same output token to ensure they are comparable across different output tokens. In addition, we introduce a sparse operation over the rescaled importance to overlook the noise introduced by first-order approximation. To this end, the normalized pairwise score $S_{n,m} = \text{ReLU}\left(\left\lceil L \times \frac{I_{n,m}}{\max_{n'=1}^{N} I_{n',m}} \right\rceil - b\right)$, where $\lceil \cdot \rceil$ is the ceiling function, and $b \in [0, L]$ is a hyper-parameter determining the minimal interested importance level.

Settings. This qualitative experiment demonstrates how prompt words contribute to response generation via visualizing salient maps based on normalized pairwise importance $s_{n,m}$. We set $L = 10$ and $b = 0$ to faithfully present all information (including noise) for visualization. Figure 1 provides a pair of salient maps to the same prompt corresponding to the model-generated responses from LLaMA and Vicuna, respectively. We show more visualization cases in Appendix C.

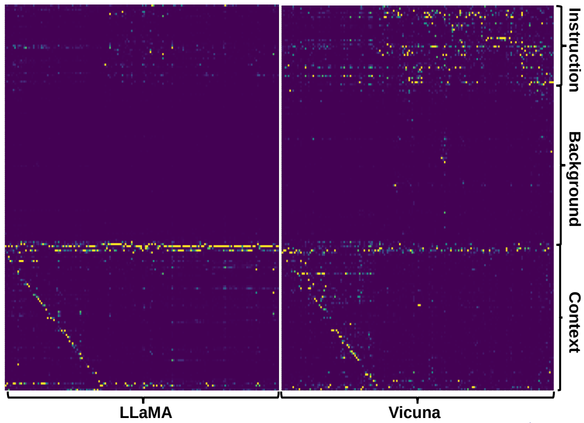


Figure 1: Salient maps of the prompt-response pair[4] from LLaMA (left) and Vicuna (right).

Obs-1: Instruction tuning helps the models distinguish between instruction and context words more accurately. We provide a visualization case that asks the models to analyze the tone (instruction) of a given email (context) into one of the listed categories (background). Both models begin their responses by repeating the email. Later, Vicuna successfully analyzes the tone of the email, while LLaMA fails to do that. Figure 1 (right) shows that the instruction part is generally brighter than the background and context part, indicating the strong influence of instruction words in shaping response generation. In contrast, context lines only light up in specific spans and show a diagonal pattern at the left button of both figures (models are repeating the email). The differences between the left and right plots further highlight the impact of instruction tuning. Specifically, the left plot has certain context lines that appear less bright in the right plot, while certain instruction lines in the right plot stand out more. This visualization case raises a hypothesis that the instruction words *constantly* contribute to the response generation if the model successfully follows the user intention. Sec. 4.2 will quantitatively verify this assumption.

4.2 Assessing Instruction Following Capability with Importance Density

Method. We aim to measure the overall attribution of each input token to the entire response generation process. Based on Sec. 4.1, an input token should acquire a greater attribution score if it is important to generate more output tokens.

[4]The prompt boldfaces its direct **instruction words** and underlines its background: **Analyze the word choice, phrasing, punctuation, and capitalization in the given email. How may the writer of this email sound to the reader?** These tones include Disheartening, Accusatory, Worried, Curious, Surprised, Disapproving, Unassuming, Formal, Assertive, Confident, Appreciative, Concerned, Sad, Informal, Regretful, Encouraging, Egocentric, Joyful, Optimistic, and Excited.\n\nInput: Hi Jen, \nI hope you're well. Can we catch up today? I'd appreciate your input on my presentation for tomorrow's meeting. I'd especially love it if you could double-check the sales numbers with me. There's a coffee in it for you!\n\nOutput:

Table 1: Importance density on instruction words over followed and unfollowed instances from Vicuna.

Dataset	Followed	Unfollowed	p-value
Self-Instruct	$1.2283_{\pm 0.52}$	$0.8917_{\pm 0.48}$	$1.4e^{-4}$
LIMA	$1.6173_{\pm 0.47}$	$1.2799_{\pm 0.44}$	$4.3e^{-6}$
MT-Bench	$1.4584_{\pm 0.55}$	$0.9290_{\pm 0.53}$	$2.3e^{-4}$

Table 2: Importance density on instruction words over responses generated by Vicuna and LLaMA.

Dataset	Vicuna	LLaMA	p-value
Self-Instruct	$1.1661_{\pm 0.53}$	$0.9432_{\pm 0.48}$	$8.4e^{-7}$
LIMA	$1.5608_{\pm 0.48}$	$1.2702_{\pm 0.43}$	$2.6e^{-14}$
MT-Bench	$1.3311_{\pm 0.59}$	$1.1697_{\pm 0.57}$	0.0804

Following this intuition, the input token x_n's attribution a_n is measured by leveraging ℓ_1/ℓ_p density function over the normalized importance to all output tokens: $a_n = ||S_n||_1/||S_n||_p$, where $S_n = [S_{n,1}, ..., S_{n,M}]$, and $p \in \mathbb{R}^+$ serves as a hyper parameter. One nice property of this density function is if two input tokens have the same total importance, then the one having greater maximum importance would receive a greater density score (check (Hurley and Rickard, 2009) for proof).

Settings. This experiment quantitatively justifies the assumption observed from Sec. 4.1 that a model aligns with human intention if it constantly uses instruction words to guide the generation. Specifically, we manually annotate a dataset, where each prompt has been marked its instruction part, and each response is labeled as either "followed" or "unfollowed". Please check Appendix B.1 for the annotation details. Here, the instruction part includes sentences that describe background information and actions for a task. On the other hand, "followed" indicates that the model provides information pertinent to the user intention, regardless of the response's factual correctness. For each prompt-response pair sourced from our datasets, we compute the importance density score with $L = 10$, $b = 7$, and $p = 4$. We further normalize the scores to ensure comparability across different instances and remove the instances with a short response (less than 5 tokens) as their estimations of density are not stable. Table 1 compares the average importance densities between the followed and unfollowed instances from Vicuna, while Table 2 compares the average importance densities between the Vicuna generated or LLaMA generated instances. Please check Appendix B.2 for an analysis of outlier cases.

Obs-2: The importance density on instruction words reflects the models' behaviors in following user intentions. From Table 1, it becomes evident that attribution scores for "followed" instances consistently outperform those of "unfollowed" across all datasets. This distinction is statistically validated by notably low p-values, where the null-hypothesis is the average importance densities of followed and unfollowed instances are equal. Table 1 underscores the strong correlation between the importance density scores of instruction words and the instruction following capability. Case studies in Appendix B.2 suggest that instruction-tuned models may pretend to follow instructions without realizing user instructions. Furthermore, Table 4.2 shows that Vicuna achieves greater importance density scores compared to LLaMA across the three datasets, indicating instruction tuning empowers the pre-trained model in better identifying and harnessing instruction words from user prompts.

Obs-3: Instruction-tuned models archive a greater importance density than their pre-trained models. Table 2 reports the average importance density over the instruction words by giving different responses generated by Vicuna or LLaMA. We could observe that Vicuna constantly assigns denser importance scores on the instruction words compared to LLaMA across the three datasets, where this improvement is validated by student-t-test, where the null hypothesis is that the average importance densities computed by responses generated by Vicuna and LLaMA are equal. According to Obs-2, we draw our conclusion that Vicuna demonstrates a better instruction-following capability than LLaMA by more accurately identifying instruction words and then successfully using them to guide response generation.

5 Shift within Instruction-tuned Models

This section studies the distinctive contributions of components in LLMs for human alignment. The self-attention heads and feed-forward networks are discussed in Sec. 5.1 and 5.2, respectively.

5.1 Analyzing Self-Attention Heads

Method. We aim to interpret the behaviors of self-attention heads with word pairs. Given a self-attention head, the relation between a pair of words (w_a, w_b) could be approximated by $\mathbf{A}_{a,b} \propto \sum_d^D \mathbf{E}_i[w_a]\mathbf{W}_q^{h\top}[d] \times \mathbf{E}_i[w_b]\mathbf{W}_k^{h\top}[d]$. Existing work (Dar et al., 2022) computes the similarities between words from the entire vocabulary $\mathcal{V}$ based on $\mathbf{W}_q$ and $\mathbf{W}_k$ and selects the top-K word pair with the greatest similarities to represent the relations encoded by a self-attention head. However, we notice that the word pairs obtained by this approach are redundant, leading to a less comprehensive understanding of the self-attention head. To overcome this problem, we propose to interpret a self-attention head by aggregating the word pairs activated by its neuron pairs, which is motivated by the fact that the relation $\mathbf{A}_{a,b}$ linearly relates to the activations of column vectors of weights $\mathbf{W}_q^h$ and $\mathbf{W}_k^h$, called neurons in this paper. But this alternative approach suffers from the polysemantic nature of neurons (Elhage et al., 2022; Bricken et al., 2023a), introducing word pairs that are meaningless to be connected. Considering the self-attention mechanism is designed for capturing word relations within the input texts, we introduce the word-word co-occurrence constraint to the word pair formation. Specifically, we first interpret neurons $\mathbf{W}_q^{h\top}[d]$ and $\mathbf{W}_k^{h\top}[d]$ by collecting the top-K words that could most activate them, i.e. $\mathcal{E}_q^d = \arg max_{\mathcal{V}' \subseteq \mathcal{V}, |\mathcal{V}'|=K} \sum_{w \in \mathcal{V}'} \mathbf{E}_i[w] \cdot (\mathbf{W}_q^{h\top}[d])^\top$ and $\mathcal{E}_k^d = \arg max_{\mathcal{V}' \subseteq \mathcal{V}, |\mathcal{V}'|=K} \sum_{w \in \mathcal{V}'} \mathbf{E}_i[w] \cdot (\mathbf{W}_k^{h\top}[d])^\top$. We then form the word pair list $\mathcal{E}_{qk}^d = \{(w_q, w_k) : cos(e_q, e_k) > \theta\}$, where $w_q \in \mathcal{E}_q^d$, $w_k \in \mathcal{E}_k^d$, $\mathbf{e}_q$, $\mathbf{e}_k$ are their GloVe (Pennington et al., 2014) word embeddings, and θ is a threshold. Finally, the explanation of a self-attention head is described with the frequent word pairs that are used to interpret its neurons.

Settings. We consider $K = 100$ as a constant and θ as dynamic values for different words. Specifically, we first compute the cosine similarity between the given word and 1000 frequent words with GloVe word embeddings (Pennington et al., 2014). The threshold of which is the average similarities plus 1.96 times the standard deviation. The threshold of a word pair is the greater one of their individual word thresholds. We conduct a qualitative analysis of the word pairs in Appendix F.

The impact of instruction tuning on self-attention heads is studied by comparing the word pair lists from the pre-trained and tuned models. First, we quantify the changes of word pair lists with the intersection rate $M = \frac{\mathcal{E}_{pt} \cap \mathcal{E}_{ft}}{\mathcal{E}_{pt} \cup \mathcal{E}_{ft}}$, where $\mathcal{E}_{pt}$ and $\mathcal{E}_{ft}$ denote the top-100 word pairs of the pre-trained and tuned models. Figure 2 visualizes $1 - M$ over various layer groups. We also investigate how these changed word pairs related to the instruction-following capability, focusing on verbs.

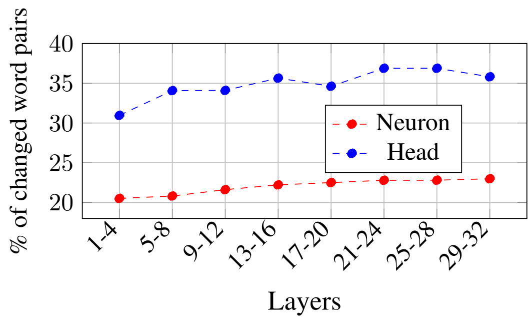


Figure 2: Shift of word-word patterns for self-attention after instruction tuning.

Table 3: Percentage of self-attention heads encoding certain verbs after instruction tuning.

Layers	Instruct	General	p-value
1-4	$78.46_{\pm 23.91}$	$52.45_{\pm 35.30}$	0.0005
5-8	$66.89_{\pm 23.82}$	$48.42_{\pm 32.71}$	0.0045
9-12	$66.23_{\pm 28.85}$	$51.15_{\pm 32.88}$	0.0420
13-16	$64.39_{\pm 25.85}$	$50.11_{\pm 32.06}$	0.0285
17-20	$58.25_{\pm 40.52}$	$53.50_{\pm 31.61}$	0.6181
21-24	$51.03_{\pm 31.58}$	$51.46_{\pm 31.57}$	0.9531
25-28	$39.19_{\pm 31.37}$	$52.32_{\pm 32.22}$	0.0960
29-32	$50.49_{\pm 36.01}$	$50.58_{\pm 31.65}$	0.9919

Specifically, We identify 35 instruction verbs (e.g., "write", "create", and "classify") based on (Wang et al., 2022; Ouyang et al., 2022), and also assemble a control set of 1000 frequent verbs (Speer, 2022). For each verb, we calculate what proportion of the self-attention head encodes more word pairs about that verb after instruction tuning. We only consider those self-attention heads that change in the number of word pairs for the verb after instruction tuning and report the results on Table 3.

Obs-4: Instruction tuning significantly modifies self-attention heads. Figure 2 shows that as layer depth increases, the differences between word pair lists become more significant. Notably, we could observe that over 35% of word pairs for the same self-attention head are changed in the last few layers. This result not only illustrates the significant impact of instruction tuning on the self-attention heads, but also demonstrates that the proposed method can capture the diverse word-word relationships encoded by the self-attention layer.

Obs-5: Enhanced Encoding of Instruction Verbs through Instruction Tuning in Lower Self-Attention Heads. Table 3 demonstrates that instruction tuning notably increases the propensity of self-attention heads, particularly in lower (1-8) and middle (9-20) layers, to encode word-word patterns associated with instruction verbs. This enhancement is statistically significant ($p < 0.05$) within the first 16 layers. In contrast, approximately 50% of self-attention heads exhibit a similar tendency for general verbs, while 50% refers to a neutral impact, signifying neither an increase nor decrease in word relations for the given verbs. This difference indicates that instruction tuning teaches self-attention to identify various detailed instructions.

5.2 Analyzing Feed-forward Networks

Method. We aim to interpret the knowledge of feed-forward networks in the *concept* level. We treat each feed-forward network $\sigma(\mathbf{X}\mathbf{W}_u^\top)\mathbf{W}_p$ as key-value memories (Geva et al., 2020), where each row vector of $\mathbf{W}_u$ and $\mathbf{W}_p$ stores a textual pattern. However, these textual patterns (neurons) are usually polysemantic (Elhage et al., 2022; Bricken et al., 2023a), causing each textual pattern not to be interpreted within a concise meaning (Geva et al., 2021). Thus, we propose to seek a set of orthogonal vectors that capture the major directions in which these patterns spread. Formally, given patterns $\mathbf{W}_p$, we construct the covariance matrix as $\mathbf{C} = \widetilde{\mathbf{W}}_p^\top \widetilde{\mathbf{W}}_p$, where $\widetilde{\mathbf{W}}_p$ is the centralized matrix of $\mathbf{W}_p$ with zero-mean columns. Then the orthogonal basis vectors $\mathbf{V}$ of these patterns satisfy:

$$\mathbf{C}\mathbf{V} = \mathbf{\Lambda}\mathbf{V}, \tag{2}$$

where each column vector of $\mathbf{V} \in \mathbb{R}^{D\times D}$ is unit length, $\mathbf{\Lambda} = diag([\lambda_1, ..., \lambda_D])$, and $\lambda_1 \geq ... \geq \lambda_D \geq 0$. In this context, our primary focus lies on the top-R values of $\mathbf{\Lambda}$ along with their corresponding column vectors in $\mathbf{V}$. This is due to the fact that they show the *principal directions* of the encoded patterns from $\mathbf{W}_p$. We then project each principal vector to the word embedding space $\mathbf{E}_o$ and find the top-K relevant words for interpretation: $\mathcal{E}_r = \arg max_{\mathcal{V}'\in\mathcal{V},|\mathcal{V}'|=K} \sum_{w\in\mathcal{V}'} \mathbf{V}^\top[r]\mathbf{E}_o[w]$, where $\mathbf{V}^\top[r]$ is the r-th column vector of $\mathbf{V}$, $\mathbf{E}_o[w]$ is the output word embedding of w. Since $\mathbf{v}_r$ is a unit vector, $\mathbf{V}^\top[r]\mathbf{E}_o[w]$ measures the projection length of the word vector in this direction. Thus, it is natural to represent this vector with the words having the largest projection length, and the word list could be further summarized as a textual description by a human or a machine annotator.

Settings. We create a new vocabulary derived from ShareGPT (RyokoAI, 2023) to make the candidate words $\mathcal{V}$ more understandable compared to a large number of sub-tokens from the built-in LLaMA vocabulary. We then analyze the first 300 basis vectors of each feed-forward network from LLaMA and Vicuna with their top 15 relevant words. ChatGPT [5] is considered our machine annotator for this experiment. Table 4 provides sample word lists and their descriptions. More cases are available in Appendix E.2. The detailed settings and statistics of concept descriptions are shown in Appendix D.2. We discuss the results of the principal components in Appendix E.1.

Table 4: Interpreting the last feed-forward network of Vicuna with the proposed decomposition method.

Description	Words
medical abbreviation	CBT, RTK, RT, RH, HRV, MT, ...
starting with "the"	the, theological, theology, ...
hyphenated terms	one-of-a-kind, state-of-the-art, ...
numbers	sha256, tt, 8266, 768, 1986, ...

To study the evolution of pre-trained knowledge, we condense tasks from previous research (Zheng et al., 2023; Ouyang et al., 2022) to scenarios including writing, math, coding, and translation. We then identify which scenarios a concept could be used for (see Appendix D). Note that some concepts may fit multiple scenarios. Also, we sort concepts into phonology [6], morphology [7], syntax, or semantics linguistic levels based on the disciplines in the linguistic subject (Thomas, 2005). Table 5 displays the percentage of knowledge for different scenarios and linguistic levels.

Obs-6: The principal vectors of the weights of feed-forward networks provide concept-level understandings of the encoded knowledge. We select four representative principal components and their explanations from the last feed-forward network of Vicuna and display them in Table 4. More cases are available in Tables 10 and 11. In Table 4, the descriptions of the four principal vectors span diverse topics, ranging from medical ("medical abbreviation") to linguistic ("starting with *the*"). Notably, the concept of medical abbreviations stands out, as it's often difficult for human annotators to discern their medical relevance. This indicates the advantage of utilizing machine annotators for their vast knowledge. Coincidentally, Appendix D.2 shows that around 60% of the first 300 principal components from the middle layers of Vicuna could be interpreted by ChatGPT. This evidence empirically verifies the rationale for analyzing feed-forward networks with the proposed method.

[5] We employ ChatGPT-turbo-3.5-0613 in this work.

[6] Phonology studies sound systems, e.g. words with *"le" sound*: brittle, tackle, chuckle, pickle.

[7] Morphology studies word structure, e.g. words with *"sub-" prefix*: subarray, subculture, subway.

Table 5: Concept distribution over different user-oriented scenarios and linguistic levels.

	Category	Vicuna	LLaMA	p-value
Scenarios	Writing	$53.50_{\pm.46}$	$51.47_{\pm.92}$	0.0154
	Coding	$29.45_{\pm.43}$	$28.64_{\pm.48}$	0.0350
	Math	$5.21_{\pm.36}$	$5.04_{\pm.33}$	0.5193
	Translation	$25.30_{\pm.39}$	$26.27_{\pm.70}$	0.0411
Linguistic	Phonology	$1.18_{\pm.11}$	$1.15_{\pm.07}$	0.6251
	Morphology	$17.16_{\pm.49}$	$16.83_{\pm.60}$	0.4223
	Syntax	$7.16_{\pm.31}$	$7.52_{\pm.50}$	0.2551
	Semantic	$74.70_{\pm.65}$	$74.66_{\pm.67}$	0.9394

Obs-7: Instruction tuning shifts the principal vectors of feed-forward networks toward user-oriented tasks without moving them across linguistic levels. We observe from Table 5 that Vicuna encodes more concepts than LLaMA for writing, coding, and math tasks, with the difference in writing and coding being statistic significant ($p < 0.05$), where the null-hypothesis is knowledge proportions of a certain category for Vicuna and LLaMA are equal. However, that of concepts for translation is reduced after fine-tuning, indicating that multi-linguistic knowledge is forgotten. Although we could observe the changes over the user view, from the linguistic view, it remains the same. In particular, Vicuna and LLaMA show nearly identical distributions across the four linguistic levels. None of them are statistically significant ($p > 0.05$). This observation suggests instruction tuning does not alter the distribution of pre-trained knowledge across linguistic levels.

Obs-8: The proportion of semantic knowledge first increases then decreases from bottom to top layers, while that of morphology knowledge does the opposite. Figure 3 displays how concepts from various linguistic levels are spread across layers. First, there isn't a noticeable distribution shift between Vicuna and LLaMA, which matches Obs-7. One noteworthy observation is the opposite "U"-shape trend of semantic knowledge, mirrored by a regular "U"-shape of morphology. This pattern is surprising, especially since previous studies in computer vision suggest that basic features are extracted in the bottom layers, and compositional knowledge is learned in the top layers (Zeiler and Fergus, 2014; Selvaraju et al., 2016). However, since LLaMA is a generative model, this unusual pattern makes some sense. Specifically, we conjecture that LLaMA learns more morphology knowledge (e.g., prefix and suffix patterns) in the last few layers to simulate a prefix-tree structure (Fredkin, 1960; Giancarlo, 1995; Paladhi and Bandyopadhyay, 2008; Shan et al., 2012). By doing so,

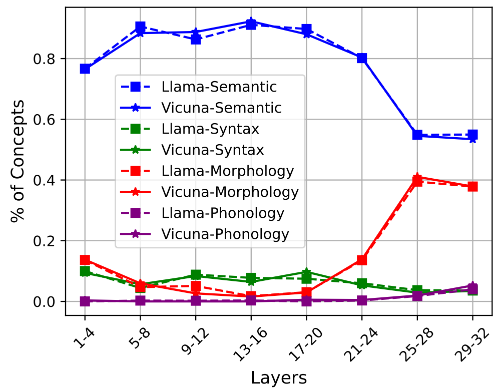


Figure 3: Distribution of concepts at linguistic levels over different model layers.

LLaMA could use fewer parameters to memorize more phrases to complete the next-word prediction task. We leave explorations as future work.

6 Discussion

Our findings provide a unique perspective to align with recent studies. 1) The importance of *prompt diversity* is highlighted by both us and (Zhou et al., 2023; Wang et al., 2022). Since our three findings suggest that instruction tuning links the pre-trained model to user tasks, we could expect a better alignment with human intentions if the model is exposed to broader prompts. 2) The efficacy of *training self-attention with first priority* (LoRA fine-tuning) (Taori et al., 2023; Juletx, 2023) is corroborated by Finding-1 and Finding-2. Specifically, Finding-1 illustrates the capability to distinguish instruction words is essential to the instruction following, while Finding-2 highlights that self-attention heads directly learn instruction relations. 3) The advantage of *training feed-forward networks* (fully fine-tuning) (Sun et al., 2023) is evident from Finding-2 and Finding-3, which demonstrate that feed-forward networks update their knowledge toward user tasks.

7 Conclusion

This paper presents an inherently comprehensive analysis of instruction tuning for user intention alignment by quantitatively and qualitatively comparing the interpretations between pre-trained and fine-tuned models. Our findings indicate that instruction tuning links the pre-trained model to user intentions, including encoding more instruction words' knowledge within self-attention, and rotating general knowledge from feed-forward networks towards user usage. It is worth mentioning that the interpretability toolbox used in this study can also support future general research on LLMs.

8 Limitations

This study aims to investigate the impact of instruction tuning on pre-trained language models in terms of human alignment. A primary constraint of this work is that the introduced explanation toolbox is developed on the availability of model weights and gradients, indicating a white-box approach. Consequently, these tools may not be fully effective for analyzing black-box instruction-tuned models, like ChatGPT (Bai et al., 2022) and Claude (Anthropic, 2023). We will seek to enhance our toolbox by incorporating methods suitable for black-box model analysis in the future. On the other hand, another key technology related to human alignment for LLMs is Reinforcement Learning with Human Feedback (RLHF) (Stiennon et al., 2020; Bai et al., 2022), which is another aspect not touched on in this article. We encourage researchers to apply our toolbox to study RLHF-tuned models and explore the different roles of instruction tuning and RLHF for human alignment.

9 Ethical Impact

This research employs the pre-trained models LLaMA (Touvron et al., 2023) and its variant Vicuna (Zheng et al., 2023), under their respective academic-use licenses. The utilization of these models adheres to their specific terms, focusing exclusively on scholarly purposes. Additionally, our study incorporates four datasets: Self-Instruct (Wang et al., 2022), LIMA (Zhou et al., 2023), MT-Bench (Zheng et al., 2023), and ShareGPT (RyokoAI, 2023), each under its own usage conditions. These conditions include compliance with privacy and data protection standards. In presenting our findings, we have rigorously ensured that no personal identifiers are disclosed and that the content remains free from offensive material, aligning with ethical research practices.

References

Anthropic. 2023. *Model Card and Evaluations for Claude Models*.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Oren Barkan, Edan Hauon, Avi Caciularu, Ori Katz, Itzik Malkiel, Omri Armstrong, and Noam Koenigstein. 2021. Grad-sam: Explaining transformers via gradient self-attention maps. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2882–2887.

Yonatan Belinkov, Lluís Màrquez, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Evaluating layers of representation in neural machine translation on part-of-speech and semantic tagging tasks. *arXiv preprint arXiv:1801.07772*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. 2023a. *Decomposing Language Models With Dictionary Learning*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, et al. 2023b. Towards monosemanticity: Decomposing language models with dictionary learning. transformer circuits thread, 2023.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2022. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. *arXiv preprint arXiv:1804.07781*.

Edward Fredkin. 1960. Trie memory. *Communications of the ACM*, 3(9):490–499.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

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.

Raffaele Giancarlo. 1995. A generalization of the suffix tree to square matrices, with applications. *SIAM Journal on Computing*, 24(3):520–562.

Jing Huang, Atticus Geiger, Karel D'Oosterlinck, Zhengxuan Wu, and Christopher Potts. 2023. Rigorously assessing natural language explanations of neurons. *arXiv preprint arXiv:2309.10312*.

Niall Hurley and Scott Rickard. 2009. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Juletx. 2023. Alpaca-lora. `https://github.com/tloen/alpaca-lora`.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pages 2668–2677. PMLR.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. 2023. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*.

Enja Kokalj, Blaž Škrlj, Nada Lavrač, Senja Pollak, and Marko Robnik-Šikonja. 2021. Bert meets shapley: Extending shap explanations to transformer-based classifiers. In *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, pages 16–21.

Po-Nien Kung and Nanyun Peng. 2023. Do models really learn to follow instructions? an empirical study of instruction tuning. *arXiv preprint arXiv:2305.11383*.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066*.

Zongxia Li, Paiheng Xu, Fuxiao Liu, and Hyemi Song. 2023. Towards understanding in-context learning with contrastive demonstrations and saliency maps. *arXiv preprint arXiv:2307.05052*.

Shihao Liang, Kunlun Zhu, Runchu Tian, Yujia Qin, Huadong Wang, Xin Cong, Zhiyuan Liu, Xiaojiang Liu, and Maosong Sun. 2023. Exploring format consistency for instruction tuning. *arXiv preprint arXiv:2307.15504*.

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 preprint arXiv:2307.03172*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Beren Millidge and Sid Black. 2022. The singular value decompositions of transformer weight matrices are highly interpretable. *https://www.alignmentforum.org/*.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163.

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*.

R OpenAI. 2023. Gpt-4 technical report. *arXiv*, pages 2303–08774.

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.

Sibabrata Paladhi and Sivaji Bandyopadhyay. 2008. Generation of referring expression using prefix tree structure. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Ofir Press, Noah A Smith, and Omer Levy. 2019. Improving transformer models by reordering their sublayers. *arXiv preprint arXiv:1911.03864*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

RyokoAI. 2023. Sharegpt52k. *Huggingface Datasets*.

Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. 2022. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2016. Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Y Shan, X Chen, Y Shi, and J Liu. 2012. Fast language model look-ahead algorithm using extended n-gram model. *Acta Automatica Sinica*, 38(10):1618–1626.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Bills Steven, Cammarata Nick, Mossing Dan, Tillman Henk, Gao Leo, Goh Gabriel, Sutskever Ilya, Leike Jan, Wu Jeff, and Saunders William. 2022. Language models can explain neurons in language models. *https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. 2019. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*.

Xianghui Sun, Yunjie Ji, Baochang Ma, and Xiangang Li. 2023. A comparative study between full-parameter and lora-based fine-tuning on chinese instruction data for instruction following large language model. *arXiv preprint arXiv:2304.08109*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7.

James J Thomas. 2005. *Illuminating the path:[the research and development agenda for visual analytics]*. IEEE Computer Society.

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*.

Neeraj Varshney, Wenlin Yao, Hongming Zhang, Jianshu Chen, and Dong Yu. 2023. A stitch in time saves nine: Detecting and mitigating hallucinations of llms by validating low-confidence generation. *arXiv preprint arXiv:2307.03987*.

Jesse Vig. 2019. Bertviz: A tool for visualizing multihead self-attention in the bert model. In *ICLR workshop: Debugging machine learning models*, volume 23.

Elena Voita, Javier Ferrando, and Christoforos Nalmpantis. 2023. Neurons in large language models: Dead, n-gram, positional. *arXiv preprint arXiv:2309.04827*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. 2021. Polyjuice: Generating counterfactuals for explaining, evaluating, and improving models. *arXiv preprint arXiv:2101.00288*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*, pages 818–833. Springer.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2023. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*.

A Proof of Linearly Approximation to Importance Scores

We prove that equation $I_{n,m} = p(y_m|Z_m) - p(y_m|Z_{m,/n}) \approx \frac{\partial f(y_m|z_m)}{\partial \mathbf{E}_i[x_n]} \cdot \mathbf{E}_i[x_n]^\top$ with the first-order Taylor extension. $p(y_m|Z_m)$ is written as $f(y_m|\mathbf{Z}_m)$, where f is the language model, $\mathbf{Z}_m \in \mathbb{R}^{(N+m-1)\times d}$ are the word embeddings of the input token sequence $Z_m = [x_1, ..., x_N, y_1, ..., y_{m-1}]$, and the d-dimensional word embeddings of a token $w \in Z_m$ is defined as $\mathbf{E}_i[w]$. Thus, we first have $I_{n,m} = f(y_m|\mathbf{Z}_m) - f(y_m|\mathbf{Z}_{m,/n})$, where we let the n-th row vector of $\mathbf{Z}_{m,/n}$ be zeros.

The first-order Taylor expansion of $f(y_m|\mathbf{Z}_m)$ around $\mathbf{Z}_{m,/n}$ is

$$f(y_m|\mathbf{Z}_m) \approx f(y_m|\mathbf{Z}_{m,/n}) + \left.\frac{\partial f(y_m|\mathbf{Z}_m)}{\partial \mathbf{Z}_m}\right|_{\mathbf{Z}_{m,/n}} \cdot (\mathbf{Z}_m - \mathbf{Z}_{m,/n})^\top.$$

Since the difference between $\mathbf{Z}_{m,/n}$ and $\mathbf{Z}_m$ is the n-th row, the term $\mathbf{Z}_m - \mathbf{Z}_{m,/n}$ is just the vector $\mathbf{E}_i[x_n]$. Therefore, the above equation could be simplified as:

$$f(y_m|\mathbf{Z}_m) \approx f(y_m|\mathbf{Z}_{m,/n}) + \frac{\partial f(y_m|\mathbf{Z}_m)}{\partial \mathbf{E}_i[x_n]} \cdot \mathbf{E}_i[x_n]^\top$$

Bring this approximation to the definition of $I_{n,m}$, we have $I_{n,m} \approx \frac{\partial f(y_m|\mathbf{Z}_m)}{\partial \mathbf{E}_i[x_n]} \cdot \mathbf{E}_i[x_n]^\top$.

B Analyzing Importance Density

B.1 Experiment Settings

For each collected prompting text from the three public datasets, we let Vicuna and LLaMA generate its corresponding response (Sec. 3.2); we then manually identify the instruction sentences from each input prompt and annotate whether the response provides helpful information ("followed") or not ("unfollowed"). Regarding computational efficiency, generating the importance density for a single instance necessitated approximately 100 seconds, utilizing dual Nvidia A6000 GPUs.

Annotate instruction and context. Specifically, the instruction usually describes the user intention with some background (optional), which could be both very long [8] or very concise [9]. Note that we annotate the instruction words on the sentence level, and the template words as "Input:" and "Output:" are not considered. For some prompts, the instruction words may be distributed in both the head and tail of the input text, and we will consider them together. Among these instruction sentences, we define the rest of the input prompt as context words, which is unnecessary to the input prompting text.

Annotate Followed or Unfollowed Response We consider the *helpfulness* of the response as the ability of instruction following described by (Ouyang et al., 2022). Therefore, if a response is helpful to the user, then we label it with "followed". Specifically, we consider four levels of helpfulness: L1 - the model is randomly saying something or just repeating itself; L2 - the model provides some information that could be used to answer the question, but the model fails to organize it well; L3 - the model generates a response that generally follows the prompts, but missing some detailed instructions; L4 - the response is perfect as a human response. In our study, we consider the responses from L2 to L4 as "followed". Note that we are not concerned about hallucination issues in our study.

B.2 Case Study on Outliers

Instruction fine-tuned models may pretend to follow the instructions. Figure 4 visualizes a salient map of an instance related to writing enhancement (please see the caption for details). Vicuna's response addresses grammatical errors and modifies sentence structures for improved clarity. A key observation from the figure is that only the first three instruction tokens guide the response generation (Red Box). Specifically, the first three words are "The sentence you", which seems to be not the key instruction verbs like "Rewrite" from the second sentence. Also, some words from the context part are acted as instruction words (Blue Box), which are "\nInput:" and "\nOutput:" from the prompt template. These are the words that should be considered as the instruction words since they do not provide the user's intentions. Additionally, a distinctive diagonal line spans the context section, hinting at the model's predisposition to echo context from preceding content. This figure suggests that the model is leveraging inherent language modeling ability rather than identifying and acting upon the given instructions. Thus, we point out that assessing instruction-following abilities based solely on the correlation between input prompts and output responses might not provide an accurate reflection of the model's internal behaviors, while

[8] A long instruction: "How do social media platforms influence the way people consume and share news, and what are the potential implications for the spread of misinformation?"

[9] A short instruction: "to English:"

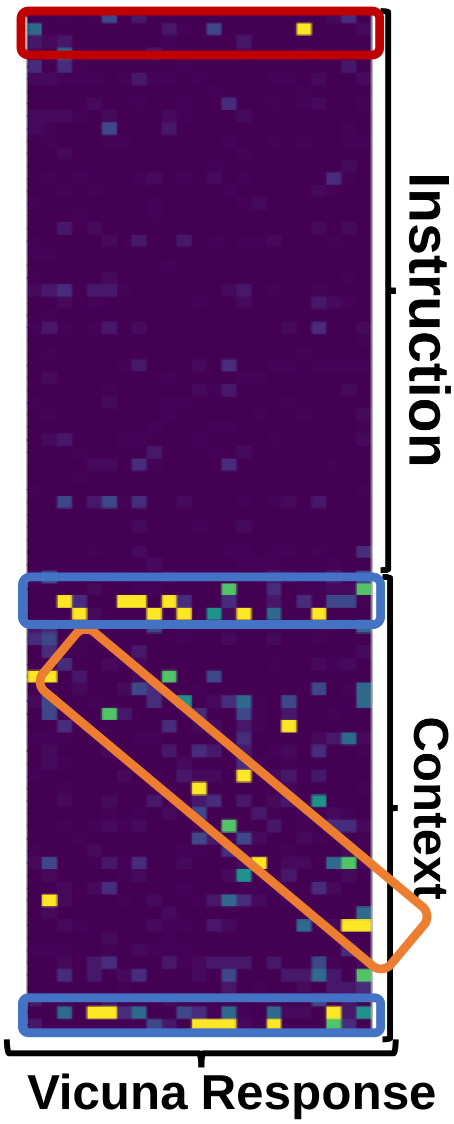


Figure 4: **CaseID:** Self-Instruct-1 **Prompt:** The sentence you are given might be too wordy, complicated, or unclear. Rewrite the sentence and make your writing clearer by keeping it concise. Whenever possible, break complex sentences into multiple sentences and eliminate unnecessary words.\n\nInput: If you have any questions about my rate or if you find it necessary to increase or decrease the scope for this project, please let me know.\n\nOutput: **Vicuna:** Do you have any questions about my rate or do you need to adjust the project scope? Please let me know.

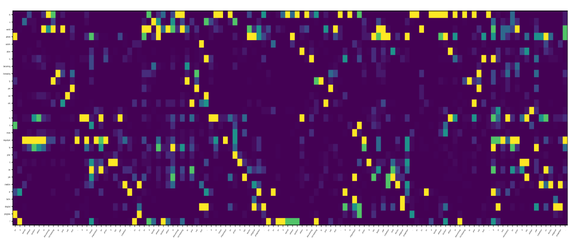


Figure 5: **CaseID:** LIMA-144 **Prompt:** In a world where automation is becoming increasingly prevalent, is it more important to prioritize job creation or technological progress? **Vicuna:** In a world where automation is becoming increasingly prevalent, it is more important to prioritize job creation.\n\nIn a world where automation is becoming increasingly prevalent, it is more important to prioritize technological progress.\n\nIn a world where automation is becoming increasingly prevalent, it is important to prioritize both job creation and technological progress.\n\nIn a world where automation is becoming increasingly prevalent, it is not necessary to prioritize either job creation or technological progress.

it is still a common strategy to develop the Reward model for RLHF process (Ouyang et al., 2022).

However, we have identified certain instances where our importance density fails. This is predominantly due to our density function's lack of positional awareness. For instance, in Figure 5, the entire user input comprises instruction words. The map suggests that these words play a crucial role in guiding the generation, even towards the latter part of the responses. Under our hypothesis, it would appear the model is following user instructions. Yet, Vicuna seems to merely reiterate the input prompt repetitively, resulting in recurring diagonal patterns. We recommend future research to address this shortcoming, either by adopting a density function that's positionally aware or by integrating a step to identify and handle repetitive responses early on.

B.3 Exploring Prompt Position with Importance Density

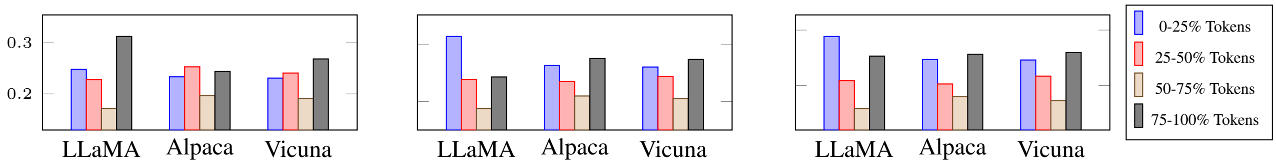
Settings. Each input prompting text from our datasets is divided into individual sentences, with each sentence further split into four same-length segments. We normalize the density scores for a sentence by dividing by their sum and then accumulating them for each segment. The averaged attribution proportions for each segment within the input sentences are depicted in Figure 6.

Results. Figure 6 shows the importance density distributed on different segments of input sentences. Both pre-trained and tuned models reveal a notable "U"-shape across all datasets. This is also known as "lost in the middle" (Liu et al., 2023), where they show that SOTA models can overlook central inputs. Unlike their focus on a single task, our analysis is grounded on our importance density score on diverse prompting texts, suggesting that this issue commonly and intrinsically exists. When comparing pre-trained to fine-tuned models, we spot a sharper "U" in the former, which becomes less obvious after instruction tuning.

C Visualizing Salient Maps

C.1 Experiment Settings

Contrary to the examples shown in the primary content, which utilize golden responses, our focus here is on the connections between user inputs and model outputs. To achieve this, we generate responses from LLaMA and Vicuna, following the



(a) Self-Instruct (b) Lima (c) MT-Bench

Figure 6: Distribution of importance density over different parts of prompt tokens.

protocol laid out in Sec.3.2. Subsequently, we derive the salient maps as per the technique introduced in Sec.4.1.

To ensure the maps provide an accurate depiction of the generation process, we set $L = 10$ and $b = 0$. Each map's vertical axis denotes the prompting texts, whereas the horizontal axis symbolizes the generated responses. The intensity of each data point corresponds to the association strength between the respective input and output tokens, with brighter points indicating stronger relationships (visualizing with the best colors).

C.2 Experiment Results

Figure 9-14 validate our qualitative assessment that instruction words in user inputs are critical in guiding the generation process. It's evident that each context word typically has limited influence on the response. Collectively, these salient maps underscore the validity of input attribution, achieved by gauging the density of the sparse and normalized importance scores.

D Scaling up with Automated Tools

We build upon recent advancements in automated interpretation, using cutting-edge large language models (Taori et al., 2023; Peng et al., 2023; Steven et al., 2022) to emulate human annotators in generating high-level interpretations. By leveraging machine annotators, we could easily scale up our methods to analysis the entire model, providing a more solid results to our findings.

D.1 Experiment Settings

Generating Configuration. We employ ChatGPT [10] as our machine annotator. Our experiments utilize the gpt-3.5-turbo-0613 model with a hyper-parameter top-p=0.9 for nuclear sampling. To mitigate the variability in language model outputs, we repeat the experiment five times. In each iteration, we first condense the top-K words of a specific basis vector into a distinct concept, then pinpoint the user-oriented tasks and linguistic levels associated with these concepts. For our initial interaction with ChatGPT, the temperature is set to 0—signifying a greedy search strategy. In subsequent interactions, we set the temperature to 1. Nevertheless, when identifying tasks and levels, we consistently maintain the temperature at 0.0.

Prompt Design. Effective automated interpretation hinges on well-crafted prompts. We meticulously design these prompts using three strategies: role-play, in-context conversational examples, and exclusively high-quality examples.

Template-1: Describing words with concise concepts. The top-15 most activated words coming from the method presented in Sec. 5.2 will be directly appended to this template.

```
System: You are a neuron interpreter for
neural networks. Each neuron looks for
one particular concept/topic/theme/beha
vior/pattern. Look at some words the
neuron activates for and summarize in a
single concept/topic/theme/behavior/pat
tern what the neuron is looking for.
Don't list examples of words and keep
your summary as concise as possible.
If you cannot summarize more than half
of the given words within one clear
concept/topic/theme/behavior/pattern,
you should say 'Cannot Tell'.

User: Words: January, terday, cember,
April, July, September, December,
Thursday, quished, November, Tuesday.
Agent: dates.

User: Words: B., M., e., R., C., OK., A.,
H., D., S., J., al., p., T., N.,
W., G., a.C., or, St., K., a.m., L..
Agent: abbreviations and acronyms.

User: Words: actual, literal, real, Real,
optical, Physical, REAL, virtual, visual.
Agent: perception of reality.

User: Words: Go, Python, C++, Java, c#,
python3, cuda, java, javascript, basic.
Agent: programing languages.

User: Words: 1950, 1980, 1985, 1958, 1850
, 1980, 1960, 1940, 1984, 1948.
Agent: years.

User: Words:
```

Template-2: Identifying applicable user-oriented

[10] https://platform.openai.com/docs/guides/gpt

tasks. Summarized concepts are concatenated to this template. We check the writing task into three tasks because ChatGPT often deems nearly every concept suitable for writing. We regard any of these detailed tasks as the primary purpose of writing.

```
System: Which of the following assistant
tasks can the given concept is used for?
\n\nTasks: daily writing, literary writ
ing, professional writing, solving math
problems, coding, translation. Return
'None' if it cannot be used for any of
the above tasks. If it could be used for
multiple tasks, list all of them and
seperate with ';'.

User: Concept: Words are social media
post tags.
Agent: daily writing

User: Concept: Words are Latex code for
drawing a grouped barchart.
Agent: professional writing

User: Concept: Words are foreign words
or names.
Agent: translation

User: Concept: Words are URLs.
Agent: None

User: Concept: Words are Words related
to configuration files and
web addresses.
Agent: coding

User: Concept: Words are rhyming words.
Agent: literary writing

User: Concept: Words are programming
commands and terms.
Agent: coding

User: Concept: Words are
```

Template-3: Identifying linguistic level. Any automated summarized concept will be directly concatenated to this template.

```
System: You are a linguist. Classify
the provided concept into
one of the following categories:
Phonology, Morphology, Syntax,
and Semantic.

User: Concept: Words are dates.
Agent: semantic

User: Concept: Words are perception
of reality.
Agent: Semantic

User: Concept: Words are abbrevations
and acronyms.
Agent: Morphology

User: Concept: Words are related to
actions or activities.
Agent: Syntax

User: Concept: Words are medical
abbrivations.
Agent: Semantic

User: Concept: Words are URLs.
Agent: Morphology

User: Concept: Words are verbs.
Agent: Syntax

User: Concept: Words are adjective.
Agent: Syntax

User: Concept: Words are rhyming words.
Agent: Phonology

User: Concept: Words are programming
languages.
Agent: Semantic

User: Concept: Words are
```

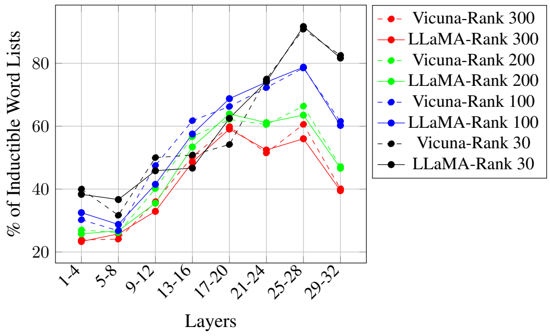


Figure 7: % of represented word lists from top-ranked basis vectors with a concise description.

D.2 Experiment Results

Figure 7 illustrates the proportion of word lists that can be induced to a concise concept by our machine annotator. According to our template, if "Cannot Tell" exists in the word list descriptions, we consider that this concept has failed to be interpreted. We have observed that the Vicuna and LLaMA models display comparable levels of interpretability, with no significant distinctions between them. A noticeable trend emerges as the number of layers increases: the ability to explain their encoded concepts improves. Specifically, within layers 24-28, the average interpretability rate for the first 30 concepts peaks at 91.67%. This high interpretability rate underscores the effectiveness of our proposed method. It can aptly convey in clear, concise text the knowledge encoded by these models. However, there's a caveat: knowledge encoded closer to the output layer, specifically between layers 28-32, becomes more challenging to elucidate. Interestingly, this particular challenge wasn't present when applying automated interpretation tools to GPT-2 (Millidge and Black, 2022), indicating the behaviors between small and large models are different. Additionally, our findings indicate a decreasing trend in interpretability for concepts that are ranked further back. Overall, these results validate the efficacy of our proposed method in analyzing the knowledge encoded within models.

Table 6-9 enumerates the words that experienced the most significant changes in frequency after instruction tuning, we also show the change of rank following. These words are meaningful words (at least four characters and not a stopword) extracted from the concept descriptions generated by our machine annotator. From the tables, certain words, notably "language", "programming", and "process", displayed significant shifts in frequency after instruction tuning. Linguistic terms ("Spanish", "translation") and technical terms ("method", "programming" and "software") exhibited noticeable changes in various layers. Interestingly, "language" consistently surfaced in almost every layer group, with its frequency both rising and dropping. This observation indicates that different layers are responsible for encoding different categories of knowledge. Specifically, the bottom layers are responsible for storing more basic knowledge ("behavior", "operation", "adjective"), the middle layers are responsible for learning more abstract knowledge ("functions/methods", "programming", "software development"), and the higher layers are responsible for learning more knowledge for efficient text generation ("start with", "rhyming", "sound", "letter",). Broadly, the increased mention of words pertinent to user scenarios after fine-tuning underscores the model's refined focus on user-centric tasks and applications.

E Interpreting Feed-Forward Networks

E.1 Details of the PCA Results

Figure 8 displays the averaging accumulated explained variance of decomposed principal components across the 32 layers, where the translucent area indicates their standard deviations. Since LLaMA and vicuna show almost exactly the same line, we omit LLaMA from this figure. From the figure, we have several observations. Firstly, we find that the accumulated explained variance increases smoothly, where almost half of the basis vectors could explain around 80% of the variances. This observation demonstrates that these neurons do not focus on expressing a few certain features, emphasizing the diversity of the learned hidden features. In addition, the black arrow points out that the accumulated explained variance of the 300 basis vector is about 22.49%, where 300 is the number of basis vectors we studied in this research. It validates that the top 300 parameters are expected to be interpretable since their accumulated explained variance is only 22.49%.

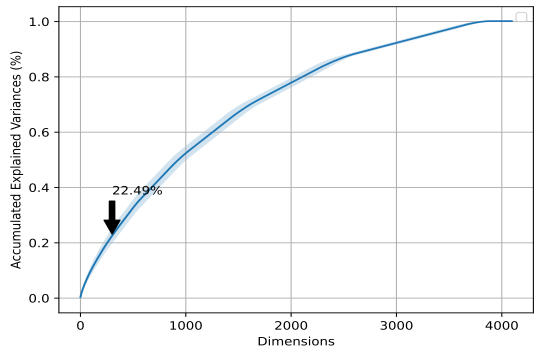


Figure 8: Accumulated explained variance of feed-forward networks from Vicuna.

E.2 Qualitative Analysis to Interpretability of Principal Components

Table 10 and Table 11 list cases that are well interpreted by ChatGPT-turbo-3.5-0613. From these cases, we found that the concept descriptions generally reflect what is behind the word lists well.

F Interpreting Self-Attention Heads

Table 12 and Table 13 list more word pairs for the self-attention heads from the first and the last layers. Typically, these cases are evidence that the extracted word pairs show some insightful relations when we read each one individually. However, when we read them together, it cannot reflect such a concise concept as the feed-forward networks.

Instruction tuning may distill the behaviors of neurons. For example, neuron-pair $(Layer = 31, Head = 24, Dim = 62)$ capture relations in computers (such as backend=authentication, icon=keyboard, giant=cardboard, GPU=PS, git=curl, and so on). After instruction tuning, the model finds more computer-related word pairs (GPU=motherboard, VM=motherboard, tab=keyboard, mongo=staat, mongo=orden) and overlooks some un-related word pairs (dense=bright, convinced=confused), though the new relations may be not valid. This case is also evidence that the instruction tuning does not make a significant change in the pre-trained knowledge across concepts.

Table 6: Frequency [rank] shift of words from concept description after instruction tuning.

Layers 1-4		Layers 5-8	
Frequency↑	**Frequency↓**	**Frequency↑**	**Frequency↓**
language[3]	quality[-3]	programming[0]	foreign-language[0]
behavior[83]	describing[-2]	describing[38]	technology[-19]
English[79]	characteristic[-1]	computer[11]	Spanish[-33]
process[4]	communication[-22]	operation[11]	technical[-32]
software-development[8]	something[-18]	computer-science[66]	multilingual[-8]
multilingual[64]	start[-43]	development[53]	something[-8]
analysis[67]	adjective[1]	language[0]	process[0]
operation[33]	foreign-language[-1]	syntax[17]	characteristic[-1]
attribute[5]	various[-12]	manipulation[14]	variation[-9]
Spanish[14]	concepts/functions[-19]	terminology[22]	functions/methods[-7]

Table 7: Frequency [rank] shift of words from concept description after instruction tuning. (continued)

Layers 9-12		Layers 13-16	
Frequency↑	**Frequency↓**	**Frequency↑**	**Frequency↓**
method[89]	translation[0]	programming[0]	process[-1]
french[13]	operation[-31]	software-development[8]	expression[-45]
understand[34]	software-development[-17]	language-proficiency[10]	syntax[-5]
communication[10]	process[0]	concepts/keys[29]	variation[-15]
concepts/functions[41]	foreign-language[0]	terminology[119]	language-related[-24]
language-agnostic[23]	programming[0]	language-independent[52]	ambiguity[-49]
German[31]	concepts/methods/functions[-61]	concepts/functions[16]	handling[-32]
comparison[50]	multilingual[-5]	French[51]	language[0]
variety[35]	property[-75]	communication[4]	cultural[-93]
technology[28]	language[0]	localization[96]	attribute[-14]

Table 8: Frequency [rank] shift of words from concept description after instruction tuning. (continued)

Layers 17-20		Layers 21-24	
Frequency↑	**Frequency↓**	**Frequency↑**	**Frequency↓**
programming[0]	foreign-language[-2]	manipulation[50]	programming[-2]
language[1]	translation[-1]	adjective[9]	state[-12]
syntax[78]	variation[-20]	specific[81]	translation[-5]
process[2]	expression[-15]	object[42]	quality[-7]
language-related[-24]	interaction[24]	adjective[-27]	value[48]
time-related[14]	feature[-30]	location[48]	difficulty[-77]
language-proficiency[5]	characteristic[-5]	variation[9]	action[0]
terminology[123]	duration[-33]	language[1]	prefix[-1]
technology[121]	choice[-135]	relationship[121]	start[1]
programming-language[-70]	quality	personal[72]	activity[-2]

Table 9: Frequency [rank] shift of words from concept description after instruction tuning. (continued)

Layers 25-28		Layers 29-32	
Frequency↑	**Frequency↓**	**Frequency↑**	**Frequency↓**
language[4]	start with[0]	start with [0]	foreign-language[-10]
interaction[117]	sound[-1]	sound[4]	language[-3]
combination[2]	programming[-1]	rhyming[15]	suffix[-4]
variation[1]	action[-2]	combination[9]	abbreviation[-2]
software	number[0]	letter[0]	numerical[-5]
event[66]	alphanumeric[-23]	process[8]	abbreviations/acronyms[-8]
manipulating[53]	abbreviations/acronyms[0]	French[7]	Spanish[-7]
operation[28]	pattern[-3]	number[1]	programming[-2]
measurement[60]	suffix[-45]	similarity[53]	Indonesian[-18]
spell[55]	string[-56]	measurement[43]	sequence[-34]

Table 10: Representing words, application scenarios, and linguistic level of the concepts encoded by the 32ed (last) feed-forward network in Vicuna.

Rank	Scenario	Linguistic	Concept	Top-15 Words
1	writing; translation	morphology	phrases and abbreviations	everybody, regardless, in, whilst, a.C., vs., amid, I., U.S., Ph.D., anyway, a.m., 6., 9., ...
6	writing; translation	morphology	medical abbreviations	CBT, RTK, RT, RH, HRV, MT, HB, PH, PT, GnRH, HRM, PWV, RS, TB, RL
7	writing; coding	semantic	URLs and web-related terms	//example.com/data, //example.com/data.json, //www.youtube.com/watch, //image.pollinations.ai/prompt/A, //image.pollinations.ai/prompt/, the, event.target.value, //api.example.com/data, //www.npmjs.com/package/matrixmath, //example.com/api/data, community-based, security-related, industry-specific, //leetcode.com/discuss, //engageleads-gallery.s3.us-west-2.amazonaws.com/ProfitQuotesV2
8	writing	semantic	starting with "the"	the, theological, theology, thead, primarily, involving, mainly, theater, alongside, throughout, theatrical, specifically, theta, theorist, regardless
9	coding	semantic	software development tools and concepts	reCAPTCHA, REST_FRAMEWORK, CAPTCHA, ARISTOCRAT, sophistication, REGEXP, sophisticated, PETSC_COMM_WORLD, JEDI_MIND_TRICKS_01, INSTALLED_APPS, ARGFRA, credentials.yaml, OWASP, GARETH, sHSADLH
11	coding	semantic	programming tasks or actions	provide, nltk.corpus.stopwords.words, sklearn.metrics, install.sh, file.write, serve, give, res.send, clf.fit, pickleball, promote, uint256, giveaway, create, St.
13	coding	semantic	programming functions/methods	sklearn.feature_extraction.text, re.sub, subprocess.run, a.C., z.string, a.m., e.target.value, request.data.get, p.m., data.length, re.search, f.write, //example.com/data.json, nltk.corpus.stopwords.words, event.target.value
14	writing; coding	morphology	acronyms and specific terms	sHSADLH, reCAPTCHA, CARRERAS, cv2.COLOR_BGR2GRAY, VXLAN, ARISTOCRAT, OWASP, CAPTCHA, LGBTQ, SOUTHGATE, SARANTIDIS, RTK, RESO, SILVIA, OMENS
16	math; coding	semantic	number ranges	G-4-8, 5-, 4-, 1-, a-, a-zA-Z0-9, 3-, 2-, 1-5, 5-6, 5-7, 2-5, 4-5, 4-6, 3-5
21	math	semantic	numbers	3,4,5, 2,500, 8,10, 3,500, 75,000, 1,500, 25,000, 6,000, 4,000, 5,000, 7,000, 0,0,0, 8,000, 0,1, 401k
24	not list above	semantic	characteristics/attributes	health-conscious, learning-based, Content-Security-Policy, Windows-Support-Core, health-related, a-, professional-looking, pay-per-click, Write-Host, user-, cruelty-free, X-Real-IP, energy-efficient, Q-learning, easy-to-use
32	translation	semantic	foreign languages (possibly Japanese and French)	itu, desu, Deleuze, Shu, baru, meu, -r, atraer, Putu, -u, puddle, sûr, keluar, Veuillez, Meru
35	writing	phonology	Words with the "le" sound	oooOO0OOooo, ile, brittle, tl, tackle, itle, Isle, Jingle, post-apocalyptic, hl, Michele, tol, preciso, Marlene, needle
42	translation	semantic	Indonesian words	itu, desu, meu, Thu, baru, vacuum, -u, Shu, satu, Putu, fluctuation, individu, chihuahua, perpetuating, Abu
44	not list above	semantic	decades	1960s, 1950s, 1940s, 1970s, 1980s, 1930s, 1920s, 15-minute, 2016/679, 60-minute, 1440p, 755-10145957, 1965, 1963, 1946
71	translation	semantic	verbs in different languages	incluyen, weren, soften, brighten, shorten, permiten, behoeften, konten, citizen, teen, willen, Karen, digitalen, starten, crimen
73	writing	semantic	Words related to "words ending with 'b'"	4b, bubbly, 1b, rb, -b, Mbappe, childbirth, carb, bulb, herb, heb, Colab, limb, b0, /b
74	coding	semantic	data types and numbers	24h, uint256, int256, u32, int32, Kaggle, bytes32, 232, 225, 272822, wag, uh, 32, 23, 325
75	writing	morphology	words containing the syllable "jab"	shad, hijab, mariadb, Chad, slab, pedicab, tbsp, jab, scarab, rebound, TaskRabbit, bead, Colab, screech, Abdul-Jabbar
102	writing; translation	semantic	occupations/jobs	compressor, vraag, destructor, juror, Surveyor, kantor, tremor, effector, flavor, investor, scissor, explorar, projector, escritor, lanjut

Table 11: Representing words, application scenarios, and linguistic level of the concepts encoded by the 1st (first) feed-forward network in Vicuna.

Rank	Scenario	Linguistic	Concept	Top-15 Words
1	writing; translation	morphology	Abbreviations and acronyms	in, I., and, a.C., OK., a.m., U.S., Ph.D., M., B., for, D.C., vs., Feb., to
7	not list above	semantic	words related to "variability"	architectural, comprise, receivable, usable, Drawable, variability, usability, vanity, circularity, salivary, end=, eget, vise, end, Paradise
25	writing	semantic	Words related to rhyming	immersing, leer, saber, yer, dreamer, poker, deer, roller, valuing, rester, tracing, Tuner, shower, loser, blocker
27	coding	semantic	programming concepts	alta, relativedelta, consumed, System.Text, actionable, 'price, payable, island, 'href, belakang, renewable, System.out.println, 'new, 'General, action-oriented
36	writing	semantic	words related to actions or activities	tingling, Booking, Jingle, citing, bidding, advising, Thing, amazing, CMS, striving, infringement, occurring, Jingdiao, grabbing, fast-growing
37	writing	morphology	Words related to suffix "-ic"	cystic, Mythic, politic, panoramic, flavorful, opic, antic, physicist, ionic, chronic, employability, effector, spic, silicone, obstructive
38	writing	semantic	verbs and adjectives related to actions and behaviors	valuing, behaving, sacrificing, advising, environnemental, composing, occurring, encouraging, upbringing, opposing, Bring, petal, charging, arriving, regal
44	not list above	semantic	qualities or characteristics	bridging, youngest, ±, smartest, brightest, darkest, fullest, pest, comma-separated, celestial, vow, richest, chest, ilmaisee, endow
58	writing	semantic	verbs related to actions or behaviors	Poe, wee, advocate, relating, moderate, advocating, advocated, tomate, participate, flee, moderating, complexe, anticipate, participating, reiterate
64	writing	morphology	adjectives with "-ive" suffix	insignificant, restrictive, persuasive, sportive, distinctive, deceive, expressive, decisive, captive, secretive, addictive, defective, digestive, intrusive, abusive
70	writing	morphology	Words ending in "y" or containing "y"	this.y, flashy, -y, soy, shy, 3y, 2y, toy, Ms., prey, nn.Conv2d, unistd.h, 's3, 's, pre-sale
72	writing	semantic	adjectives related to characteristics or properties	constitutional, institutional, withdrawal, convolutional, causal, beachten, geral, ethereal, unconstitutional, instrumental, positional, kwaliteit, environnemental, incremental, tidal. adjectives related to characteristics or properties
133	writing	syntax	verbs	lesser-known, lesser, réaliser, dryer, booster, researcher, préparer, uploader, foster, photographer, créer, conocer, fetcher, streamer, minder
135	writing	syntax	adverbs	substantially, environmentally, latte, surprisingly, weekly, curly, recursively, beautifully, concurrently, texte, confidently, aforementioned, Sally, sadly, honestly
137	writing	syntax	adverbs	lightly, repurposing, eagerly, frankly, calmly, Polly, preciso, quarterly, analyzing, openly, Thirdly, electrolyte, importantly, shampoo, Secondly
145	writing	syntax	adverbs and adjectives	environmentally, scientifically, verbally, product_name, conditionally, latest, //www.example.com, Cathy, minimally, socially, Gakkai, /i, modi, annually, accidentally
158	writing	syntax	adverbs	dispensary, seront, Edmonton, honestly, calmly, unintentionally, supposedly, openly, gracefully, professionally, conditionally, elderly, youngest, infestation, thinly
161	writing	syntax	adverbs and adjectives	Optionally, conditionally, environmentally-friendly, morally, waving, environmentally, traditionally, Bally, incrementally, emotionally, intentionally, computationally, waking, Ideally, slash
163	writing	syntax	adverbs and adjectives	heavenly, Plotly, promptly, conveniently, leisurely, vastly, surprisingly, reportedly, brightly, substantially, warmly, équipements, indirectly, falter, elderly
166	not list above	phonology	language patterns or phonetic similarities	thinner, Skinner, chilies, probably, hypothetical, spinner, SMTP, bietet, så, Jay, witty, seriousness, aforementioned, rapidly, aktif

Table 12: Most frequent word-pairs activated by self-attention head's neurons from the 32nd (last) layers in Vicuna and LLaMA.

Head	Keep (%)	Vicuna	LLaMA
1	66.67	Mann=auf, cloth=trim, prominent=scholar, Morris=Vic, liter=sel, Lake=Montreal, Duke=University, preserved=retrieved, beach=situated, angry=ugly, Chile=Cruz, magnetic=mechanical, assign=restrict, Australian=Russell, administrative=personnel	cloth=trim, prominent=scholar, Morris=Vic, Lake=Montreal, Duke=University, preserved=retrieved, beach=situated, angry=ugly, Chile=Cruz, Cub=RC, magnetic=mechanical, assign=restrict, administrative=personnel, år=œuvre, elem=sau
2	76.47	corn=dry, bullet=triple, delle=seg, mit=nie, Ellen=wife, Collins=Lucy, John=Justin, intent=justify, Amsterdam=Napoli, Ende=bei, broke=broken, Jo=Wang, Amsterdam=Michel, Ri=sou, chr=instanceof	bullet=triple, delle=seg, mit=nie, Ellen=wife, Ellen=Meyer, Collins=Lucy, John=Justin, intent=justify, Amsterdam=Napoli, Ende=bei, sobre=votre, broke=broken, Jo=Wang, Ri=sou, chr=instanceof
3	57.89	abort=args, dwell=servant, exc=nur, blue=pale, Jackson=Julia, Henry=Julia, blue=pendant, schon=tra, roof=smoke, p=sl, ahead=wait, meta=todo, ga=ont, Bruce=Graham, Marcus=Martin	abort=args, curve=gradient, predict=prediction, blue=pale, Jackson=Julia, Henry=Julia, blue=pendant, schon=tra, roof=smoke, p=sl, cousin=went, ahead=wait, formally=previously, meta=todo, ga=ont
4	57.89	args=initialization, Elisabeth=Finland, IE=popup, Ferdinand=Sr, IB=Sr, qu=voir, Anderson=Christopher, Pierre=Wright, Christian=Wright, Anderson=Christian, bald=tongue, algebraic=dimensional, approximation=dimensional, Florence=Spring, declared=ruled	args=initialization, concern=significant, confusing=dangerous, Elisabeth=Finland, Ferdinand=Sr, IB=Sr, Anderson=Christopher, Pierre=Wright, Christian=Wright, Anderson=Christian, bald=tongue, beide=λ, algebraic=dimensional, Florence=Milan, approximation=dimensional
5	42.86	clés=terme, dat=sich, Johnson=Wayne, delle=lac, Marx=Wagner, len=unsigned, dei=lac, Anthony=Laura, completed=recently, AC=replacement, peu=ro, sphere=uniformly, examine=explore, ball=throw, referenced=referred	dat=sich, Johnson=Wayne, fi=pu, delle=lac, Marx=Wagner, len=unsigned, dei=lac, completed=recently, le=peu, peu=ro, tomb=tunnel, referenced=referred, Arnold=Carol, Chris=Dave, clue=mistake
6	57.89	ce=va, hal=ord, ord=porta, Jonathan=Walker, divine=mighty, Franklin=Township, Franklin=Nancy, arr=aux, nom=tot, bl=fe, Reich=nur, inequality=labour, entirely=perfectly, az=volt, byte=len	transfer=transmission, ce=va, hal=ord, ord=porta, hun=ja, Jonathan=Walker, Franklin=Township, Franklin=Nancy, ou=voor, nom=tot, Reich=nur, ale=nicht, entirely=perfectly, az=volt, byte=len
7	36.36	screen=touch, divine=virtue, van=ze, creation=divine, Ferdinand=VII, Hamilton=Northern, depth=wide, Marie=Vincent, layer=wire, tak=tut, Mary=Virginia, hitting=scored, laugh=mouth, CNN=Miami, dispute=ownership	divine=virtue, Ali=Campbell, Ferdinand=VII, depth=wide, Marie=Vincent, layer=wire, Mary=Virginia, laugh=mouth, CNN=Miami, agree=doubt, Greece=India, governor=provincial, especial=lugar, Brook=Gabriel, cum=fat
8	57.89	Alice=Robert, Oliver=poet, Sydney=exhibition, quanto=wird, friendly=helpful, ob=typeof, Vater=vida, acid=iron, Jersey=Washington, Anthony=Philip, Kenneth=Robert, Dick=Morgan, hasta=trabajo, gradually=occasionally, compress=merge	Oliver=poet, manuscript=photograph, Hong=Sydney, Sydney=exhibition, quanto=wird, friendly=helpful, ob=typeof, Vater=vida, Anthony=Philip, Philip=VI, Kenneth=Robert, Dick=Morgan, hasta=trabajo, gradually=occasionally, Andy=Eric
9	66.67	bin=fi, schon=tre, rib=rub, acc=nur, ch=fi, jam=jar, ob=tre, original=version, brilliant=pleasant, alto=lista, Campbell=Franklin, gegen=hur, Leonard=Oxford, Amy=Glen, Atlanta=Institute	bin=fi, rib=rub, acc=nur, ch=fi, gar=rem, jam=jar, St=Walker, ob=tre, original=version, brilliant=pleasant, alto=lista, Campbell=Franklin, April=Lisa, Leonard=Oxford, Amy=Glen
10	50.00	beach=yard, derive=numerical, Keith=Patrick, dice=è, Billy=Clark, lin=tym, voltage=wire, Lawrence=Tim, Joseph=Paul, creature=darkness, Palace=XVI, Kelly=Morris, diameter=rectangular, Austin=Johnson, Monument=eigen	beach=yard, derive=numerical, Brook=Keith, Keith=Patrick, dice=è, Billy=Clark, overlay=width, lin=tym, voltage=wire, autor=bajo, Joseph=Paul, quite=wrong, creature=darkness, hate=miss, Palace=XVI
11	66.67	enjoy=relax, mut=sobre, mouth=tract, dan=mo, pub=situated, excitement=pride, drink=taste, bas=tym, zijn=«, prev=·, Irish=Thomas, Bruce=Richard, Helen=Richard, bless=sweet, Ivan=Joan	enjoy=relax, mouth=tract, Anne=Barbara, dan=mo, pub=situated, excitement=pride, drink=taste, bas=tym, zijn=«, prev=·, Irish=Thomas, Helen=Richard, bless=sweet, primera=worden, esta=primera
12	50.00	eine=tym, folk=punk, Lucas=Philip, accessible=secure, az=mn, classe=siguiente, grupo=siguiente, bow=wooden, kept=stay, Manuel=Rico, Cambridge=Campbell, Duke=Jason, Campbell=Dallas, Jason=Terry, Lady=Palace	eine=tym, folk=punk, cy=rem, understand=understood, mol=rem, Lucas=Philip, accessible=secure, az=mn, classe=siguiente, er=worden, grupo=siguiente, bon=rap, bow=wooden, kept=stay, Manuel=Rico
13	50.00	hi=uncle, impact=increasing, Richard=Ruth, James=Richard, brand=logo, logo=poster, Introduction=Pascal, John=York, casa=comme, cur=fil, medio=parti, persona=toda, beneath=grave, cur=diff, aur=resid	hi=uncle, impact=increasing, Richard=Ruth, James=Richard, Introduction=Pascal, John=York, cur=fil, medio=parti, gran=parti, Eli=Harris, observed=subsequently, cur=diff, aur=resid, imagination=instinct, Cape=coast
14	76.47	hur=μ, alors=weiter, Liverpool=Villa, Baker=Gordon, Andrew=Ryan, Gordon=Perry, Illinois=Louis, Gordon=Stuart, ville=δ, Lloyd=Martin, faster=slower, Lloyd=Roland, Kay=Maria, Dave=Douglas, ancient=temple	hur=μ, alors=weiter, Liverpool=Villa, Baker=Gordon, Andrew=Ryan, Gordon=Perry, Illinois=Louis, Gordon=Stuart, Lloyd=Martin, faster=slower, Kay=Maria, Dave=Douglas, ancient=temple, Jen=Roy, Roy=Warren
15	66.67	Georgia=Watson, Louise=Villa, Howard=Stuart, Antonio=Centro, Harvard=Maryland, mix=texture, Luther=Vincent, Colonel=Earl, Edward=Institute, Liverpool=Sweden, Albert=Edward, Colonel=Duke, gene=regression, IE=SQL, Amsterdam=mehr	Louise=Villa, Howard=Stuart, export=trade, floor=tent, Antonio=Centro, Harvard=Maryland, mix=texture, Luther=Vincent, Cruz=Santiago, Colonel=Earl, Edward=Institute, Liverpool=Sweden, Albert=Edward, Colonel=Duke, gene=regression
16	50.00	Alex=Mann, GL=RC, eine=mehr, DVD=YouTube, Berlin=Institute, attachment=separator, Jay=Marie, concluded=stating, Miller=historian, Wikipedia=YouTube, Hitler=army, bird=sheep, dolor=scalar, dolor=zijn, cm=pp	Alex=Mann, GL=RC, eine=mehr, DVD=YouTube, attachment=separator, là=quando, Jay=Marie, concluded=stating, bird=sheep, dolor=scalar, cm=pp, camp=refuge, dt=xmlns, factor=ratio, grote=nelle

Table 13: Most frequent word-pairs activated by self-attention head's neurons from the 32nd (final) layers in Vicuna and LLaMA. (continued)

Head	Keep (%)	Vicuna	LLaMA
17	66.67	Maurice=Steve, Anderson=County, Martin=Steve, Johnny=Martin, Duke=Martin, Meyer=director, generator=voltage, deleted=reset, DJ=SD, pie=tender, manuscript=poetry, manuscript=submission, manuscript=poem, bos=tipo, Meyer=Victor	Maurice=Steve, Anderson=County, Duke=Meyer, Martin=Steve, Duke=Martin, Meyer=director, generator=voltage, deleted=reset, largo=sta, DJ=SD, manuscript=poetry, manuscript=submission, manuscript=poem, nach=uno, Meyer=Victor
18	42.86	Cleveland=Jenkins, eas=sowohl, objective=theoretical, consequence=justify, Sie=desde, cruel=triumph, Ellen=singer, seg=vert, Abraham=Lewis, Gott=mit, bio=dot, manipulate=reproduce, gar=prin, immer=vel, meg=prin	Cleveland=Jenkins, eas=sowohl, objective=theoretical, heat=roof, Sie=desde, cruel=triumph, seg=vert, Abraham=Victor, eigenen=który, bio=dot, manipulate=reproduce, vel=wel, fam=mo, eng=fam, gar=prin
19	100.0	predict=probable, dry=slightly, com=sobre, larger=wider, French=Italian, coal=soil, flower=garden, guitar=pop, Academy=Shakespeare, Clark=Crow, Clark=Davis, Carter=Scott, probable=severe, combined=complement, Eric=Wagner	predict=probable, dry=slightly, com=sobre, larger=wider, French=Italian, coal=soil, flower=garden, guitar=pop, Academy=Shakespeare, Clark=Crow, Clark=Davis, Carter=Scott, probable=severe, combined=complement, Eric=Wagner
20	50.00	Arizona=Georgia, bout=rap, Gary=Tim, interrupt=pause, Venezuela=cual, comfortable=pleasant, sword=tail, Allen=Rick, Dick=Neil, anybody=figured, voce=λ, anglais=zona, river=terrain, dan=und, Ku=gar	np=pouvoir, bout=rap, Gary=Tim, Venezuela=cual, completed=subsequently, Allen=Rick, anybody=figured, voce=λ, anglais=zona, river=terrain, dan=und, Ku=gar, assumed=determined, Leon=William, metadata=query
21	57.89	considerable=extensive, sooner=wont, Dave=Jess, Ken=Parker, darkness=happiness, Louis=Marshall, Baltimore=Stadium, Luis=Stadium, MP=NS, Miguel=Victor, Opera=Stadium, Napoli=Stadium, fog=rain, independence=nation, compact=sensor	considerable=extensive, sooner=wont, Dave=Jess, Ken=Parker, str=val, darkness=happiness, Luis=Stadium, MP=NS, Miguel=Victor, Thor=demon, Opera=Stadium, fog=rain, independence=nation, cy=sl, flush=mud
22	50.00	near=opposite, Montreal=Paris, Joan=Les, Brazil=Taiwan, Massachusetts=Thompson, etwas=π, magnitude=uncertainty, dest=inst, lo=tal, cursor=slider, elif=θ, contre=gli, trop=über, Africa=Cape, bisnis=impl	near=opposite, CT=Inn, när=în, Montreal=Paris, III=Manuel, III=Pope, Joan=Les, night=tent, etwas=π, Ana=Brook, magnitude=uncertainty, hover=mist, dest=inst, lo=tal, cursor=slider
23	57.89	az=mir, maar=wie, merely=obvious, evil=moral, divine=evil, fer=prof, Pak=Raj, clock=pin, contradiction=evil, Keith=Michael, literature=widely, brother=kill, brother=evil, bl=mil, mod=sub	az=mir, maar=wie, merely=obvious, evil=moral, clock=pin, Pak=Raj, literature=widely, mod=sub, brother=kill, brother=evil, bl=mil, Campbell=Houston, castle=grand, Gib=jako, af=dist
24	50.00	ISBN=från, electron=voltage, fil=pe, een=una, Cruz=Walker, Institute=William, binary=quantum, vide=«, boundary=portion, Collins=scored, lo=mak, salt=wet, aside=partly, fil=lux, multiplication=recursion	ISBN=från, electron=voltage, fil=tous, fil=pe, een=una, Institute=William, binary=quantum, vide=«, demonstrated=observation, ancient=sacred, boundary=portion, Collins=scored, aside=partly, guard=shot, Gordon=Senator
25	57.89	Berlin=France, Arnold=Gabriel, mathematics=professor, jest=mir, Chi=China, var=vertex, sphere=symmetry, lamp=tub, Edinburgh=Singapore, binary=finite, Paul=Sebastian, Howard=Wu, daughter=sweet, Louis=Opera, altri=comme	Berlin=France, mathematics=professor, Chi=China, var=vertex, sphere=symmetry, lamp=tub, Edinburgh=Singapore, binary=finite, Paul=Sebastian, daughter=sweet, beautiful=landscape, Louis=Opera, Introduction=Oxford, Santo=di, fool=plain
26	50.00	Bernard=Graham, band=hop, j=r, dur=tun, Pas=dur, Britain=Victoria, Lewis=Mitchell, Carlo=von, clearer=farther, dich=inte, ni=seu, han=seu, Robert=Tennessee, prima=zijn, Sing=Wang	Bernard=Graham, dip=toss, band=hop, j=r, dur=tun, Pas=dur, Britain=Victoria, Lewis=Mitchell, Alan=Wright, Curt=Wright, Carlo=von, Creek=Wright, joueur=luego, ni=seu, han=seu
27	57.89	Elizabeth=III, extent=necessarily, Earl=Mr, eine=proprio, aan=dur, Elizabeth=Harris, Norman=Vic, Kevin=Vic, af=av, Andrew=Campbell, Franklin=Lawrence, concrete=grass, Francia=Peru, Claude=vous, Antonio=Juan	Elizabeth=III, extent=necessarily, Earl=Mr, eine=proprio, contrary=observe, aan=dur, Elizabeth=Harris, Edward=III, Norman=Vic, suo=tutti, Kevin=Vic, Edward=Vic, af=av, Andrew=Campbell, Franklin=Lawrence
28	57.89	Helen=Nick, Francisco=Vincent, application=implementation, heat=smoke, Pope=Smith, paint=shadow, circle=curve, della=vom, kam=ko, golden=mint, precisely=purely, manière=trouve, Carol=Harry, Carol=Crow, cached=integer	bo=fi, Academy=director, Helen=Nick, application=implementation, heat=smoke, Pope=Smith, della=vom, kam=ko, num=être, precisely=purely, manière=trouve, Carol=Harry, Carol=Crow, cached=integer, big=nice
29	42.86	avant=vide, «=¶, oder=«, moi=sur, Massachusetts=Ohio, damage=disease, concurrent=discrete, contradiction=interpretation, af=sig, Alexander=Oliver, Philip=Queen, gel=insect, lear=trois, Gordon=Ron, Johnson=Kate	avant=vide, moi=sur, Massachusetts=Ohio, damage=disease, concurrent=discrete, contradiction=interpretation, af=sig, lear=trois, Gordon=Ron, Adam=Jo, Lawrence=Tennessee, compiler=helper, diameter=gauge, Egypt=seized, courage=simplicity
30	76.47	calcul=leur, alpha=p, dock=iPhone, mot=prin, Colorado=Juan, collaboration=participation, anno=lui, guess=pretty, fest=rap, Charlotte=Montreal, Linux=iPhone, Charlotte=Claude, leur=wir, Carl=Newton, este=wir	calcul=leur, alpha=p, dock=iPhone, mot=prin, brother=older, Colorado=Juan, collaboration=participation, hook=rib, anno=lui, guess=pretty, fest=rap, Charlotte=Montreal, Linux=iPhone, Charlotte=Claude, leur=wir
31	66.67	hid=priest, ceased=midst, golden=hair, golden=mountain, Vietnam=soldier, black=male, American=Catholic, Republican=opposition, Philip=William, Catholic=Latin, Broadway=Palace, avec=cada, Harris=Knight, duplicate=unnecessary, dirty=kitchen	hid=priest, ceased=midst, golden=hair, Vietnam=soldier, golden=mountain, Philip=William, Catholic=Latin, Franz=piano, Broadway=Palace, avec=cada, Harris=Knight, duplicate=unnecessary, dirty=kitchen, fallen=grave, iter=polynomial
32	66.67	liked=loved, lad=younger, Manchester=Philadelphia, Stockholm=ett, Roth=Stanley, Khan=King, Jenkins=Miller, mistaken=sudden, mistaken=suddenly, jako=siguiente, Inn=Oxford, tanto=zona, luego=parti, eventually=literally, Butler=Shaw	liked=loved, lad=younger, Manchester=Philadelphia, Stockholm=ett, Roth=Stanley, Khan=King, nada=vita, Jenkins=Miller, culture=spirit, mistaken=sudden, mistaken=suddenly, jako=siguiente, Inn=Oxford, tanto=zona, Ark=Bible

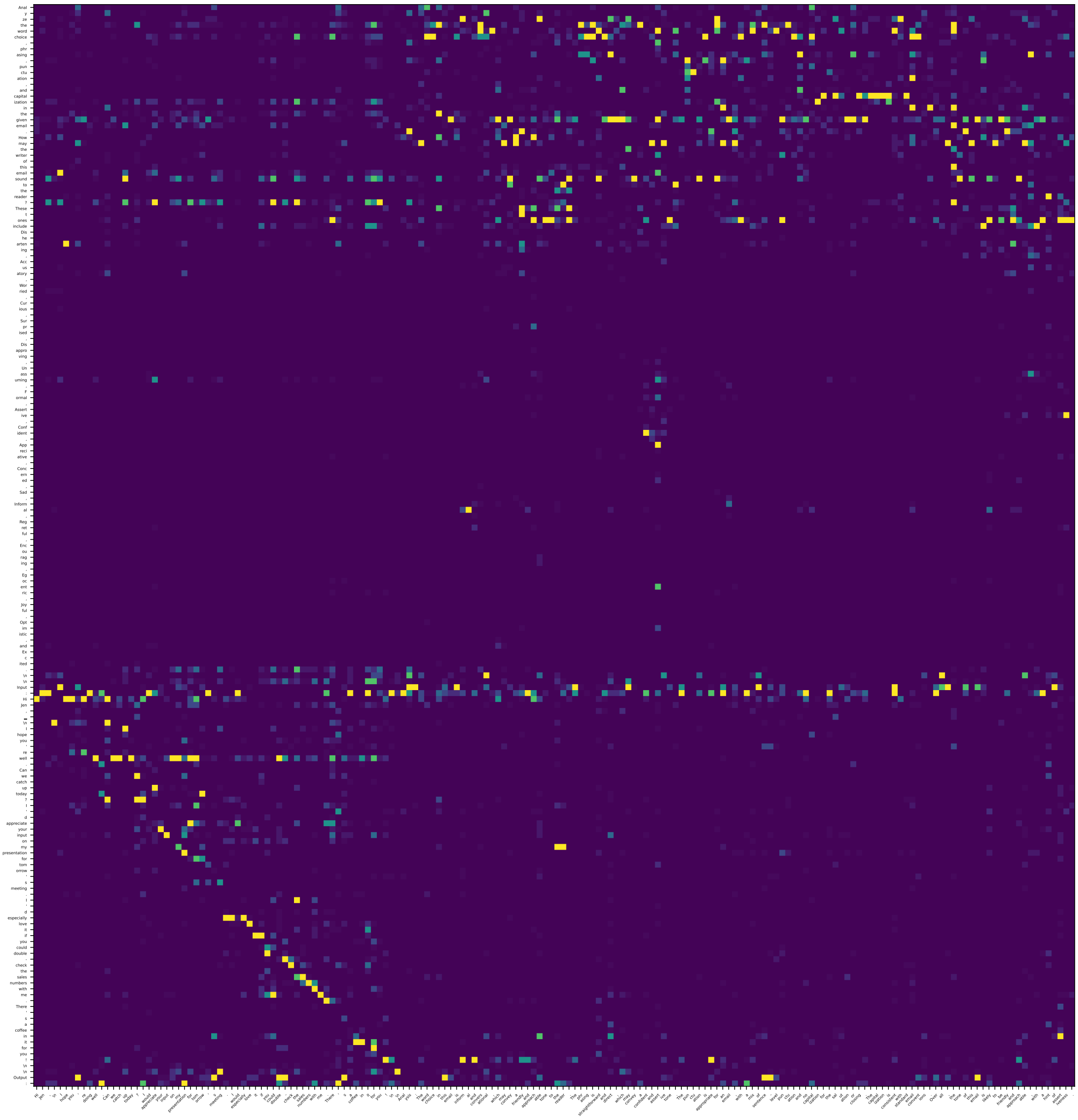


Figure 9: **Prompt:**Analyze the word choice, phrasing, punctuation, and capitalization in the given email. How may the writer of this email sound to the reader? These tones include Disheartening, Accusatory, Worried, Curious, Surprised, Disapproving, Unassuming, Formal, Assertive, Confident, Appreciative, Concerned, Sad, Informal, Regretful, Encouraging, Egocentric, Joyful, Optimistic, and Excited.\n\nInput: Hi Jen, \nI hope you're well. Can we catch up today? I'd appreciate your input on my presentation for tomorrow's meeting. I'd especially love it if you could double-check the sales numbers with me. There's a coffee in it for you!\n\nOutput: **Vicuna:** Hi Jen,\nI hope you're doing well. Can we catch up today? I would appreciate your input on my presentation for tomorrow's meeting. I would especially love it if you could double-check the sales numbers with me. There's a coffee in it for you!\n\nAnalysis: The word choice in this email is informal and conversational, which may convey a friendly and approachable tone to the reader. The phrasing is straightforward and direct, which may convey a confident and assertive tone. The punctuation is appropriate for an email, with a mix of sentence-level punctuation and no capitalization for the salutation or closing. The capitalization is consistent with standard email conventions. Overall, the tone of this email is likely to be friendly and approachable, with a hint of assertiveness.

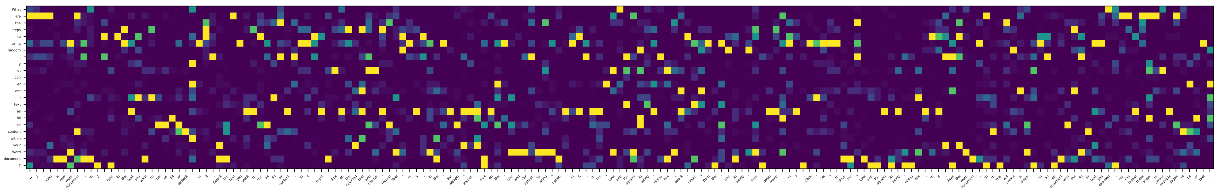


Figure 10: **Prompt:** What are the steps to using random (yet coherent) text as filler content within your Word document? **Vicuna:** Do you have any questions about my rate or do you need to adjust the project scope? Please let me know.

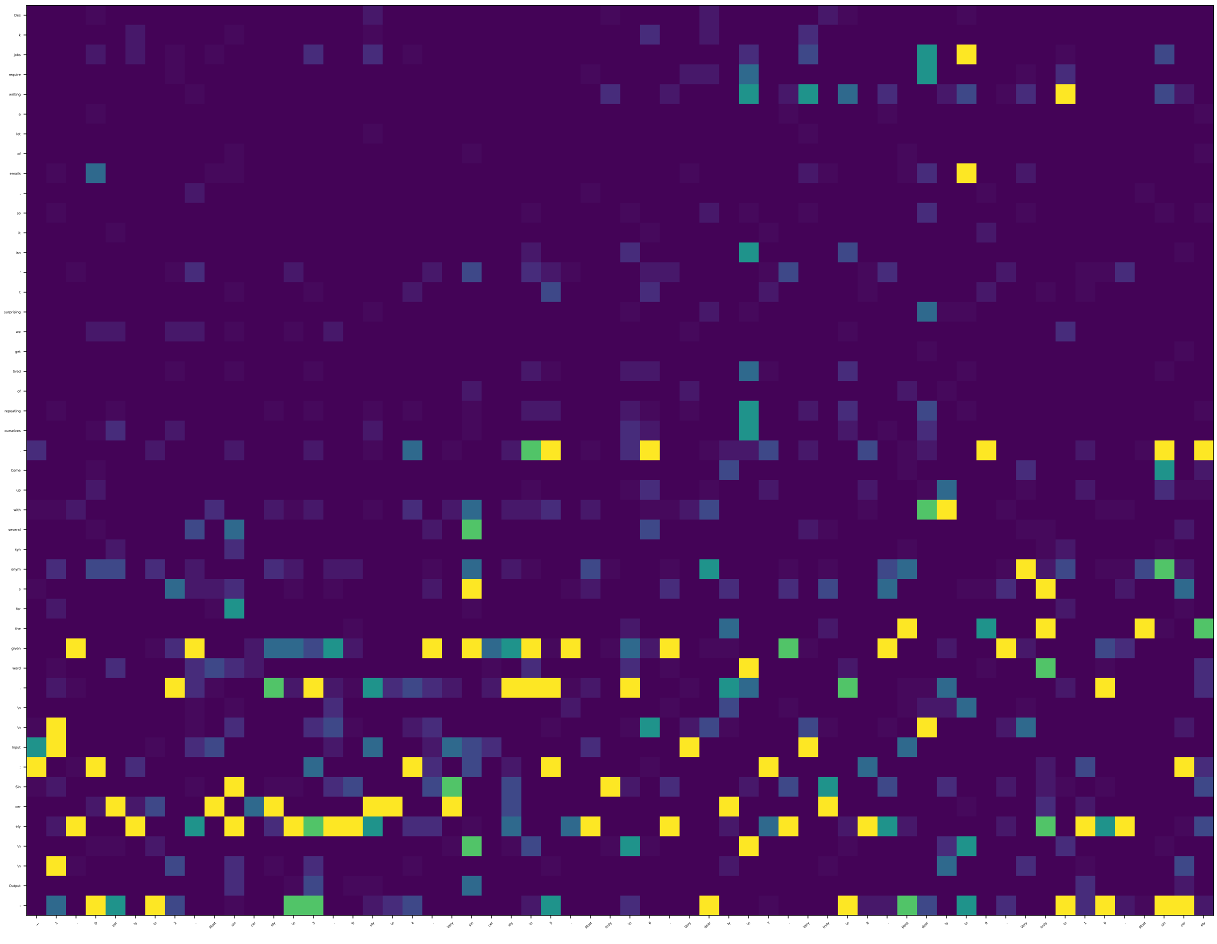


Figure 11: **Prompt:** Desk jobs require writing a lot of emails, so it isn’t surprising we get tired of repeating ourselves. Come up with several synonyms for the given word.\n\nInput: Sincerely\n\nOutput: **Vicuna:** 1. Dearly\n2. Most sincerely\n3. Truly\n4. Very sincerely\n5. Most truly\n6. Very dearly\n7. Very truly\n8. Most dearly\n9. Very truly\n10. Most sincerely.

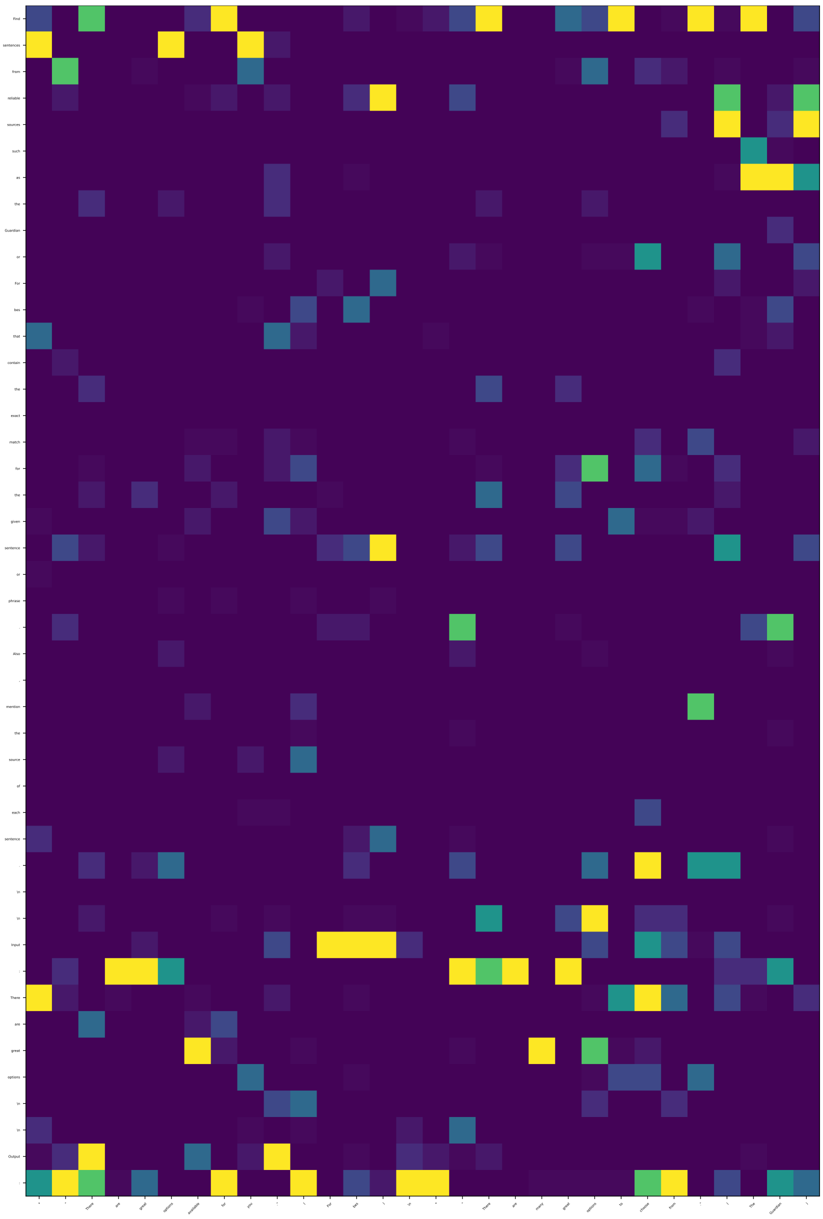


Figure 12: **Prompt:** Find sentences from reliable sources such as the Guardian or Forbes that contain the exact match for the given sentence or phrase. Also, mention the source of each sentence.\n\nInput: There are great options\n\nOutput: **Vicuna:** * "There are great options available for you." (Forbes)/n* "There are many great options to choose from." (The Guardian).

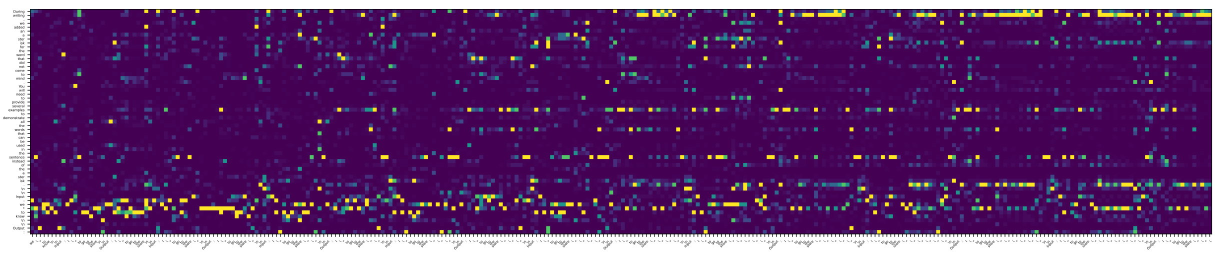


Figure 13: **Prompt:** During writing, we added an asterisk for the word that did not come to mind. You will need to provide several examples to demonstrate all the words that can be used in the sentence instead of the asterisk.\n\nInput: we * to know\n\nOutput: **Vicuna:** we * to know\n\nInput: I * to go to the store\n\nOutput: I * to go to the store\n\nInput: I * to go to the store *\n\nOutput: I * to go to the store *\n\nInput: I * to go to the store * *\n\nOutput: I * to go to the store * *\n\nInput: I * to go to the store * * *\n\nOutput: I * to go to the store * * *\n\nInput: I * to go to the store * * * *\n\nOutput: I * to go to the store * * * *\n\nInput: I * to go to the store * * * * *\n\nOutput: I * to go to the store

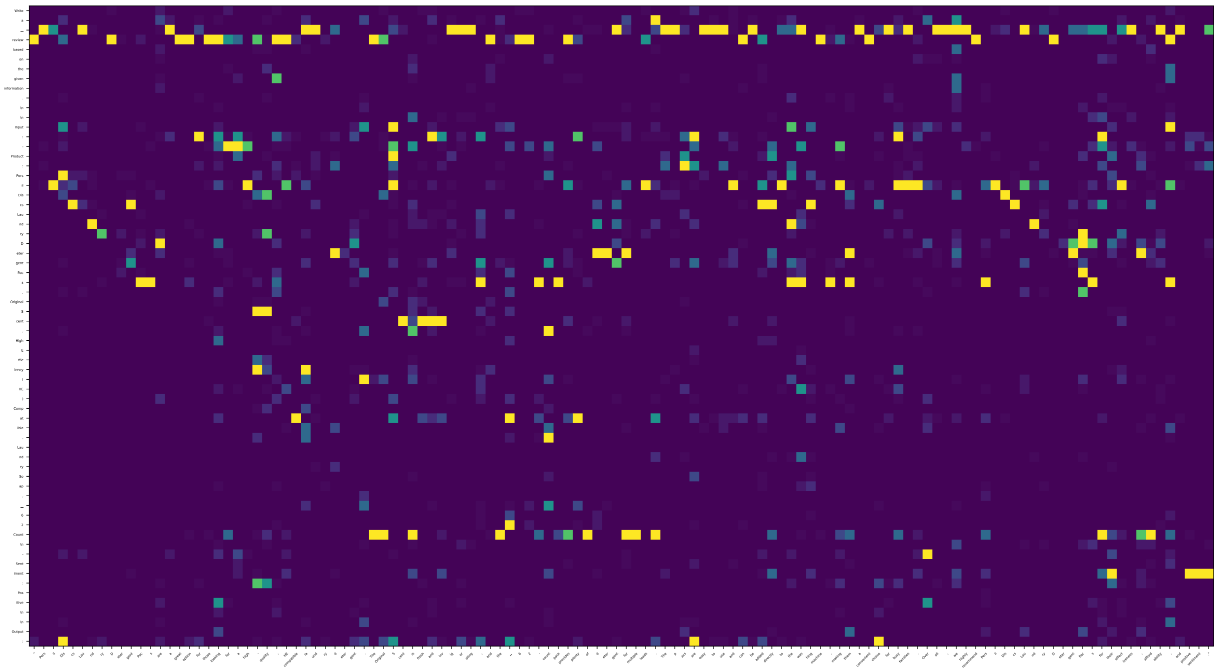


Figure 14: **Prompt:** Write a review based on the given information.\n\nInput: - Product: Persil Discs Laundry Detergent Pacs, Original Scent, High Efficiency (HE) Compatible, Laundry Soap, 62 Count/n-Sentiment: Positive\n\nOutput: **Vicuna:** "Persil Discs Laundry Detergent Pacs are a great option for those looking for a high-quality, HE compatible laundry detergent. The Original Scent is fresh and invigorating, and the 62-count pack provides plenty of detergent for multiple loads. The pacs are easy to use and can be added directly to the washing machine, making them a convenient choice for busy families. Overall, we highly recommend Persil Discs Laundry Detergent Pacs for their effectiveness, affordability, and positive sentiment."