Interpreting Tiny Transformers

Florian Micliuc[1], Martin Kirkegaard[1], Markus Sibbesen[1], Philip Winstrøm-Jespersen[1]

[1]{flmi | marki | mksi | phwi}@itu.dk
Course Code: KSANLPD1KU

Abstract

Transformer-based Large Language Models (LLMs) have shown remarkable performance across different domains. How the LLMs work is, however, still largely a mystery. We investigate the attention mechanism in a small transformer-based language model, TinyStories-1M. This model is trained exclusively on short, simple texts, and has remarkable abilities to produce grammatically correct text, despite its small size.

We train classification probes that predict Part-Of-Speech tag (POS-tag) on key and query vectors of all attention heads in the model and find that their performance varies considerably across heads. Based on these results, we focus our investigations into heads, for which the corresponding probe's performance is particularly good, to figure out when the head is active and what it attends to. We identify (1): a quotation head, which is active inside a quotation and attends to quotation-marks. (2): a past tense verb predictor head, which is active on tokens immediately preceding a past tense verb, and attends back to previous past tense verbs. (3): a definite noun predictor head, which is active on possessive pronouns and definite articles and attends back to preceding nouns.

We are not certain of the precise function of the attention heads we identified and caution against premature conclusions and interpretations. We are likewise unsure if these results generalize to larger models. We are, however, hopeful, that research into smaller models like TinyStories-1M will eventually contribute to a deeper understanding of the large models, which play an increasing role in our lives and society. All code can be found on our GitHub.

1 Introduction

Large Language Models (LLMs) have shown a remarkable performance on different tasks, spanning from logic and mathematics to more creative domains, simply by predicting the next token, as seen in Malach (2024).

However, it is still largely unknown how LLMs work and why do they behave as they do. However, state of the art models require a significant amount of computing power to be ran[1], which can often limit independent researchers to conduct interpretation studies.

drTherefore, in this paper, we aim to investigate deeper on how a small transformer model works, such as TinyStories-1M, in an attempt to extract attention patterns that could lead to a better decision interpretation. We expect this outcome, as the model has been trained on a low-complexity data, which can make the attention mechanism prioritize basic linguistic features over intricate semantic knowledge. To accomplish this, we are running a simple experimenting pipeline which involves probing, to identify attention heads that contain the representation of a POS-Tag and then explore when the head is active and what it attends to, attempting to figure out its function.

Furthermore, Bricken et al. (2023) showed that sparse autoencoders could be used to identify features in a smaller neural model. What this paper aims to do is to create a setup that can identify certain properties of a much smaller transformer based language model from TinyStories with 1M parameters, in the hopes that the methods can be scaled to larger models, similarly to what they did in Templeton et al. (2024).

2 TinyStories Model and Dataset

Eldan and Li (2023) presents a suite of simplistic models with varying parameter count ranging from 1M to 33M parameters, and has also released the dataset used to train said models. The dataset is called TinyStories and is synthetically generated using GPT-3.5 and GPT-4, prompted to write

[1] Llama 3.1 Requirements

a children's story understandable to a 3-year-old. The primary objective of the TinyStories dataset is to produce text that incorporates key language features, such as grammar and vocabulary, while remaining simple and straightforward. The combination of a small parameter count and simple training data limits what the model encodes, resulting in a model remarkably competent at generating grammatically correct text for its size. We hope this will also allow us to more easily find key linguistic features.

Eldan and Li (2023) themselves report good results for interpretability, finding that in a model with only one transformer block trained on the TinyStories dataset, certain attention heads exhibited clear *semantic* based attention, for example attending to names in the sentence, while others exhibit *distance* based attention, attending a fixed number of tokens back. We will focus our more thorough analysis on the TinyStories-1M model, which has a more standard architecture of 8 layers and 16 heads in each layer, focusing specifically on heads whose function can be interpreted linguistically.

Since the data used to train the TinyStories models is so specific, we thought the most suitable data to use when interpreting the model was the validation set from the same TinyStories dataset.

3 Related Work

Previous research have show certain linguistic features that are computed in the attention mechanism of the LLM BERT, including the relationship between transitive verbs and associated direct object and co-referent mentions and their antecedents Clark (2019), using, among other methods, classification probes on individual attention heads. This is an automated and quantifiable way to extract meaning from individual layer of a model, more on this in section 4.1.

There are several other approaches for extracting attention, from non-linear classifiers White et al. (2021) to methods focusing on cross-head interaction of multi-head attention. Kang et al. (2024) In this project we will focus on a linear classifier for extracting attention within the heads. Limitations of our approach will be further assessed in the discussion 6.

4 Methodology

We use three different but complimentary methods to analyze the attention mechanism of the model. The first method uses linear probing to narrow the scope to a few interesting attention heads, the second method measures how much the head contributes to the residual stream at different tokens, to find where the head is active, and in the last method, we qualitatively inspect the attention pattern in hopes of explaining what the head does.

4.1 Identify important heads

As described by Alain (2016), the objective of a *classification probe* is aimed at determining where specific information is encoded within a model. By training a simple model (the probe) to effectively classify a feature based on certain model components - such as activations from a hidden layer or an attention vector - it can be inferred that these components encode the corresponding feature. In our project, we focus on identifying attention heads that encode POS tags, using classification probes trained on the *key* and *query* vectors. Accordingly, for each layer and head of TinyStories-1M, we train a multinomial logistic regression probe

$$p_k(\mathbf{x}) = \mathrm{softmax}_k(W\mathbf{x} + \mathbf{b})$$

on key and query vectors $\mathbf{x} \in \mathbb{R}^{d_k}$, for K different POS-tags. To fit the weight matrix $W \in \mathbb{R}^{K \times d_k}$ and bias term $\mathbf{b} \in \mathbb{R}^K$, we minimize the expected *cross-entropy loss*:

$$\mathcal{L} = -\sum_{k=1}^{K} [y_i = k] \log(p_k(\mathbf{x_i}))$$

for a given activation $\mathbf{x}_i$ with label $y_i \in \{1, \ldots, K\}$ (converted from POS-tag to an integer).

As our data did not come with POS-tags included, the ground truth was tagged by the POS-tagger from the `nltk` library[2].

The probe results indicate which heads encode the most information about POS tags, enabling us to identify heads with a high potential for further analysis. By training the probes on both the keys and the queries, we can analyze from two perspectives. If the key-probe demonstrates high performance in classifying whether a token corresponds to a specific POS tag, it indicates that the specific

[2] NLTK Repository

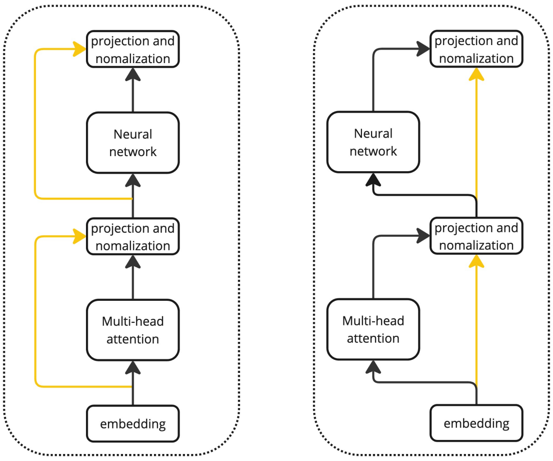

projection and nomalization
Neural network
projection and nomalization
Multi-head attention
embedding
projection and nomalization
Neural network
projection and nomalization
Multi-head attention
embedding

Figure 1: Two transformer drawings, left figure showing the classical interpretation of a transformer, where each block is the main part of the information flow. The right drawing has the "new" way of thinking with the residual stream (yellow arrow) as the main information flow.

attention head tries to retrieve information from the token with that POS tag. In contrast, high performance in the query-probe implies that the attention head seeks information that is influenced by the token's specific POS tag. As a result, we use both sets of results to qualitatively identify and examine the most promising heads.

4.2 Measuring Contribution of an Attention Head to Residual Stream

When transformers were first introduced by Vaswani et al. (2017) they illustrated the transformer more akin to figure (1, left) with the residual stream not being the main information flow of the model, but the "blocks" themselves. Elhage et al. (2021) proposes a way of thinking about transformers in which the residual stream is the main information flow and the different blocks *filter* information in the stream.

We measure the *magnitude of the contribution* of a head to the residual stream, for different input tokens. As an example, if the output of an attention head contributes a lot to the residual stream when the input token is a verb, we reason that the information attended to and from verbs is particularly important for the head. We say the head is highly *active* on verbs.

For head_i, we compute its contribution[3] ($c_i \in \mathbb{R}^{d_{model}}$) to the Multi-Head Attention (MHA) output for each token in a sequence:

$$c_i = \text{head}_i W^O_{[start_i\,:\,end_i,\;:\;]}$$

with $start_i, end_i$ being the start and end indices of head_i in the concatenated MHA. c_i is then the component of a multi-head attention block's output, which head i is responsible for. $W^O \in \mathbb{R}^{h \cdot d_k \times d_{\text{model}}}$ is the weight matrix that filter the MHA into the residual stream. For TinyStories-1M which has $h = 16$ heads, each with size $d_k = 4$, $d_{model} = h \cdot d_k = 64$.

We then use the euclidean norm of c_i:

$$\|c_i\|_2,$$

to measure the magnitude of head_i's contribution. This should show how "much" the corresponding head is writing to the residual stream, and thereby give us a quantifiable proxy of importance. We find that this measure varies considerably depending on input token in predictable ways.

4.3 Inspecting the Attention Patterns

In a Transformer-based model, the key-query interactions plays a critical role in the self-attention mechanism, which allows the model to assess the importance of different tokens in a sequence in relation to one another. This importance measure across tokens in the sequence is called the *attention pattern* and is calculated as the dot products of the current tokens query vector and all the previous tokens' key vectors. The head gathers information from previous tokens according to this pattern, so by inspecting which tokens the head attends to, both qualitatively at specific example sequences and quantitatively over multiple sequences, we can hopefully get an idea of what kinds of information is important for the head.

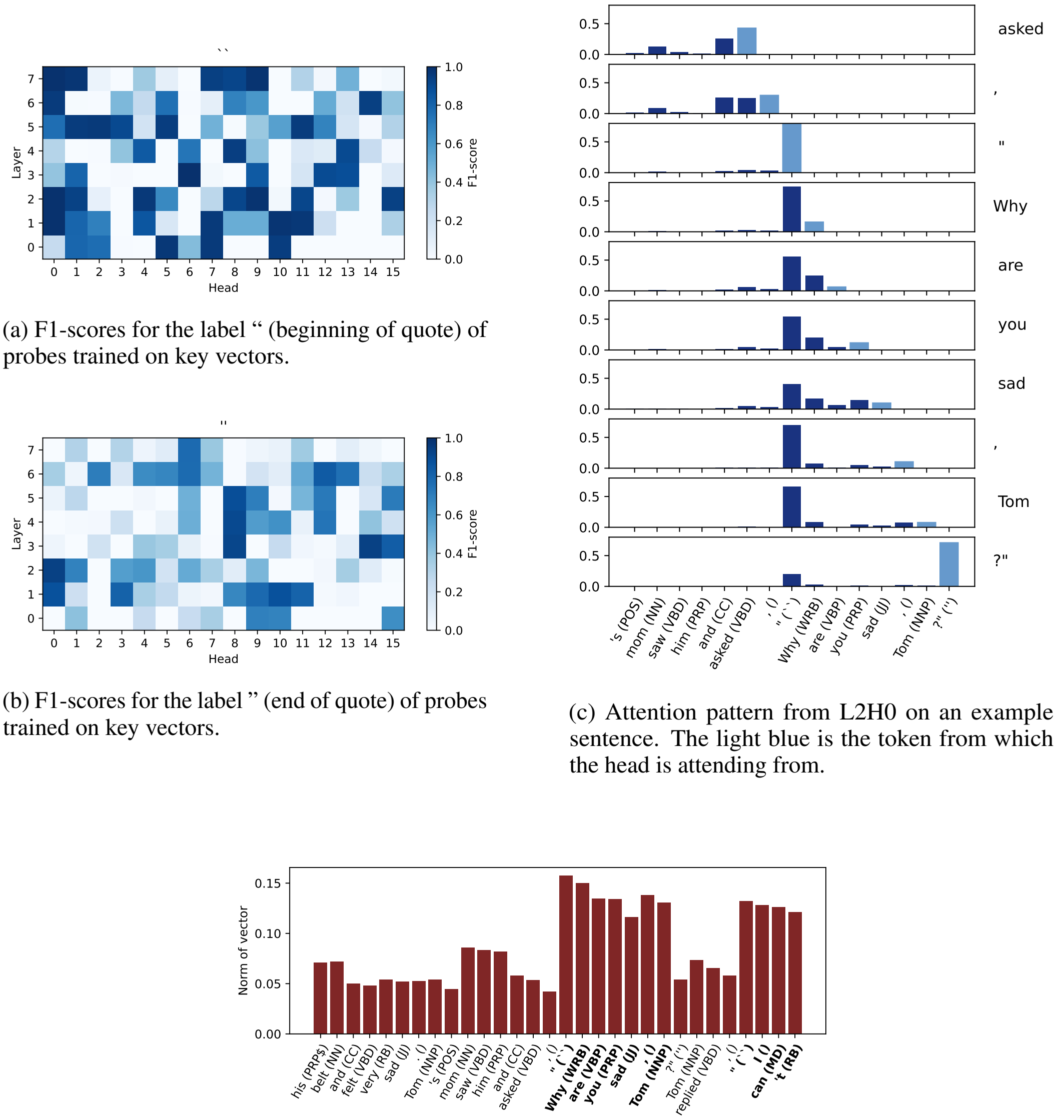
5 Results

We train key and query probes on all heads of all layers and compute their F1-scores (results for key-probes can be seen in Appendix A and query probes in Appendix B). We see that the F1-scores vary depending on POS-tag and layers. For example, conjunctions (CC), are easily identifiable from many heads, but mostly in early layers and others, like modal verbs (MD), are identifiable only from a few select heads.

5.1 Case study 1: Quotations

We assess the performance of the key-probes on the POS-tags “ (beginning of quote), shown in figure 2a, and ” (end of quote), shown in figure 2b,

[3] We were unable to figure out if this vector had an established name.



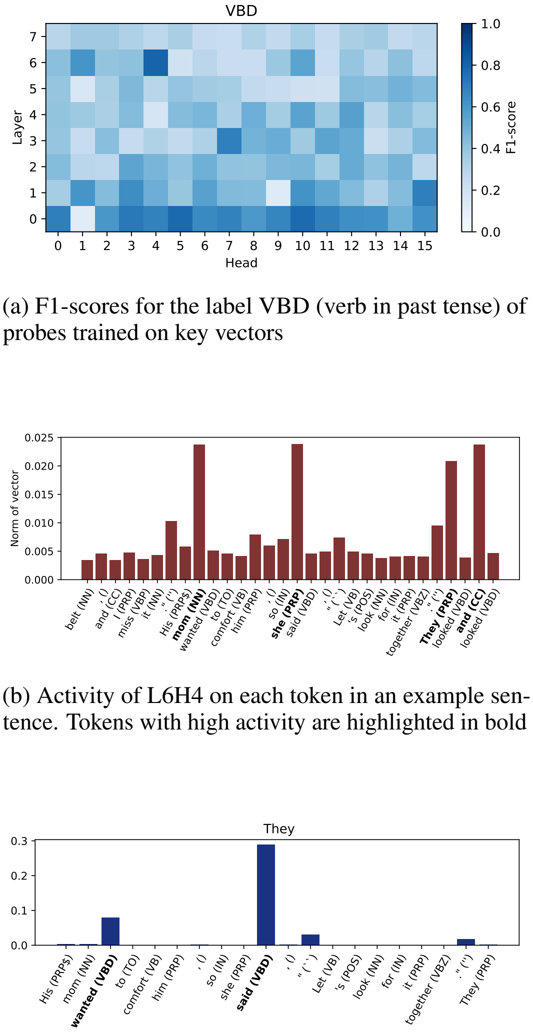
(a) F1-scores for the label “ (beginning of quote) of probes trained on key vectors.

(b) F1-scores for the label ” (end of quote) of probes trained on key vectors.

(c) Attention pattern from L2H0 on an example sentence. The light blue is the token from which the head is attending from.

(d) Activity of L2H0 on each token in an example sentence, with the bold text highlighting the inside of quotation marks.

Figure 2: Case study 1: L2H0 looks at attends to begin and end quotes, and is active within a quotation.



(a) F1-scores for the label VBD (verb in past tense) of probes trained on key vectors

(b) Activity of L6H4 on each token in an example sentence. Tokens with high activity are highlighted in bold

(c) Attention pattern from L6H4 on an example sentence, for the query of the last token in the sequence (They). Token highly attended to are highlighted in bold.

Figure 3: Case study 2: L6H4 is active immediately preceding a verb in past tense and attends back to the most recent previous verbs in past tense.

and find that L2H0[4] performs well on both. This suggests that it attends highly to both the beginning of quotation and the end of a quotations.

When observing the activity of L2H0, we find that its contribution to the residual stream is high while inside a quotation, but drops off sharply afterwards (see figure 2d for an example sentence).

We inspect the attention pattern of L2H0 (figure 2c) and find that while inside a quotation, it attends highly to the start-quotation mark, until the end, when it shifts to the end-quotation mark.

5.2 Case Study 2: Verbs in Past Tense

We inspect the key-probes and find that the performance in classifying the tag VBD (verbs in past tense) is high when trained on the early layers, but drops off gradually through the model (see figure 3a). There is, however, one notable exception: L6H4 exhibits a high F1-score, so we investigate it.

We find that the activity of L6H4 is high immediately before a verb in past tense (see figure 3b). At those points, it attends back to previous nouns (see figure 3c).

5.3 Case Study 3: Definite Nouns

We inspect the query-probes that do well in classifying the tag PRP$ (possessive pronouns), shown in figure 4a and after further qualitative investigations, we find that one of them, L5H5, exhibits an interesting pattern.

We see that the attention head is particularly active on possessive pronouns (as expected) and the definite article, always right before a noun (see figure 4b).

On these tokens, the head attends to the most recent other nouns (see figure 4c). Additionally, we computed the average attention from possessive pronouns to other parts of speech across 100 sequences (see Appendix C) and observe that possessive pronoun overwhelmingly attend to nouns. Similarly, we also compute average attention to nouns from other POS-tags, and find that nouns are mostly attended to by determiners and possessive pronouns (see Appendix D).

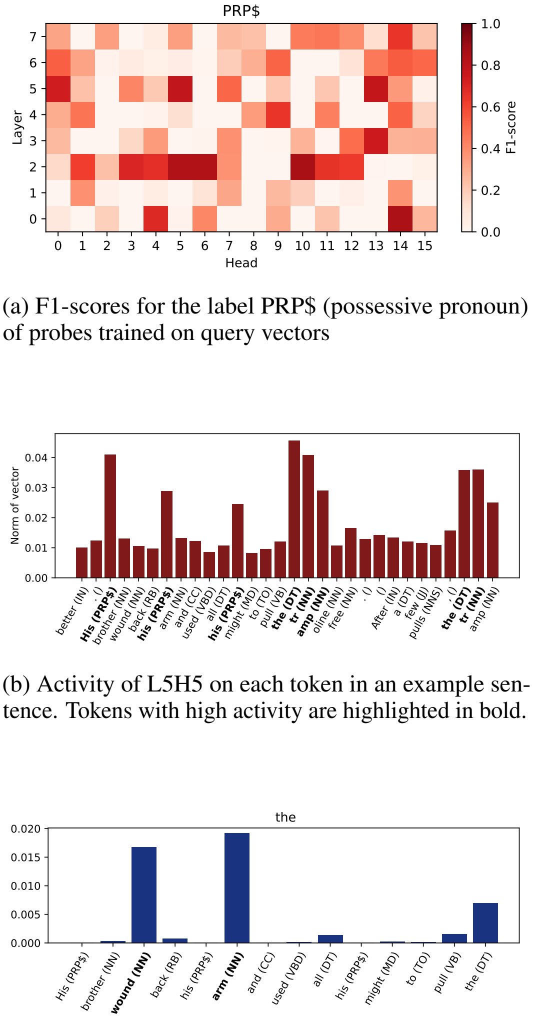
6 Discussion

Based on research done on both small and large language models, Olsson et al. (2022) propose the concept of generalized *induction head*, which they hypothesize is responsible for a large portion of in-context learning in language models. It implements a simple algorithm

```
[A*][B*] ... [A] -> [B]
```

From the current token `[A]` the head looks back at a previous occurrence of a similar token `[A*]`, attends to the *following* token `[B*]`, copying its information, increasing the probability of prediction a similar token `[B]` as the next token. Here, similarity is a wide concept, and as models get bigger,

[4] We use the same notation as McDougall et al. (2024), where L2H0 means head 0 of layer 2



(a) F1-scores for the label PRP$ (possessive pronoun) of probes trained on query vectors

(b) Activity of L5H5 on each token in an example sentence. Tokens with high activity are highlighted in bold.

(c) Attention pattern from L5H5 on an example sentence. Tokens highly attended to are highlighted in bold.

Figure 4: Case study 3: L5H5 is particularly active on possesive pronouns and definite articles (that is, right before a definite noun). From here, it attends back to previous nouns.

they could implement increasingly granular and specific conceptions of similarity.

We mention this because two of our case studies look strikingly similar to this general formula:

- In Case Study 2, L6H4 is active immediately before past tense verbs, and attend back to past tense verbs.
- In Case Study 3, L5H5 is active immediately before definite nouns and attends back to nouns.

Our findings introduce the possibility that L6H4 and L5H5 play a part in implementing a type of part-of-speech based induction algorithm. We have, however, not inspected which other heads feed into them (induction heads need help from additional heads earlier in the model to function, see Elhage et al., 2021), nor have we assessed the effect of each head on the output (they might, for all we know, implement a form of *copy suppression*, as described by McDougall et al., 2024). More research is definitely needed before any clear conclusion can be drawn.

An obvious limitation for our research is inaccuracies in our ground truth POS-tags. The POS-tagger used is not perfect (in `nltk`'s documentation, they report a 0.74 accuracy), which can be seen in figure 4c. Here, the tagger has tagged the token "wound" as a noun (NN), however in this context should be past tense verb (VBD). This could interfere with the usefulness of the probe approach described in section 4.1 as a perfect probe would still have a margin of error due to incorrect tags. Curiously, in the above mentioned example, the attention head in question seems just as "confused" as the tagger, attending to the verb as if it were a noun.

This leads is to another limitation: The TinyStories models and datasets. Standard English written and spoken by adults has a lot more complexity, longer stories, and a wider vocabulary than the 10.000 most frequent English tokens used by TinyStories Eldan and Li (2023). We do not know if our findings can generalize to larger models, trained on standard text.

For this report we train a linear classifier on the key-query matrix in order to find the important heads, but we do not look at the value vector. The value vector however is crucial for the attention mechanism (the contribution to the residual stream is in part determined by the value vectors). Ignoring the value vector may then lead us to missing an important head.

A large part of our findings in section 5 are of a qualitative nature. This is in stark contrast to much of modern NLP. However, as Olah and Jermyn (2024) mentions, truly new emerging fields, such as mechanistic interpretability, often have a comparatively high ratio of qualitative research, with the ratio declining as the field matures. Although our results are not as certain as if we had statistical arguments, we believe we observe what Olah and Jermyn (2024) describes as *signal of structure*, patterns that seem to conspicuous to be random. This indicates to us, that this direction of research may be fruitful.

References

Guillaume Alain. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Kevin Clark. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english? *Preprint*, arXiv:2305.07759.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. https://transformer-circuits.pub/2021/framework/index.html.

Hankyul Kang, Ming-Hsuan Yang, and Jongbin Ryu. 2024. Interactive multi-head self-attention with linear complexity. *Preprint*, arXiv:2402.17507.

Eran Malach. 2024. Auto-regressive next-token predictors are universal learners. *Preprint*, arXiv:2309.06979.

Callum 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.

Chris Olah and Adam Jermyn. 2024. Reflections on qualitative research. Transformer Circuits Thread. Accessed: 2024-12-19.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jennifer C. White, Tiago Pimentel, Naomi Saphra, and Ryan Cotterell. 2021. A non-linear structural probe. *Preprint*, arXiv:2105.10185.

Appendix

A Probe-keys results

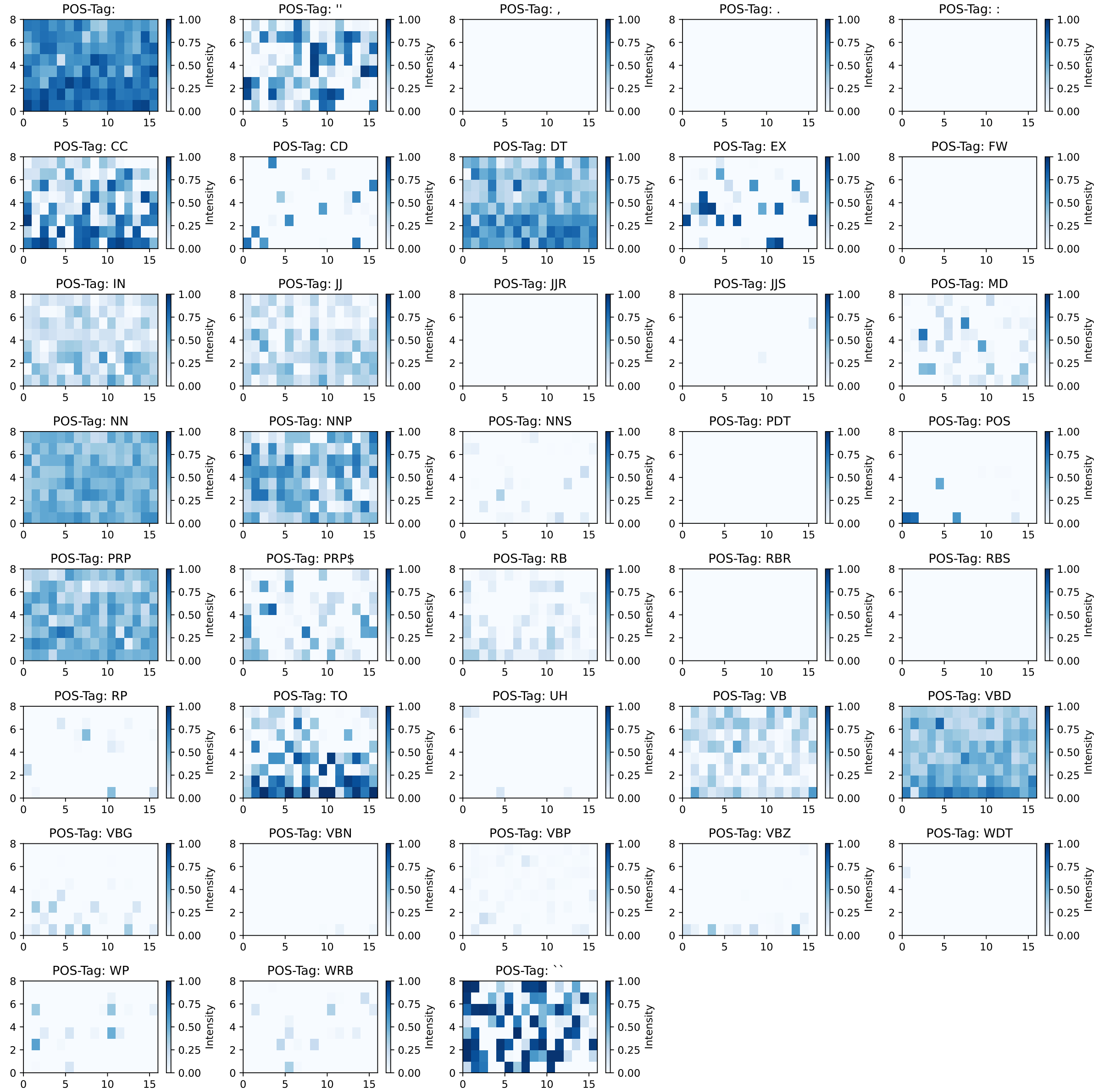


Figure 5: All probe results for key vector. The Y-axis for each plot is the layer index and the X-axis is the head index. The intensity described is the F1-score for a given probe that specific POS-tag.

B Probe-queries results

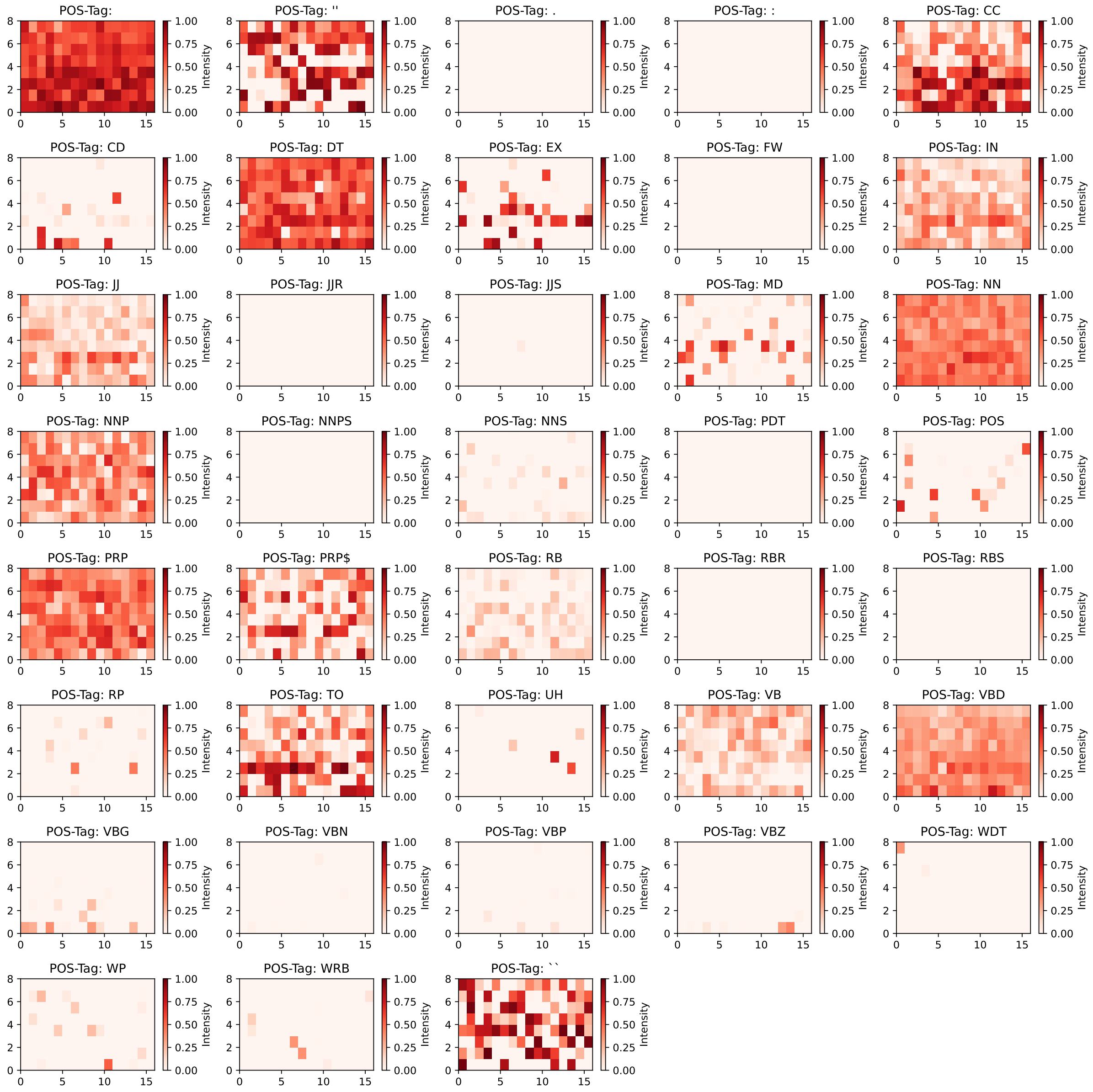


Figure 6: All probe results for query vector. The Y-axis for each plot is the layer index and the X-axis is the head index. The intensity described is the F1-score for a given probe that specific POS-tag.

C Averaged Attention from POS-Tag

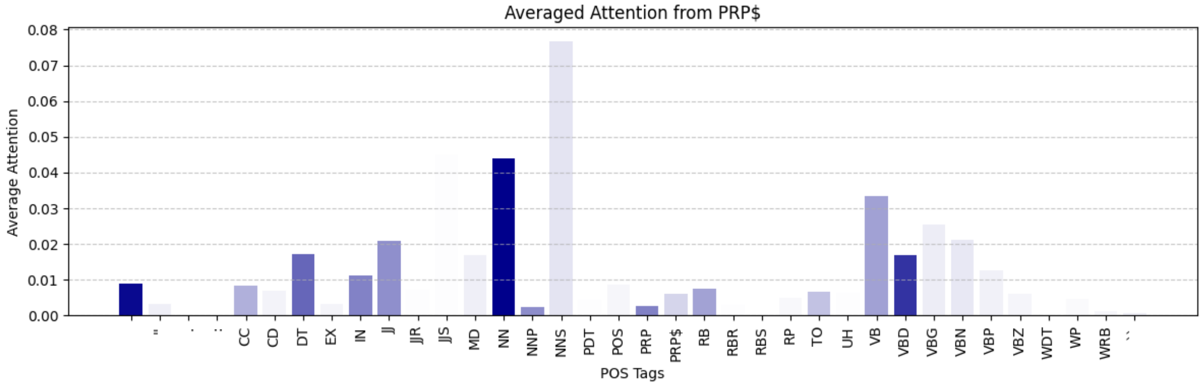


Figure 7: The plot above showcases the attention pattern from the POS-Tag PRP$ over 100 sentences. The color intensity represents how frequent the POS-Tags are in the 100 sentences. The height represents the average attention given to a specific POS-Tag.

D Averaged Attention to POS-Tag

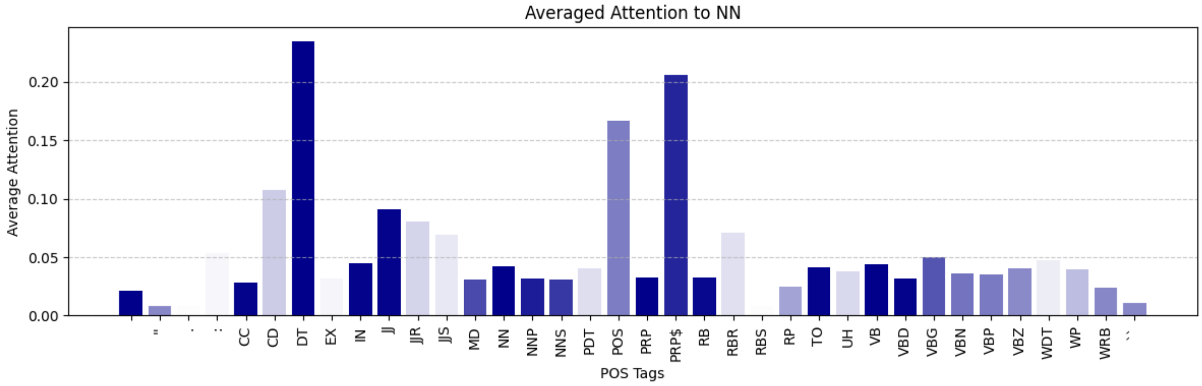


Figure 8: The plot above shows the attention pattern to the POS-Tag NN over 100 sentences. The color intensity represents how frequent the POS-Tags are in the 100 sentences.The height represents the average attention given to a specific POS-Tag.

Contribution Statement

We all contributed equally throughout the project.

AI usage

Assistance purely with the language of the paper

Some parts of the report has ben corrected using Grammarly or ChatGPT.

Short-form input assistance

We have not used any AI-tools for this.

Literature search

A minuscule amount, the vast majority was found using Google scholar.

Low-novelty text

We have not used any AI-tools for this.

New ideas

We have not used any AI-tools for this.

Codebase

Smaller parts of the codebase have been assisted using AI-tools, particularly for debugging and plotting.