Investigating Neurons and Heads in Transformer-based LLMs for Typographical Errors

Anonymous ACL submission

Abstract

This paper observes the inner workings when LLMs encode inputs with typos to understand robustness against typos. We hypothesize that specific neurons in FFN layers and attention heads in multi-head attention layers recognize typos and internally recover them to capture the originally intended meaning. We introduce a method to identify the **typo neurons** and **typo heads** that work actively only when inputs contain typos. Through our experiments with Gemma 2, the following findings are obtained: 1) Neurons in the early and early middle layers strongly respond to typos. 2) Few heads capturing contextual information also contribute to recovering typos. 3) The difference in the model size results in the different proportions of typo-related workload for neurons and heads.

1 Introduction

Large language models (LLMs) have been widely used in real applications (Dam et al., 2024), and their inputs are likely to contain typographical errors (typos). LLMs often make correct inferences on inputs with typos (Wang et al., 2024a), which suggests that LLMs can "recover" the words with typos to the originally intended meaning. However, LLMs sometimes imperfectly recover the meaning against typos, which might "damage" the performance of LLMs on downstream tasks (Zhuo et al., 2023; Wang et al., 2023; Zhu et al., 2023; Edman et al., 2024). To reduce the impact of typos on LLMs, it is essential to understand both their robustness against typos and the reasons for performance degradation caused by typos more deeply.

Existing studies have primarily focused on the surface-level exhibition of performance degradation due to typos (Wang et al., 2023; Zhu et al., 2023) and methods for improving robustness against typos (Zheng and Saparov, 2023; Zhuo et al., 2023; Almagro et al., 2023). Few studies have investigated how typos affect LLM's inner workings (Kaplan et al., 2024; García-Carrasco et al., 2024). However, the previous work focused on the case where the input has only one word with a typo, and there is a large room to be explored for the case where the typo appears with contextualized words, which is a more realistic situation. Besides, the previous work investigated the inner workings of LLMs with typos only from the viewpoints of attention heads. We believe that typos are recovered with contextual judgments across both neurons and attention heads, which are the main structural layers of the Transformer architecture.

We hypothesize that the robustness against typos is provided by inner workings such as neurons (**typo neurons**) and attentional heads (**typo heads**) with contexts around typos. We investigated the inner workings against typos in contextualized words using a word identification task (§3). Our work proposes a method to identify typo neurons (§4) and typo heads (§5). Then, we investigate how these neurons and heads change with different strengths of typos. Subsequently, we analyze the differences in their behavior between cases where the model is damaged by typos and cases or not.

We conducted experiments using Gemma 2 (Team et al., 2024) to investigate the inner workings when feeding inputs with typos to the LLM. Our findings on Gemma 2 suggest the following:

- There are neurons that perform typo recognition and typo-recovering in the early and early middle layers. Specifically, neurons in the early middle layer are responsible for the core of typo-recovering.
- A few heads that capture the basic grammatical information such as contexts and check the immediately preceding tokens also contribute to recovering typos.
- The workloads of typo neurons and typo heads differ depending on the size of LLMs.

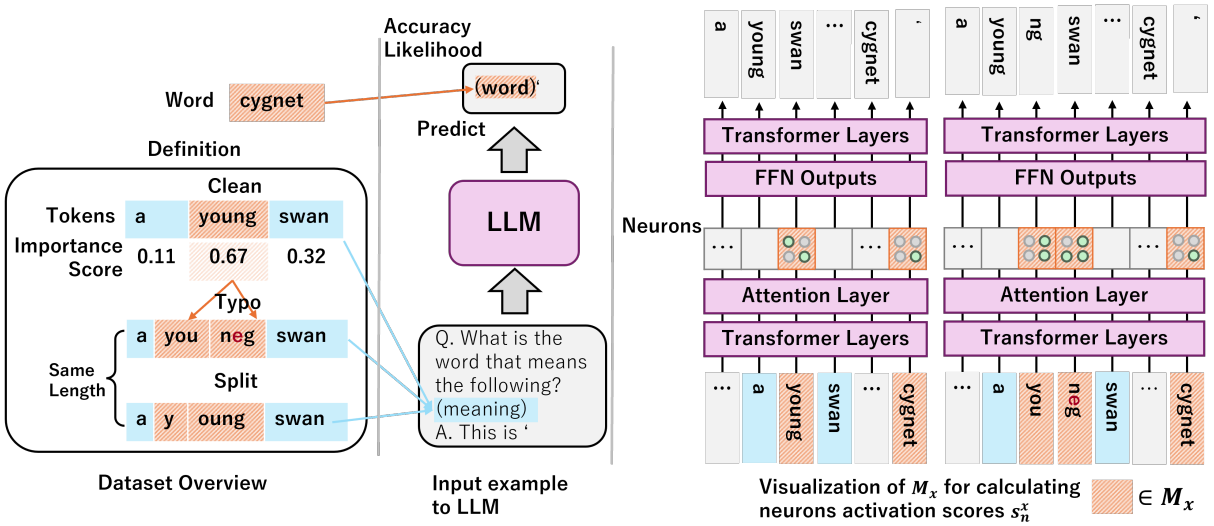

Figure 1: The dataset overview (left), an input example to LLM (middle), and the visualization of M_x for calculating neurons activation score s_n^x (right).

2 Related work

2.1 Analysis of LLMs against Typos

Typos are mistakes in writing or typing letters, categorized into insertion, deletion, substitution, and reordering (Gao et al., 2018). Research on the robustness of LLMs regards typos as a perturbation applied to input text. Typos changes the token sequence obtained through the tokenization process. Changing the token sequence potentially leads to a different output, even if the sentence is the same (Tsuji et al., 2024). Most existing LLM studies on typos focus on task performance by creating datasets to measure the model's robustness against perturbed inputs (Wang et al., 2021, 2023; Zhu et al., 2023; Edman et al., 2024) or modifying the architecture or prompts to improve robustness (Zhuo et al., 2023; Zheng and Saparov, 2023; Almagro et al., 2023). Chai et al. (2024) reported that the larger models are more robust to typos.

2.2 LLM's Interpretability

The feed-forward network (FFN) layer in the Transformer (Vaswani, 2017) has two linear layers separated by an activation function. Recent studies regard the output of the activation function as "neurons" that store knowledge (Geva et al., 2021). It has been reported that some neurons promote specific tasks (Wang et al., 2022, 2024c), knowledge (Dai et al., 2022; Bau et al., 2019; Gurnee et al., 2024), and behaviors (Hiraoka and Inui, 2024; Wang et al., 2024b; Chen et al., 2024).

Similar to neurons, some attention heads have been found to respond to specific knowledge (Gould et al., 2024; Voita et al., 2019; García-Carrasco et al., 2024) or behaviors (McDougall et al., 2024; Crosbie and Shutova, 2024). Additionally, some heads are responsible for merging multiple subwords of a word (Correia et al., 2019; Ferrando and Voita, 2024). Mosbach et al. (2024) concludes that understanding the inner workings is important to improve the model performance.

Kaplan et al. (2024) has investigated which layers are responsible for typo-recovering. However, they primarily focused on isolated words as inputs and only examined which layers recover typos. Our study is different from Kaplan et al. (2024) in focusing on neurons and attention heads and conducting experiments that allow contextual typo-recovering.

3 Preliminary

3.1 Research Overview

We created a dataset to investigate the typo-related phenomena (§3.3). Then, we applied typos to the dataset (§3.4) and conducted a preliminary experiment to observe accuracy when inputs include typos (§3.5). Next, we identify typo neurons and reveal their specific roles (§4). Similarly, we conduct analogous experiments for attention heads (§5).

3.2 Models

We used the 2B, 9B, and 27B models from Google's Gemma 2 (Team et al., 2024); only the 27B model was loaded in bfloat16, while the 2B

and 9B models were loaded in float32[1]. We conducted all experiments using greedy generation.

3.3 Clean Datasets without Typos

To investigate typos in contextualized words, we utilize a word identification task in which the LLMs are required to output a single word corresponding to a given definition. For instance, we feed the definition of the word as input, like "*a young swan*", to the LLMs, and then the model is expected to output the corresponding word "*cygnet*". Following Greco et al. (2024), we extracted 62,643 word-definition pairs from WordNet (Fellbaum, 2005)[2]. We created the word identification task with these pairs. We designed a prompt so that LLMs can solve this task as predicting tokens following outputs, as shown in the middle part of Figure 1.

For our analysis, we need a dataset composed of samples that LLMs can correctly answer when the samples do not include typos. Therefore, we extracted the top 5,000 word-definition pairs after sorting the samples by descending order of likelihood for the correct words. Note that we created unique datasets of the same size for three variations of Gemma-2 (i.e., 2B, 9B, 27B)[3].

3.4 Applying typos

3.4.1 Typo Dataset

Since our work focuses on text with typos, we manually applied typos to the definition part of the clean dataset created in §3.3. We selected the top t most important tokens depending on their importance scores on the word identification task. Then, we injected a random single letter or digit into each selected token as a typo. The importance scores are calculated with the method used in Wang et al. (2023); Li et al. (2019), using Gemma 2 2B. Specifically, we obtained the importance scores by performing back-propagation while predicting words from their definitions. This process assigns higher gradients to tokens that are important to predict the correct answer. For example, consider the sentence "*a young swan*" with $t = 2$ and the top two most important words are "*young*" and "*swan*." In this case, we inject random letters such as "*e*" and "*5*" into random positions[4] of each word, which results in "*a youneg s5wan.*"

3.4.2 Split Dataset

We often obtain a different number of subwords when tokenizing typo inputs compared to clean inputs. For instance, the Gemma-2 tokenizer encodes the word "young" into a single token, but it tokenizes the typo version "youneg" into two tokens (e.g., "you / neg"). When comparing the inner workings when LLMs encode the clean inputs and the typo inputs, the difference in the token length might prevent appropriate analysis[5].

To break down typo-related inner workings into the factor corresponding to typos and the one to tokenization difference, we created a "split-dataset" in addition to the "typo-dataset" mentioned in §3.4.1. The split-dataset comprises samples that are tokenized into the same number of tokens as the one with typos. For example, when the typo-dataset has a sample whose tokenized sequence is "a / you / neg / swan", an example of counterparts in the split-dataset is "a / y / oung / swan" whose length is equivalent to the one of the typo version. We can obtain the various tokenization candidates using the tokenizer and we randomly selected one candidate with the same length as the typo input. This process is shown in Figure 1 (left).

3.5 Preliminary Experiment

To examine the effect of typos on the model performance, we applied typos to t tokens ($1 \leq t \leq 16$) and analyzed the change in accuracy and average likelihood of predicting correct words.

Figure 2 shows the preliminary experimental results. The accuracy and the average likelihood of $t = 0$ indicate the performance of the clean data without typos. Since the clean data consists of samples that each model was able to answer correctly, the accuracy for all models is 1.0. While the accuracy of the 2B model drops to about 50% for the case of $t = 16$, the 9B model maintains more than 70%, and the 27B model more than 80%. This result supports the existing work reporting that the larger model has robustness against typos (Chai et al., 2024). This preliminary result also indicates that the robustness of larger models against typos is insufficient, resulting in a performance drop. This

[1]We used Xeon Gold 6230R + NVIDIA A100 40GB*2

[2]We used WordNet via NLTK (Bird and Loper, 2004) ver.3.9.1.

[3]Most samples overlap across three models.

[4]We exclude the positions before the spaces to avoid the situation where a typo would appear at the end of the previous token rather than within the target token.

[5]Kaplan et al. (2024) reported that there are inner workings to recover the original token from differently tokenized subwords. We need to exclude the effect of this factor to deeply focus on the typo-related inner workings.

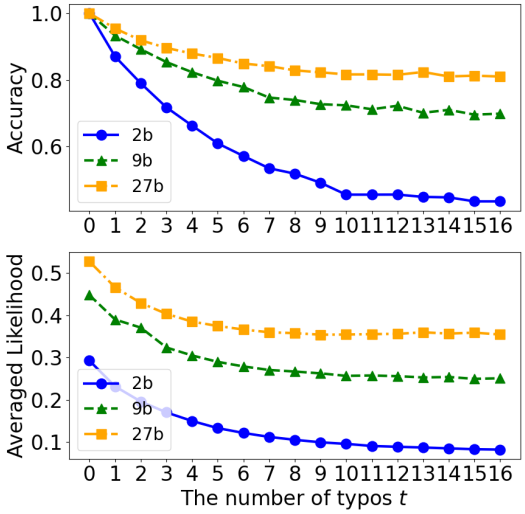

Figure 2: Accuracy (top) and likelihood (bottom) on the word identification task with a different number of tokens with typo t.

fact motivates us to reveal the inner workings related to typo inputs. For the average likelihood, the 2B model starts with a relatively low value; however, all models followed a similar pattern of decrease as the number of typos increased.

From the preliminary results, we conclude that typos damage performance, but larger LLMs have some robustness against typos. This fact motivates us to investigate the reasons for the differences in robustness against typos by model sizes for further improvement.

4 Typo Neurons

Some FFN layers have been found to combine multiple tokens into a single representation vector (Kaplan et al., 2024; Elhage et al., 2022; Lad et al., 2024). Additionally, it has been reported that certain neurons within LLMs function as "skill neurons" with specific roles (Wang et al., 2022). In this section, we investigate the existence of typo neurons, a particular type of skill neuron that is responsible for recognizing and recovering typos.

4.1 Method to Identify Typo Neurons

Following the approach of Hiraoka and Inui (2024), we compare the activation values of neurons between clean inputs and typo inputs to identify neurons that specifically respond to typos. Let $x = w_1, ..., w_m, ..., w_{|x|}$ be a sample of the completed input the word identification task composed of the prompt (e.g., "*Q. What is ... A. This is* ") and the answer (e.g., "*cygnet*"), where $|x|$ is the number of tokens comprises x.

The activation value s_n^X of a neuron n when feeding a dataset $X \ni x$ is defined as the following:

$$s_n^X = \frac{1}{|X|} \sum_{x \in X} \left(\frac{1}{|M_x|} \sum_{m \in M_x} f(x_1^m, n) \right), \quad (1)$$

where $|X|$ is the number of samples in the dataset. $f(x_1^m, n)$ is a function calculating the activation value of the neuron n corresponding to w_m when the LLM reads the input $x_1^m = w_1, ..., w_m$. M_x is a set of indices that indicates the token positions, and $|M_x|$ is the number of indices. We define M_x as the indices comprising the answer word tokens and t important words.

For example, in Figure 1, M_x for the clean input is composed of "young" and "swan", while M_x for the typo input is composed of "you", "neg", and "cygnet". Similarly, M_x for the split input is "y", "oung", and "cygnet". In the figure, tokens comprising M_x are indicated with an orange background.

We obtain the responsibility of neurons specialized to the typo inputs separated from clean and split inputs with the following score Δ_n:

$$\Delta_n = s_n^{X_{\text{typo}}} - \max\left(s_n^{X_{\text{clean}}}, s_n^{X_{\text{split}}} \right), \quad (2)$$

where X_{typo}, X_{clean}, and X_{split} are the typo, clean, and the split datasets, respectively.

A larger Δ_n indicates the neuron n that responds specifically to typos but not clean inputs or split inputs. Among the neurons, the top K neurons based on Δ_n scores are identified as typo neurons.

4.2 Experimental Results

This section investigates the typo neurons found with the method introduced in §4.1. We selected two settings of the number of typos, $t \in 1, 16$. Figure 3 shows the distribution of Δ_n and the distribution of the typo neurons in each layer. We extracted the top 0.5% of neurons with the highest Δ_n and $\Delta_n > 0$ as the typo neurons[6]. The distribution of Δ_n reveals that a few neurons have significantly larger scores than others both in $t = 1$ and $t = 16$, similar to knowledge neurons and skill neurons (Dai et al., 2022; Wang et al., 2022).

For $t = 1$, many typo neurons exist in the early layers (i.e., from 0.0 to 0.2). Especially in the

[6] $\Delta_n < 0$ indicates the fact that activations in inputs without typo are greater than activation in typo input.

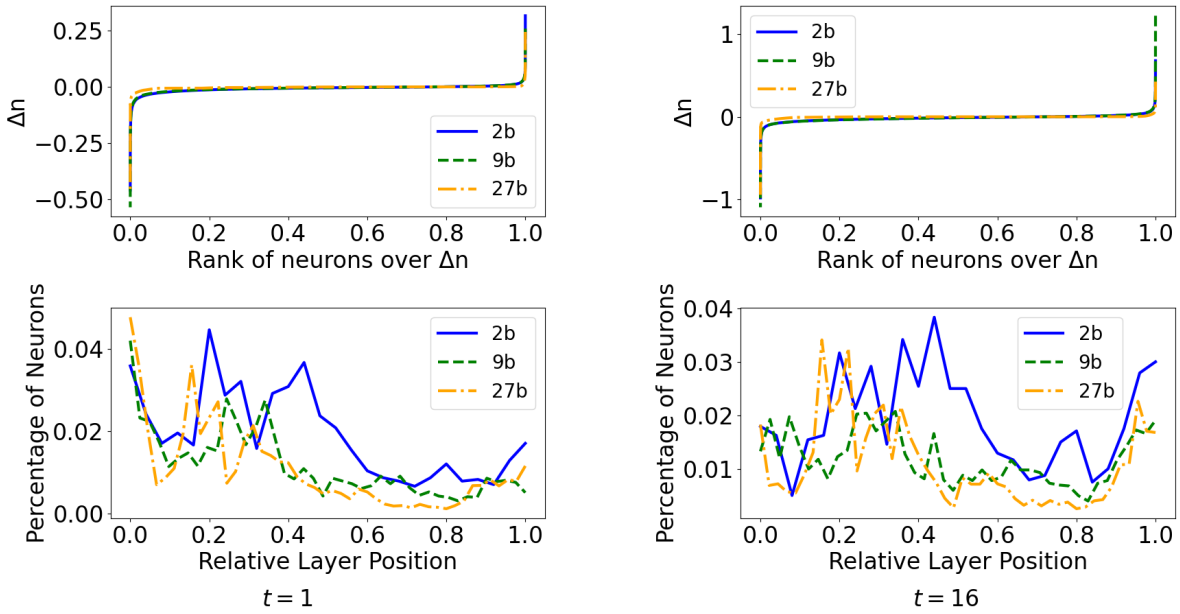

Figure 3: Distribution of Δ_n (upper) and percentage of typo neurons per layer (lower). Left figures are for $t = 1$ and right figures are for $t = 16$.

9B and 27B models, the largest number of typo neurons exist in the early layers. In contrast, for $t = 16$, the number of typo neurons in the early layers is decreased. The distribution of typo neurons per layer reveals that typo neurons are not limited to the early layers in all models. This partly supports the existing work (Lad et al., 2024) reporting that early layers perform *de-tokenization*, which integrates local context to transform raw token representations into coherent entities. Many typo neurons also exist in the early middle layers (i.e., from 0.2 to 0.5). In contrast to the early layers, increasing typos does not decrease the number of typo neurons in the early middle layers. This suggests that typo neurons in the early middle layers may play a significant role overall among typo neurons. This observation is consistent with Kaplan et al. (2024), which reported that typos are not recovered in the early layers but done in later layers. This result also aligns with Lad et al. (2024) reporting that early middle layers have the role of *feature engineering*, which iteratively builds feature representation depending on token context. Existing studies and our experimental results suggest that *feature engineering* with a wider context recovers the typos if *de-tokenization* with local context fails.

Additionally, neurons near the final layers exhibit significantly higher activation in all models for $t = 16$. We consider two possible causes for this phenomenon. First, as the number of typos increases, the corrections performed by the early layers become insufficient, leading to suppressed activation of typo neurons in the final layers. Second, as the number of typos increases, the internal state contains more errors. This may lead to different neuron activation in the final layers.

	2B		9B	
	Clean	Typo	Clean	Typo
Vanilla	1.00	0.86	1.00	0.93
⊖ Random Neurons	0.98	0.87	0.99	0.93
⊖ Typo Neurons	0.84	0.73	0.96	0.90

Table 1: Accuracy of the word identification task with neuron ablation (⊖) on clean and typo datasets. "Vanilla" indicates the accuracy without neuron ablation.

4.3 Discussion

While the experimental results in §4.2 suggest the existence of typo neurons, the impact of these typo neurons has not been clarified. This section investigates their impact in detail, focusing on 2B and 9B models in this section because we can see the different tendency of neurons between the small model (2B) and larger models (9B, 27B) in §4.2.

4.3.1 Neuron ablation

Typo neurons are expected to work typo-recovering. Therefore, ablating them should result in a significant decrease in performance in the typo inputs. In contrast, since they are not activated for clean inputs, ablating them is expected to have minimal impact on performance in the clean inputs.

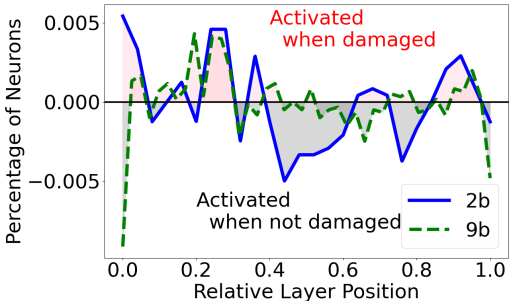

Figure 4: Distribution of typo neurons per layer for samples damaged or not. Values above the black line indicate many typo neurons activated when the LLMs predicted correct words.

We test this hypothesis by conducting ablation experiments on typo neurons and randomly selected neurons. From a dataset of 5,000 samples, 100 randomly selected samples were used to identify typo neurons. Then, we evaluate the performance of the word identification task using the remaining 4,900 samples by deactivating the identified neurons. Following the approach in §4.2, the top 0.5% of neurons were identified as typo neurons. We also randomly selected 0.5% of neurons as a baseline. Deactivation was performed by setting the output values of the neurons to zero. The experiments were conducted for the clean inputs and the typo inputs with $t = 1$.

Table 1 shows the experimental results. For typo inputs, performance remained largely unchanged when random neurons were ablated, regardless of the model. However, performance decreased when typo neurons were ablated. This suggests that a small number of typo neurons play a dominant role in typo-covering for typo inputs. For clean inputs, the ablation of typo neurons also resulted in a larger performance decrease compared to the random neuron ablation. This indicates that typo neurons may not exclusively act on typos but could also play a crucial role in processing general grammar or morphological features.

4.3.2 Neurons for Typo-recovering

The experiments in §4.2 sought typo neurons by comparing clean and typo inputs without considering whether the LLMs could correctly solve the task with typo inputs. This section focuses on the difference in typo neurons between cases where the LLMs answer with typos correctly and incorrectly.

From the dataset of 5,000 samples, we extracted 100 samples where typos did not damage the inferences and the correct word was predicted. Similarly, we extracted another 100 samples where typos damaged the inferences and led to incorrect word prediction. We compared differences in the activation of typo neurons in these two groups. We conducted this experiment with $t = 1$ and compared by the difference in the layer distribution of the typo neurons that have the top 0.5% Δ_n.

Figure 4 shows the result. In the 9B model, the number of typo neurons in the early layers increases when incorrect inferences are predicted. This suggests that some neurons in the early layers might play other roles than typo-related phenomena, and activation of those neurons prevents correct recognition of typos. In the 2B model, neurons in the middle-middle layers were activated when incorrect predictions were more frequent. This difference between model sizes can be attributed to the fact that, as described in §4.2, the 2B model has fewer typo neurons in the early layers and relies more heavily on the middle layers. Across all models, more typo neurons in the early middle layer were activated when typos did not damage inferences. This indicates the importance of typo neurons in the early middle layers.

5 Typo Heads

5.1 Method to Identify Typo Heads

Typo-recovering may not be solely dependent on neurons but also relates to subword merging by attention heads (Correia et al., 2019; Ferrando and Voita, 2024) and based on contextual understanding. Such heads are expected to become nearly uniform attention across all tokens for clean inputs while showing concentrated attention between specific tokens for typo inputs.

In this section, we investigate the attention heads specialized to typo inputs by comparing attention maps. Herein, we calculated the KL divergence between a uniform distribution and the rows of attention maps by considering them as a probability distribution. The KL divergence increases monotonically with the number of tokens, which can result in higher values for typo inputs or split inputs, as they often have more tokens than clean inputs. We alleviate this problem by normalizing the KL divergence with the maximum score $\log_2 m$, defined as follows:

$$s_h^X = \frac{1}{|X|} \sum_{x \in X} \left(\sum_m \left(\frac{D_{\mathrm{KL}}(P_{x,m,h} || U_m)}{\log_2 m} \right) \right), \quad (3)$$

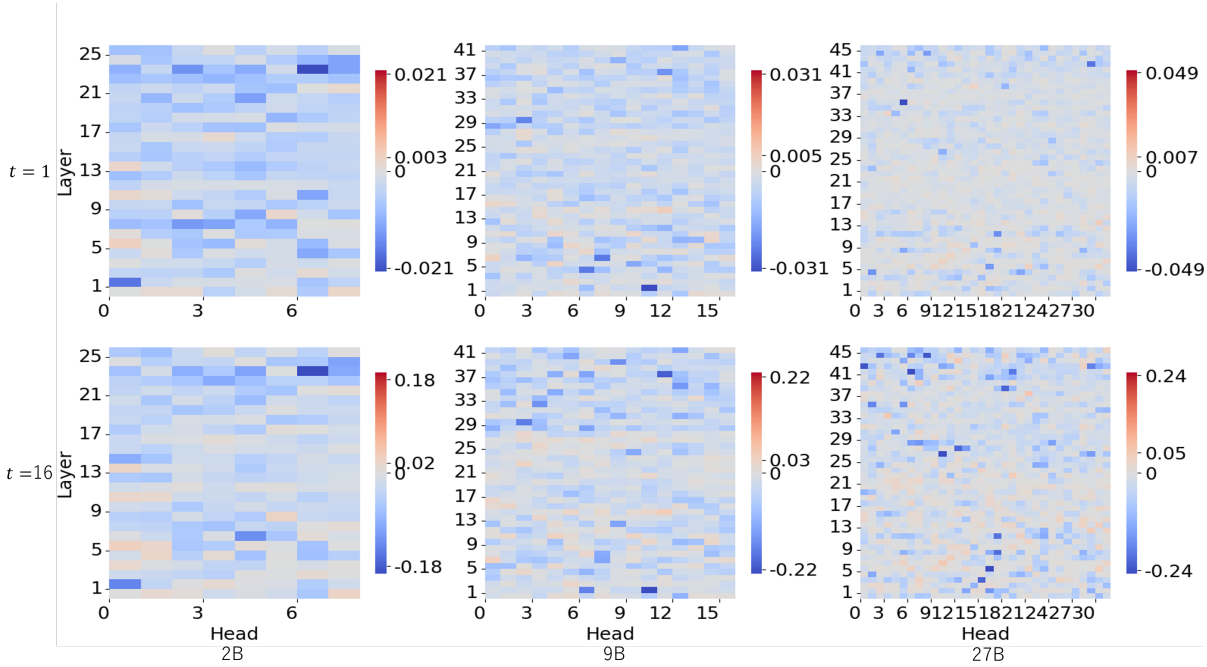

Figure 5: Distribution of Δ_h for each model and each number of typos. The heat map colors are centered around 0, and the tick mark closest to 0 on the positive side of the heat bar represents the maximum Δ_h.

where $D_{\mathrm{KL}}(\cdot)$ is the function that returns the KL divergence, U_m is a uniform distribution over m elements. $P_{x,m,h}$ is the m-th row of the attention map output by head h for the token sequence x. In decoder models, attention scores for the m-th token and each token from the 1st to the m-th token sum to 1. Unlike neurons, for the calculation of typo head identification, we did not narrow down the tokens to calculate and used all tokens in prompts.

Similar to Eq. (2) in neurons, the responsibility score of the heads to the typos is defined as follows:

$$\Delta_h = s_h^{X_{\mathrm{typo}}} - \max\left(s_h^{X_{\mathrm{clean}}}, s_h^{X_{\mathrm{split}}}\right), \quad (4)$$

where X_{typo}, X_{clean}, and X_{split} are the typo, clean, and split datasets, respectively. A larger Δ_h means a head that responds specifically to typos, concentrating on specific tokens in typo inputs, but not in clean or split inputs. The top J heads with the highest Δ_h scores were identified as typo heads.

5.2 Experimental Results

Figure 5 shows Δ_h for $t \in 1, 16$ across all heads in each model. In all models and settings, the differences between the maximum and absolute minimum scores are approximately 10 times. Despite normalization to eliminate dependency on token length in the KL divergence, this result suggests that typo recognition and typo-recovering in the attention layers are not handled by specific heads but are performed using all heads, unlike in neurons.

	2B		9B	
	Clean	Typo	Clean	Typo
Vanilla	1.00	0.86	1.00	0.93
⊖ Random Heads	0.75	0.64	0.60	0.55
⊖ Typo Heads	0.68	0.58	0.94	0.87

Table 2: Accuracy of the word identification task with head ablation (⊖) on clean and typo datasets. "Vanilla" indicates the accuracy without neuron ablation.

5.3 Discussion

Although the experimental results in §5.2 did not provide strong evidence of typo-specific heads, Figure 5 shows some heads have positive values. §4.2 suggest that the wide context is sometimes used to recover typos. We believe that such a wide context is encoded by attention layers. Therefore, we clarify the contribution of typo heads to recovering typos, even if their response is slight. This section discusses the effect of typo heads on the downstream task in detail. Similar to §4.3, the analysis focuses on experiments with the 2B and 9B models in this section.

5.3.1 Head Ablation

Following the approach in §4.3.1, we identified typo heads from 100 randomly selected samples.

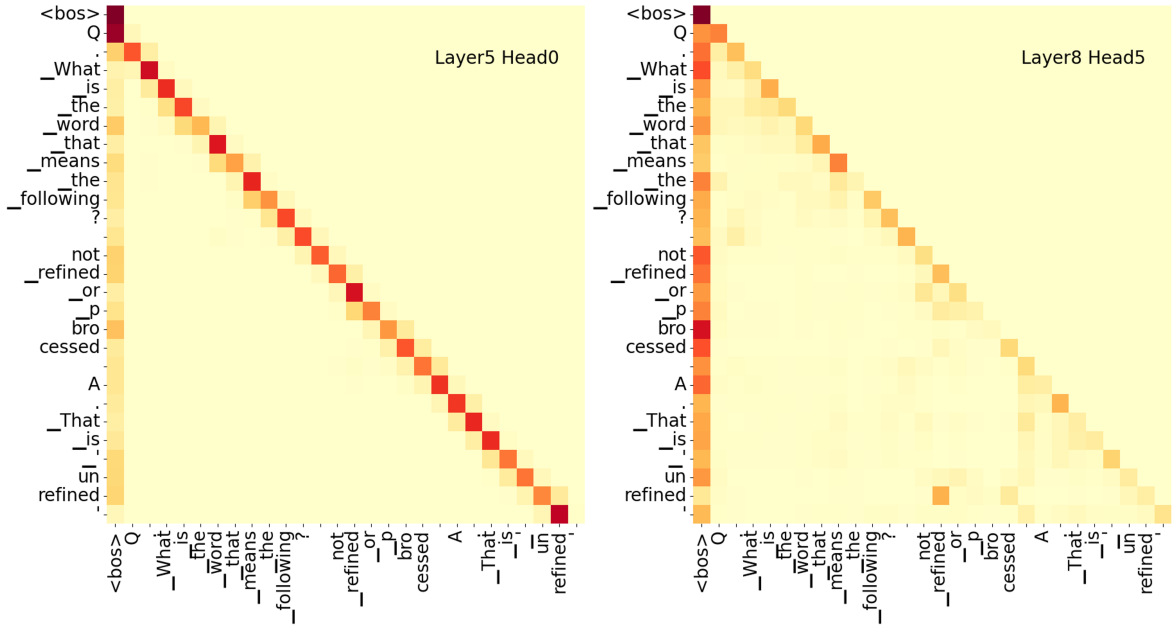

Figure 6: Visualization of typo heads in the 2B model. The word definition in the clean input is "not refined or processed," and the correct answer is "unrefined". The word "processed" was changed with a typo to "pbrocessed."

Then, we ablated these identified heads and measured the accuracy on the remaining 4900 samples. Since the total number of heads is smaller than neurons, we identified the top 2.5% of heads as typo heads. We also randomly selected 2.5% of heads as a baseline. We performed ablation by setting all attention scores of the heads to 0. The experiments were conducted for the clean inputs and the typo inputs with $t = 1$.

Table 2 shows the experimental result. For the 9B model, the accuracy drop caused by the typo head ablation is smaller than that caused by the randomly selected head ablation, regardless of whether the inputs have typos. This indicates that the typo heads in the 9B model react to typos but play less specialized roles. This is a different result from the typo neurons in §4.3.1 and suggests that the heads with the least impact on overall inference are identified as the typo head in the 9B model. In contrast, for the 2B model, which has fewer heads, the ablation of either random or typo heads resulted in a significant drop in accuracy. The accuracy decrease was even greater when typo heads were ablated. This suggests that when the number of heads and parameters is limited, they are actively used for typo-recovering. Furthermore, it suggests that the typo heads are also used for inference with clean inputs like the typo neurons.

From the results, we conclude that larger models do not have heads specifically playing a role in typo-recovering as discussed in §5.2. However, in smaller models, important heads for inference without typos also play a role in typo-recovering.

5.3.2 Visualization of Typo Heads.

To investigate how the typo heads identified in the 2B model in §5.2 behave, we visualize their attention maps of one typo input example in Figure 6. Most typo heads consistently concentrate on the immediately preceding token (left) or always focus on '<bos>' (right). In the head shown on the right, for example, the typo token 'bro' focuses on '<bos>' more than any other token except '<bos>' itself. This suggests that this head uses context aggregated in '<bos>' to recover typos.

6 Conclusion

This paper investigated how the neurons and heads of Transformer-based LLMs respond to inputs with typos. Experimental results show that some neurons perform typo recognition and typo-recovering in the early and early middle layers. Specifically, neurons in the early middle layer are responsible for the core of typo-recovering. Besides, a few heads capturing contextual information also contribute to recovering typos. Although the workload of typo neurons and typo heads differs depending on the model size, our study concludes that it is important to focus on the early and early middle layers for the typo-related analysis.

Limitation

Our analysis was limited to the Gemma 2 model and examined models with sizes up to 27B. Larger models or LLMs with different architectures may have different properties. in §4.3 and §5.3, we have limited our experiments to the 2B and 9B models. For hyperparameters, our experiments were performed only at $t \in \{1, 16\}$. Furthermore, our experiments focused on a specific task, and models may show different properties in a wider variety of tasks. We ran all experiments only once, although there was randomness in applying typos and conducting some experiments. Additionally, in identifying typo heads, we defined them as heads that differ from a uniform distribution across all tokens. As a result, we were unable to find heads specifically responsible for typo-recovering. However, using alternative methods might reveal the existence of typo heads.

References

Mario Almagro, Emilio Almazán, Diego Ortego, and David Jiménez. 2023. Lea: Improving sentence similarity robustness to typos using lexical attention bias. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 36–46.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2019. Identifying and controlling important neurons in neural machine translation. In *International Conference on Learning Representations*.

Steven Bird and Edward Loper. 2004. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain. Association for Computational Linguistics.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. 2024. Tokenization falling short: On subword robustness in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1582–1599, Miami, Florida, USA. Association for Computational Linguistics.

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. 2024. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Joy Crosbie and Ekaterina Shutova. 2024. Induction heads as an essential mechanism for pattern matching in in-context learning. *arXiv preprint arXiv:2407.07011*.

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

Sumit Kumar Dam, Choong Seon Hong, Yu Qiao, and Chaoning Zhang. 2024. A complete survey on llm-based ai chatbots. *arXiv preprint arXiv:2406.16937*.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026, Miami, Florida, USA. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. 2022. Softmax linear units. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/solu/index.html.

Christiane Fellbaum. 2005. Wordnet and wordnets. In Keith Brown, editor, *Encyclopedia of Language and Linguistics*, pages 2–665. Elsevier.

Javier Ferrando and Elena Voita. 2024. Information flow routes: Automatically interpreting language models at scale. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17432–17445, Miami, Florida, USA. Association for Computational Linguistics.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

Jorge García-Carrasco, Alejandro Maté, and Juan Carlos Trujillo. 2024. How does gpt-2 predict acronyms? extracting and understanding a circuit via mechanistic interpretability. In *International Conference on Artificial Intelligence and Statistics*, pages 3322–3330. PMLR.

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, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rhys Gould, Euan Ong, George Ogden, and Arthur Conmy. 2024. Successor heads: Recurring, interpretable attention heads in the wild. In *The Twelfth International Conference on Learning Representations*.

Candida Maria Greco, Lucio La Cava, and Andrea Tagarelli. 2024. Talking the talk does not entail walking the walk: On the limits of large language models in lexical entailment recognition. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14991–15011, Miami, Florida, USA. Association for Computational Linguistics.

Wes Gurnee, Theo Horsley, Zifan Carl Guo, Tara Rezaei Kheirkhah, Qinyi Sun, Will Hathaway, Neel Nanda, and Dimitris Bertsimas. 2024. Universal neurons in gpt2 language models. *arXiv preprint arXiv:2401.12181*.

Tatsuya Hiraoka and Kentaro Inui. 2024. Repetition neurons: How do language models produce repetitions? *arXiv preprint arXiv:2410.13497*.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2024. From tokens to words: On the inner lexicon of llms. *arXiv preprint arXiv:2410.05864*.

Vedang Lad, Wes Gurnee, and Max Tegmark. 2024. The remarkable robustness of LLMs: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*.

J Li, S Ji, T Du, B Li, and T Wang. 2019. Textbugger: Generating adversarial text against real-world applications. In *26th Annual Network and Distributed System Security Symposium*.

Callum Stuart McDougall, Arthur Conmy, Cody Rushing, Thomas McGrath, and Neel Nanda. 2024. Copy suppression: Comprehensively understanding a motif in language model attention heads. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 337–363, Miami, Florida, US. Association for Computational Linguistics.

Marius Mosbach, Vagrant Gautam, Tomás Vergara Browne, Dietrich Klakow, and Mor Geva. 2024. From insights to actions: The impact of interpretability and analysis research on NLP. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3078–3105, Miami, Florida, USA. Association for Computational Linguistics.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Kohei Tsuji, Tatsuya Hiraoka, Yuchang Cheng, and Tomoya Iwakura. 2024. Subregweigh: Effective and efficient annotation weighing with subword regularization. *arXiv preprint arXiv:2409.06216*.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, et al. 2023. Decodingtrust: A comprehensive assessment of trustworthiness in gpt models. *Advances in Neural Information Processing Systems*, 36.

Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. 2021. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. In *Advances in Neural Information Processing Systems*.

Jindong Wang, Xixu Hu, Wenxin Hou, Hao Chen, Runkai Zheng, Yidong Wang, Linyi Yang, Wei Ye, Haojun Huang, Xiubo Geng, et al. 2024a. On the robustness of chatgpt: An adversarial and out-of-distribution perspective. *Data Engineering*, page 48.

Weichuan Wang, Zhaoyi Li, Defu Lian, Chen Ma, Linqi Song, and Ying Wei. 2024b. Mitigating the language mismatch and repetition issues in LLM-based machine translation via model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15681–15700, Miami, Florida, USA. Association for Computational Linguistics.

Weixuan Wang, Barry Haddow, Minghao Wu, Wei Peng, and Alexandra Birch. 2024c. Sharing matters: Analysing neurons across languages and tasks in llms. *arXiv preprint arXiv:2406.09265*.

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

Hongyi Zheng and Abulhair Saparov. 2023. Noisy exemplars make large language models more robust: A domain-agnostic behavioral analysis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4560–4568, Singapore. Association for Computational Linguistics.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Yue Zhang, Neil Zhenqiang Gong, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv e-prints*, pages arXiv–2306.

Terry Yue Zhuo, Zhuang Li, Yujin Huang, Fatemeh Shiri, Weiqing Wang, Gholamreza Haffari, and Yuan-Fang Li. 2023. On robustness of prompt-based semantic parsing with large pre-trained language model: An empirical study on codex. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1090–1102, Dubrovnik, Croatia. Association for Computational Linguistics.