A.3.2 Multi-subject mood stories circuit - Pythia 2.8b

We also examined the circuit for this sentence template: Carl hates parties, and avoids them whenever possible. Jack loves parties, and joins them whenever possible. One day, they were invited to a grand gala. Jack feels very [excited/nervous]. We did not attempt to reverse-engineer the entire circuit, but examined it from the perspective of what matters causally for sentiment processing–especially determining to what extent summarization occurred.

Following the same process as with GPT-2 with preference/sentiment-flipped prompts (that is, taking x_{orig} to be "John hates parties,... Mary loves parties," and $x_{flipped}$ to be "John loves parties,... Mary hates parties"), we initially identified 5 key heads that were most causally important to the logit difference at END: 17.19, 22.5, 14.4, 20.10, and 12.2 (in "layer.head" notation). Examining the value-weighted attention patterns, we observed that the top token receiving attention from END was always the repeated name RNAME (e.g., "John" in "John feels very") or the "feels" token FEEL, indicating that some summarization may have taken place there.

We also observed that the top token attended to from RNAME and FEEL was in fact the comma at the end of the queried preference phrase (that is, the comma at the end of "John hates parties"). We designate this position COMMASUM.

Multi-functional heads Interestingly, we observed that most of these heads were multi-functional: that is, they both attended to COMMASUM from RNAME and FEEL, and also attended to RNAME and FEEL from END, producing output in the direction of the logit difference. This is possible because these heads exist at different layers, and later heads can read the summarized information from previous heads as well as writing their own summary information.

Direct effect heads Specifically, the direct effect heads were:

- Head 17.19 did not attend to commas significantly, but did attend to the periods at the end of each preference sentence in addition to its primary attention to RNAME and FEEL, and did not display COMMASUM-reading behavior.
- Head 22.5 attended almost exclusively to FEEL, and did not display COMMASUM-reading behavior.
- Other direct effect heads (14.4, 20.10 and 12.2) did show COMMASUM-reading behavior as well as reading from the near-end tokens to produce output in the direction of the logit difference. In each case, we verified with path-patching that information from these positions was causally relevant.

Name summary writers We also found important heads (12.17 being by far the most important) that are only engaged with attending to COMMASUM and producing output at RNAME and FEEL.

Comma summary writers We further investigated what circuitry was causally important to task performance mediated through the COMMASUM positions, but did not flesh this out in full detail; after finding initial examples of summarization, we focused on its causal relevance and interaction with the sentiment direction, leaving deeper investigation to future work.

A.4 Additional summarization findings

Circuitry for processing commas vs. original phrases is semi-separate Though there is overlap between the attention heads involved in the circuitry for processing sentiment from key phrases and that from summarization points, there are also some clear differences, suggesting that the ability to read summaries could be a specific capability developed by the model (rather than the model simply attending to high-sentiment tokens).

As can be seen in Figure A.8, there are distinct groups of attention heads that result in damage to the logit difference in different situations–that is, some react when phrases are patched, some react disproportionately to comma patching, and one head seems to have a strong response for either patching case. This is suggestive of semi-separate summary-reading circuitry, and we hope future work will result in further insights in this direction.

A.5 Neurons writing to sentiment direction in GPT2-small are interpretable

We observed that the cosine similarities of neuron out-directions with the sentiment direction are extremely heavy tailed (Figure A.10). Thanks to Neuroscope (Nanda, 2023a), we can quickly see whether these neurons are interpretable. Indeed, here are a few examples from the tails of that distribution:

- L3N1605 activates on "hesitate" following a negation
- Neuron L6N828 seems to be activating on words like "however" or "on the other hand" *if* they follow something negative
- Neuron L5N671 activates on negative words that follow a "not" contraction (e.g. didn't, doesn't)
- L6N1237 activates strongly on "but" following "not bad"

We take L3N1605, the "not hesitate" neuron, as an extended example and trace backwards through the network using Direct Logit Attribution[7]. We computed the relative effect of different model components on L3N1605 in the two different cases "I would not hesitate" vs. "I would always hesitate". The main contributors to this difference are L1H0, L3H10, L3H11 and MLP2. Expanding out MLP2 into individual neurons we find that the contributions to L3N1605 are sparse. For example, L2N1154 activates on words like "don't", "not", "no", etc. It activates on "not" but not "hesitate" in "I would not hesitate" but activates on "hesitate" in "I would always hesitate". Visualizing the attention pattern of L1H0 shows that it attends from "hesitate" to the previous token if it is "not", but not if it is "always".

These anecdotal examples suggest at a complex network of machinery for transmitting sentiment information across components of the network using a single critical axis of the residual stream as a communication channel. We think that exploring these neurons further could be a very interesting avenue of future research, particularly for understanding how the model updates sentiment based on negations where these neurons seem to play a critical role.

A.6 Detailed description of metrics

- **Logit Difference:** We extend the logit difference metric used by Wang et al. (2022) to the setting with 2 *classes* of next token rather than only 2 valid next tokens. This is useful in situations where there are many possible choices of positively or negatively valenced next tokens. Specifically, we examine the average difference in logits between sets of positive/negative next-tokens $T^{\text{positive}} = \{t_i^{\text{positive}} : 1 \leq i \leq n\}$ and $T^{\text{negative}} = \{t_i^{\text{negative}} : 1 \leq i \leq n\}$ in order to get a smooth measure of the model's ability to differentiate between sentiment. That is, we define the logit difference as $\frac{1}{n}\sum_i \left[\text{logit}(t_i^{\text{positive}}) - \text{logit}(t_i^{\text{negative}})\right]$. Larger differences indicate more robust separation of the positive/negative tokens, and zero or inverted differences indicate zero or inverted sentiment processing respectively. When used as a patching metric, this demonstrates the causal efficacy of various interventions like activation patching or ablation.[8]
- **Logit Flip:** Similar to logit difference, this is the percentage of cases where the logit difference between T^{positive} and T^{negative} is inverted after a causal intervention. This is a more discrete measure which is helpful for gauging whether the magnitude of the logit differences is sufficient to actually flip model predictions.
- **Accuracy:** Out of a set of prompts, the percentage for which the logits for tokens T^{correct} are greater than $T^{\text{incorrect}}$. In practice, usually each of these sets only has one member (e.g., "Positive" and "Negative").

[7] This technique decomposes model outputs into the sum of contributions of each component, using the insight from Elhage et al. (2021b) that components are independent and additive

[8] We use this metric often because it is more sensitive than accuracy to small shifts in model behavior, which is particularly useful for circuit identification where the effect size is small but real. That is, in many cases a token of interest might become much more likely but not cross the threshold to change accuracy metrics, and in this case logit difference will detect it. Logit difference is also useful when trying to measure the model behavior transition between two different, opposing prompts–in this case, the logit difference for each of the prompts is used for lower and upper baselines, and we can measure the degree to which the logit difference behavior moves from one pole to the other.

A.7 Toy dataset details

The ToyMovieReview dataset consists of prompts of the form ”I thought this movie was ADJ, I VRB it. [NEWLINE] Conclusion: This movie is”. We substituted different adjective and verb tokens into the two variable placeholders to create a prompt for each distinct adjective. We averaged the logit difference across 5 positive and 5 negative completions to determine whether the continuation was positive or negative.

```
positive_adjectives_train:
  - perfect
  - fantastic
  - delightful
  - cheerful
  - good
  - remarkable
  - satisfactory
  - wonderful
  - nice
  - fabulous
  - outstanding
  - satisfying
  - awesome
  - exceptional
  - adequate
  - incredible
  - extraordinary
  - amazing
  - decent
  - lovely
  - brilliant
  - charming
  - terrific
  - superb
  - spectacular
  - great
  - splendid
  - beautiful
  - positive
  - excellent
  - pleasant

negative_adjectives_train:
  - dreadful
  - bad
  - dull
  - depressing
  - miserable
  - tragic
  - nasty
  - inferior
  - horrific
  - terrible
  - ugly
  - disgusting
  - disastrous
  - annoying
  - boring
  - offensive
  - frustrating
```