Retokenization
CSE 598 Final Project

Nathan Yap, Yashraj Patel, Paco Haas

Abstract

Research has shown that language models condense token representations into an inner lexicon of word representations during generation. This introduces a question of when this inner lexicon is converted into next-token predictions during inference. Our project investigates this retokenization phenomena through several experiments using logit-lens and the hidden states of language models. We show that certain layers in the latter half of LLAMA-3.1-8B have diverging token representations which are indicative of a language model retokenizing. In addition, we show how the representation of multi-tokens words are promoted and demoted in comparison with individual token logits.

1 Introduction

This project focuses on the transformations of token representation within language model representations. Previous work has shown that transformers decompose token representations into an inner lexicon in order to represent multi-token representations at the last token [6]. They term this process detokenization and show that this process is robust to typos, arbitrary splits, and out-of-vocabulary words.

We seek to explore the retokenization process, where these representations are condensed into token sequences once again. We use logit-lens [7] to view the logit probabilities within of tokens within a word of interest. For example, we would take the word "blessings" into "b", "less", "ings". Then monitor the probabilities of these tokens as each of them are being predicted. This will give us some idea of where the probabilties of each of tokens start diverging with logit-lens.

In addition, we also plot the cosine similarity of the hidden state of a word of interest, to measure where the word representation starts to decline compared to the token representation above.

By observing the representations of each token in this way, we hope to gain insight as to where in the model the retokenization process occurs, and and whether this is consistent for all multi-token words for a model.

2 Related Work

2.1 Logit Lens

A core related work which we utilized widely within our project was the original logit-lens implementation [7]. This article introduce a technique to pass residual stream after each layer within a language model through the unembedding layer to generate output logits. These logits can reveal information about the inner representations within a language model and how token representations develop through subsequent layers.

Specifically for our application, we hope to see how the representations of each token within a multi-token word change and diverge throughout the layers of a large language model. Which makes logit-lens and various other methods inspired by logit lens [2, 4] useful tools for our experiments.

2.2 Patchscopes

The Patchscopes paper [4] extends upon the work done by previous layer-based interpretability methods [2, 7]. They showed that by taking the hidden representation from a generated source prompt at various layers within the model and patching it into the state of a target prompt, they could measure the recovery rate of a certain target token from the source prompt. This provided a more robust metric to determine the representation of certain tokens within a language model's hidden states.

Within this paper, they utilized certain prompts to elicit a specific response from a language model. For example, a prompt "Repeat this word 1) Blessings 2)" was shown to be a consistent method to generate hidden states related to the generation of the multi-token word "Blessings". This method was used in the detokenization paper [6] and was also used within our implementations as well.

2.3 Detokenization

The primary paper that we are working from is "From Tokens to Words: On the Inner Lexicon of LLMs" [6], which works to further the understanding of detokenization as introduced in [3] and [5].

The motivating observation in the paper is that LLMs can tell words from non-words. The authors first create a dataset of nonwords, as shown in Figure 1, by tokenizing words, mixing the tokens, and then putting those tokens back together in random order. As seen from the resulting "nonwords dataset" box in Figure 1, these nonwords still contain tokens from words, but are sampled and ordered in a way that does not create words.

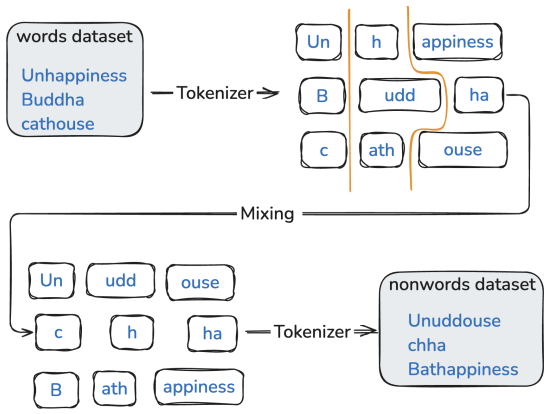


Figure 1: Nonwords Creation Process

Following the creation of the nonwords dataset, the paper then tests the accuracy of the LLM on word versus nonword prediction. As seen in Figure 2, this experiement reveals a 3-stage pattern. In the initial layer, the model performed no better than random guessing, with accuracy less than 50%. In layers 2-6, we see a distinction between words and non-words emerge, with accuracy consistently increasing. Finally, between layers 6 and 20, there is consistent high accuracy, with a performance drop after layer 20.

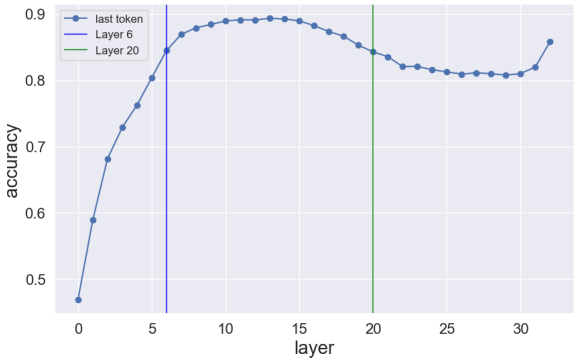


Figure 2: Classification Results

These results indicate that language models internally represent words and nonwords differently. This difference in representation supports the hypothesis that the model performs a detokenization process as it shows that the model holds an internal lexicon of words, rather than just tokens.

The next observation by the detokenization paper is where our question arises. To further show that the language model treats multi-token words as part of an inner lexicon, the authors evaluated the model by computing the proportion of times the model was able to succesfully repeat a multi-token word. As we can see in Figure 3, layers 5-7 are where the model does best as it can repeat the word in a majority of cases, despite never seeing it as input previously. Specifically, layer 7 has the absolute highest retrival rate and will be used for comparison going forward. While these results successfully reinforce that the model treats multi-token words as part of the inner lexicon, the decrease in accuracy after layer 7 suggests another phenonmenon may be happening.

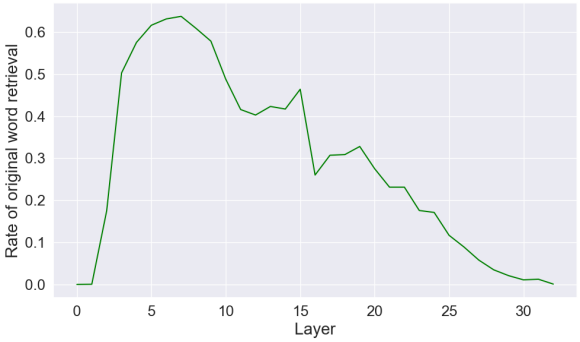


Figure 3: Multi-token Rate of Retrival Results

3 Methodology

3.1 Our work - Retokenization

As shown in the detokenization paper [6] and discussed in subsection 2.3, language models seem to hold an inner lexicon of words. However, Figure 3 shows a decrease in the model's ability to retrive multi-token words in later layers, suggesting that perhaps this inner lexicon decreases in strength.

If LLMs convert tokens to their own inner lexicon, it thereby follows that they should have a method of converting from the inner lexicon back to token representations. This is what we called retokenization and try to discover rigorously through our experiments.

We used the logit lens method to analyze the probabilities of the tokens of multi-tokens words. Specifically, we seperate a word corpus into 2-token words, 3-token words, and 4-token words and analyze each of the logit trends seperately.

We utilized NLTK's library to download all the words from the project Gutenberg version of Jane Austin's Emma, such that we had a list of com-

monly used words. We then split these words into buckets based on the number of tokens they contain.

Using a prompt "Repeat this word. 1){word} 2)", we prompted LLaMA-3.1-8B-Instruct [1] we induced LLaMA to output our expected multi-token word. While the model was generating logits, we use logit-lens on the prediction stream of the last token, such that we could generate inferences for tokens from each layer in the model. We then plot the activations for our tokens of interest and found that the activations only diverge for the tokens within the word at later layers when predicting the first token of a word (see **??**).

We look at the hidden states as well, seeing how well they capture the semantic meaning of the next word, through two main ways. First we analyze the distance of the hidden state to the model's 7th layer representation of the word, inspired by the detokenization paper which found that the 7th layer represents the concept of the word the most. Additionally, we trained multiple k-NN classifiers on the hidden states of the LLM to classify between multitoken and single token words. The motivation behind this, is that if the model represents its inner lexicon before a token the k-NN should be able to distinguish between the two types of words more effectively as there is information encoding the difference in the hidden states. This should demonstrate how the model represents the inner lexicon over tokens throughout the layers. So, there is no bias for the k-NN, we ensure that all multitoken words in our dataset start with a single token word so that the model does not know based off only the token being generated whether it is a multitoken or singletoken word.

4 Results

4.1 Logit Lens Experiments

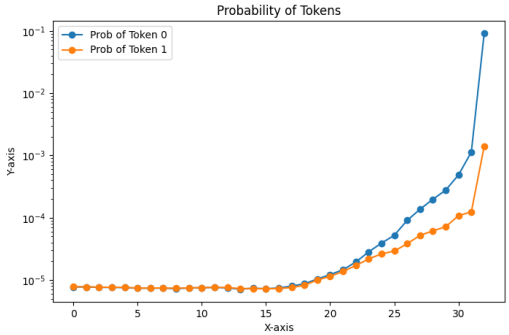


Figure 4: Token proportions of total logit sum

Figure 4 shows the trends of the logits for two token words. Looking at the left plot we see that next token prediction seems to start around layer 17 where the logits for the tokens in the word to be repeated starts increasing. The two tokens for the multi-token word remain relatively similar until the 30th layer. Interestingly, the tokens start diverging as early as the 25th layer until they apparently separate at the last layers with the probability of the both tokens being promoted but the first overtaking the second. In the graph on the right, the ranking of the first token starts diverging at the 23rd layer. This indicates that from around layers 15 to 25 we see the concept of the multi-token word being promoted. Onwards of layer 25 and more intensly at layer 30 onwards, it can be interpreted that the LLM starts the retokenization task, where it turns the concept of the multi-token word into discrete tokens to then output. This interpretation is supported by the rankings graph where at layer 25 the rankings diverge slightly and then more prominently after layer 30. This trend follows for the three token words as seen in the graphs in Appendix B.

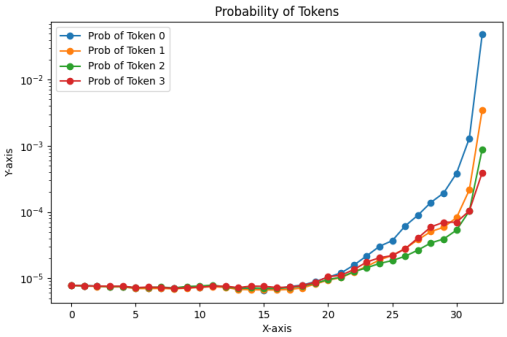


Figure 5: Token proportions of total logit sum

Figure 5 a shows the trends of the logits for the four token words. While there are similarities between the four token words and the two token words, such as the promotion of all the tokens within the word and the separation in the later layers. An interesting phenomena that occurs is that there appears to be two groups of tokens that resolve in the later layers of the LLM (layer 25+). It seems that the beginning half of the word continues to be promoted in the later layers of the LLM to the same degree as the first token of the word, whereas the latter half of the word diverges. The reason is most likely due to the fact that most of the four token words contain prefixes. For example, the word "illegitimacy" which has the prefix of "il". The LLM could be retokenizing the concept of "illegit" at the last layers explaining why the first two tokens are promoting towards the last layers. This is seen in five token words as well, which also consist of a lot of words with prefixes.

4.2 Logit Probability Proportions

Utilizing logit lens, we took a set of words and averaged their probabilities between layers when predicting the first token in the word, generating figure 6. From this plot, we can see that the probabilities of each token remain the same throughout earlier layers within the model, before diverging at layer 22, where the first token starts to be promoted more. This again shows how the model starts deciding on the next-token prediction later in the model after a word representation has been condensed.

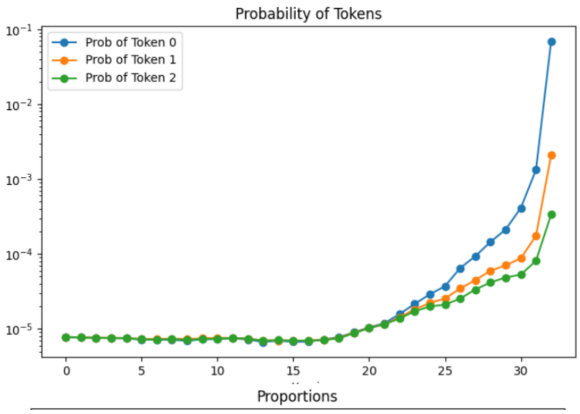


Figure 6: Token probabilities across layers from logit-lens

To better show the relative representation of each token in a 3-token word, we summed up the logits of each token and plotted their relative proportion of the sum in figure 7. This again shows that the 3 tokens in a multi-token word are uniformly represented in earlier layers, before diverging at layer 22 as token 0 is being promoted.

We hypothesize that in earlier layers, the tokens are equally likely due to the representation of the word being more prominent than individual tokens. Then at layer 22, the model transitions into promoting individual tokens in order to generate a next token prediction based on the latent representation of the word produced until then.

4.3 Word Hidden State Comparison

In order to determine when the "inner lexicon" or word representation of each word was most represented, we utilized results from the detokenization paper [6] which showed that the hidden state representation from layer 7 of LLAMA models had the highest recovery rate when using the patchscopes method (see figure 3 for the paper's plot). Based on this, we took the hidden state at layer 7 for each multi-token word in a subset, then took the cosine similarity of layer 7 with each hidden state while generating each individual token of the given multi-token word. The plot of the cosine similarities is shown in figure 8.

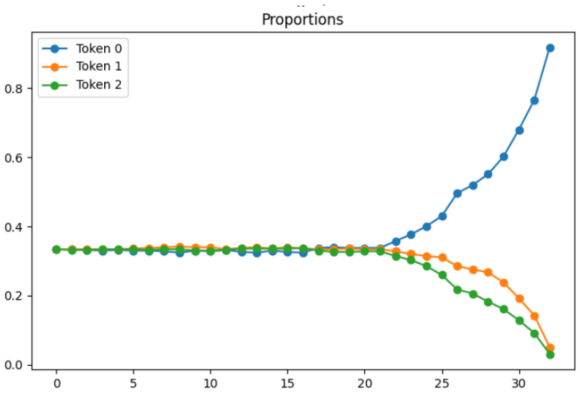


Figure 7: Token proportions of total logit sum

Figure 8 shows that the word representation is closer to each individual token-representation in earlier layers, with the similarity peaking at around layer 17. Before the representation diverges and starts to decrease until the last layer.

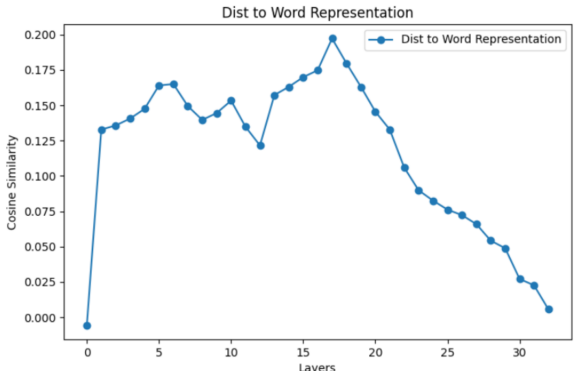


Figure 8: Cosine similarity of token hidden states and layer 7 representation

Comparing this to the logit proportions from the earlier section (see figure 9), we can see that the word representation is promoted the most before decreasing as the token probabilities start to diverage at approximately layer 22. We believe that this supports our hypothesis that the model uses some layer within the model to convert from a conceptual representation to a next token prediction. As we can see that the word representation starts to decrease in prominence slightly before logit probabilities diverge.

4.4 KNN Experiments

Figure 10 displays the result of our k-NN classifier tests. The classifier performed no better than a random guess at around 50% correct in the first layer. This suggests that the initial embeddings contain very little or no information about the multi-token vs. single-token structure of words.

Immediately after the first layer, though, we have a sudden spike in accuracy to nearly 100%.

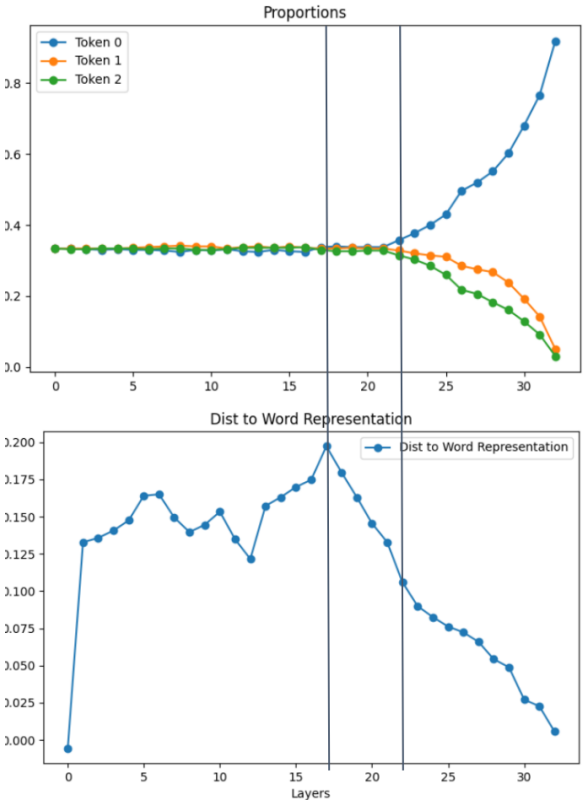


Figure 9: Comparing the logit proportions of each token compared to the distnace from the word representation

The classifier is nearly perfectly capable of distinguishing single-token words from multi-token words on a test set of over 2000 examples. That is, even in the very first few layers of the model, the network rapidly begins encoding information about not just the next token but also the whole word.

There is a slight downward trend in accuracy between layers 15 and 25, although general performance is still extremely high. This slight decrease may be due to the model starting to pay more attention to the next token prediction task rather than trying to understand which next word to predict.

These findings suggest that large language models (LLMs) are extremely sensitive to words vs tokens as soon as they encounter them. Rather than immediately tokenize the word and then conceptually understand the entire word, the model immediately attempts to next word prediction over next token prediction. Perhaps most strikingly, is that such information is embedded so early as to establish that LLMs build their knowledge on the foundation of word-conscious features.

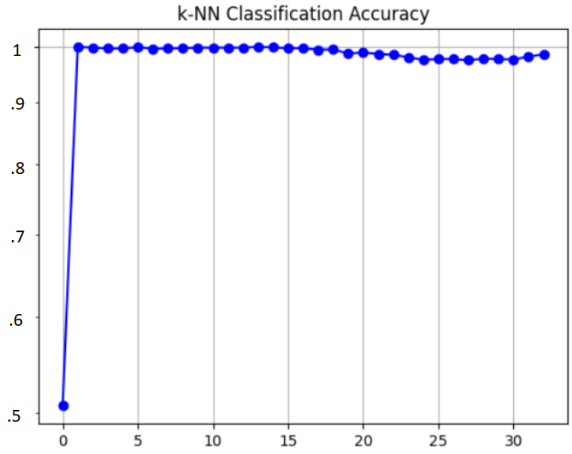


Figure 10: KNN Performance when distinguishing between multi-token and single-token words

5 Discussion

5.1 Limitations

5.1.1 Hidden State Comparisons

In order to compare the hidden representations between layers, euclidean distances and cosine similarity between the tensor representations from layers of interest. It has been shown that there is a linear representation of high-level concepts within language models to some degree [8], which is what motivated our usage of cosine similarity to compare hidden states. However, these are not guaranteed to be true measures of similarity between states. Nevertheless, we still believe that the hidden state euclidean distances can still be interpreted as some form of similarity between token representations, though it may not be perfectly precise in reality.

5.1.2 Token Significance Metrics

We use the proportion of total word probability as a measure of relative token significance. For example, a multi token word "Blessings" would have output logits for "b", "less", and "ings", and we would measure the relative significance of each token with their logits' proportion of the total logit probability of the word. Throughout our experiments, we saw that the relative representation of these tokens was uniform up until later layers, which we believe indicates that certain tokens are promoted in these later layers, however this metric could also be more robustly evaluated.

5.2 Contributions

We believe that understanding the process of retokenization could influence various domains in the field of language models.

We believe that having an understanding of the types of representations present in the layers of a model (token versus conceptual representations) is foundationally important for the interpretability of

language models. By understanding the way that models transition from tokens to concepts to tokens again, we can have a better of idea of how language models are able to reason or condense concepts layer by layer.

In addition, if there is a better understanding of what layers contribute to retokenization, model architectures and training processes could be optimized to support this process. For example, if the later layers are shown to be mainly utilized for retokenization, then it might be possible to freeze these layers during finetuning or training for specific tasks. As these layers would theoretically have less impact on general reasoning and conceptual representations.

6 Future Work

6.1 Logit Lens Alternatives

There are many layer-based interpretability metrics which are more robust than Logit Lens (such as Patchscopes or tuned Logit Lens [2, 4]). These could be used to more concretely demonstrate the trends of hidden token representations in our layer-based experiments.

6.2 Beyond Single-Word Tokens

These experiments with detokenization and retokenization have all been in the context of single/multi-token words. However, the same concepts could be applied to phrases or sentences to further explore the relationship between inner representations and tokens.

6.3 Extend to More Models

Our experiments focused on Llama-3.1-8B, running on other models would allow us to generalize our findings. Within the detokenization paper [6], they found that the inner representations of words varied widely between model architectures, and we chose to utilize Llama models due to their consistent representations between layers. Therefore, we may find that other models have differing trends when it comes to token vs conceptual representations.

6.4 Theoretical Justification

We focused on experimental results and while our results make intuitive sense, we have yet to provide strong mechanistic explanations for the results. Potentially we could show how word and token representations diverge more robustly through further experiments.

7 Conclusion

Our experiments show promising results indicating that retokenization occurs within language models. As a model will convert from a conceptual word representation to a token representation at a specific layer within the model. This was supported through experiments with logit-lens and comparing hidden state representations to track the "inner lexicon" or word representation between layers. There are multiple avenues which could be used to further support evidence of the retokenization phenomena, and better understanding of retokenization could possibly have substantial impact on the way we interpret language models and how they are engineered as well.

References

[1] et. al Aaron Grattafiori. The llama 3 herd of models, 2024.

[2] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2023.

[3] 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. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

[4] Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models, 2024.

[5] Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*, 2023.

[6] Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. From tokens to words: On the inner lexicon of llms, 2025.

[7] nostalgebraist. interpreting gpt: the logit lens, 2020.

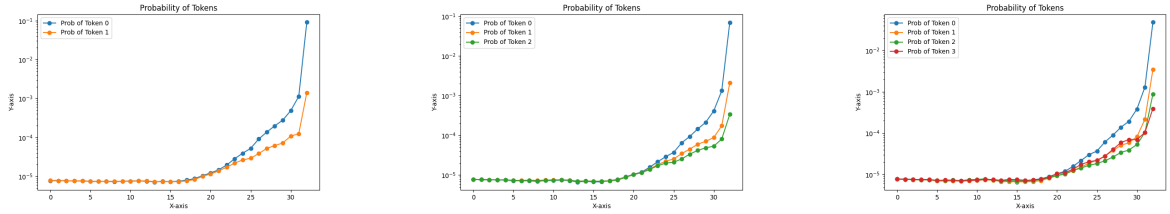
[8] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024.

A Code Respository

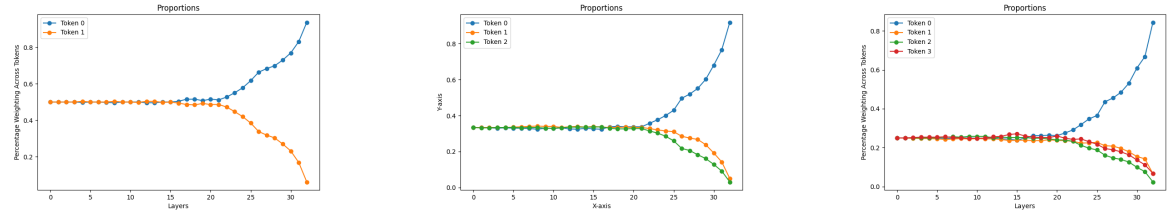
See our GitHub repo for our code implementation.

B Additional Plots

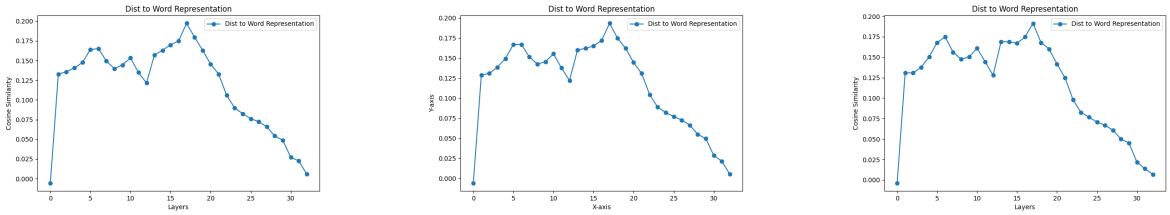
All logit-lens plots for various numbers of tokens per word are shown below. Words with 2 through 4 tokens are shown in figure 11.



(a) Probabilities of Tokens



(b) Proportion of Probabilties of Tokens in Word



(c) Distance to Word Representation

Figure 11: Comparison of probabilities and word representations for different token counts