Tokenized SAEs: Disentangling SAE Reconstructions

Thomas Dooms [*1] **Daniel Wilhelm** [*1]

Abstract

Sparse auto-encoders (SAEs) have become a prevalent tool for interpreting language models' inner workings. However, it is unknown how strongly SAE features correspond to computationally important directions in the model. We empirically show that many RES-JB SAE features predominantly correspond to simple input statistics. We hypothesize this is caused by a large class imbalance in training data combined with a lack of complex error signals. We propose a method to reduce this behavior by disentangling token reconstruction from feature reconstruction. We achieve this by introducing a per-token bias, which provides an improved baseline for interesting reconstruction. This change yields significantly more interesting features and improved reconstruction in sparse regimes.

1. Introduction

The holy grail of mechanistic interpretability research is the ability to decompose a network into a semantically meaningful set of variables and algorithms. SAEs have emerged as a promising method to extract interpretable context (Cunningham, 2023). However, the importance of SAE features to model computation is still unknown. This paper specifically studies the impact of a training token frequency imbalance [1] on the variety of learned features.

We find that many features in medium-sized SAEs such as RES-JB (Lin & Bloom, 2024) are affected by this imbalance. This causes them largely to reconstruct a direction biased toward the direction of the most prevalent training data unigrams and bigrams. Empirically, we estimate that between 35% and 45% of the features reconstruct common unigrams and almost 70% reconstruct common bigrams. We hypothesize these features then moreso reflect training token statistics than interesting internal model behavior. We attribute this phenomenon to the following two observations:

- Local context is a strong approximation for latent representations, even in deeper layers.
- There is a prominent class imbalance in the training data of SAEs. Certain local combinations will appear much more frequently than specific global interactions.

Given both their frequency and strength in the representation, these local contexts occupy the majority of the features an SAE uses to minimize its reconstruction error. We show this to hold for all kinds of common n-grams. Furthermore, we hypothesize this to be the cause for a range of pathological behaviors exhibited by SAEs, such as the inability to generalize out-of-distribution in certain contexts.

Fortunately, these insights can be leveraged toward a solution; we propose a means to disentangle these ”uninteresting” feature reconstruction tokens from the interesting features. This is accomplished by extending the SAE with a per-token bias, allowing the SAE to represent a ”base” reconstruction for each token. This leaves room for more semantically useful features. Furthermore, the proposed bias lookup table is efficient, resulting in SAEs becoming less compute-intensive to train. Specifically, our contributions are [2]:

- We quantify the number of features demonstrating uninteresting behavior due to the input distribution and formulate why this is the case.
- We propose a technique to mitigate this behavior by separating token reconstruction from feature reconstruction. We name this approach *Tokenized SAEs*.

2. Background

Notation. Let $\mathbb{T}$ be a set of tokens. We assume for each $\mathbf{x} \in \mathbb{T}^N$ that $t_0 = \text{BOS} \in \mathbb{T}$, the beginning-of-sequence token. Then we define an n-gram as $[\text{BOS}, t_1, t_2, \ldots, t_n] \in \mathbb{T}^{n+1}$.

[*]Equal contribution [1]Independent. Correspondence to: Thomas Dooms <doomsthomas@gmail.com>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1] The terminology ”imbalance” accurately describes its effect, although it may best be described as a weighted class.

[2] This work is preliminary and still lacks some prominent experiments.

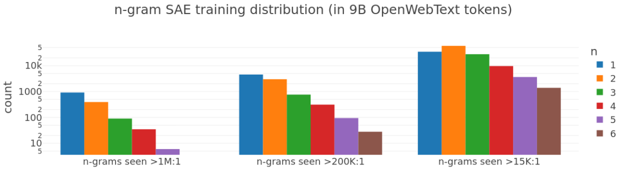


Figure 1. Particular n-grams are seen exponentially more often than others. Many combinations occur millions of times more than an arbitrary n-gram.

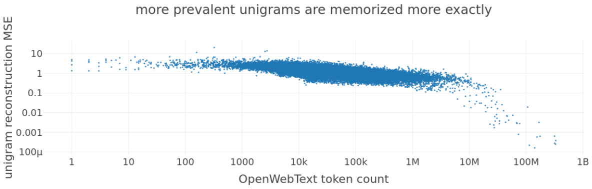


Figure 2. The reconstruction MSE of the layer 8 RES-JB SAE decreases with increasing training example frequency. This indicates the SAE effectively memorizes the most common tokens (via unigram training examples).

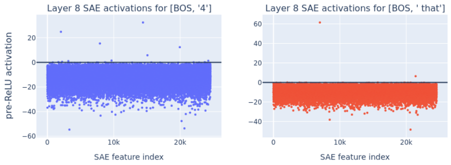


Figure 3. To memorize unigrams exactly and sparsely, the SAE represents each using a small subset of feature neurons that fire in response to the unigram. Due to the incorporation of prior token information, later layers often also strongly memorize bigrams.

We can relate the input token sequence $\mathbf{x}$ directly to observed values at some location p in the model. For example, if at p we observe the residual stream values $\mathbf{X}^p$, there exists a map $A^p : \mathbb{T}^N \mapsto \mathbb{R}^{N \times d}$ (or generally to such dimensions observable at p).

Imbalance. We will examine sparse auto-encoders at location p, which map $\text{SAE}^p : \mathbb{R}^d \mapsto \mathbb{R}^d$. The sparsity of this map is minimized, leading to seemingly interpretable features. However, short n-grams are exponentially over-represented due to an imbalanced training distribution. This biases the SAE toward these short n-gram inputs, giving rise to token reconstruction features.

As we forward-pass the $N \times d$ residual stream $\mathbf{X}$, due to attention we can consider each row vector $\mathbf{X}_i$ a function only of the first $i+1$ row vectors. For any location p and $\mathbf{x} \in \mathbb{T}^N$, this gives the identity $A^p(\mathbf{x})_i = A^p(\mathbf{x}_{\leq i})_i$, where $\mathbf{x}_{\leq i}$ is the $(i+1)$-token prefix to $\mathbf{x}$. For example: $A^p([\text{BOS}, t_0, t_1, \ldots, t_N])_1 = A^p([\text{BOS}, t_0])_1$.

It follows that a given row vector index $0 \leq i < N$ of $A^p(\mathbf{x})$ is completely described by an i-gram, of which there are at most $|\mathbb{T}|^i$. Yet, SAEs typically are trained on each row uniformly. Hence, early-row activations will be exponentially over-represented in the training distribution.

The degree of over-representation can be measured directly for a given training set. Assuming each training sequence begins at a random token, the n-gram frequency distribution follows the dataset's n-token frequency distribution. We show many n-grams are more than a million times more likely than the baseline (Figure 1).

Due to this, SAE training exhibits training example stratification based on sequence position. Each row vector follows a different distribution, which causes the SAE to become biased toward the initial (most highly-weighted) distribution. Such a class weighting causes a general MSE-trained regressor to underestimate rare labels (Ren, 2022).

This is similar to ”imbalanced regression”, where the target space distribution is sampled unevenly during training (Yang et al., 2021; Stocksieker et al., 2024). The SAE input stratification results in similar effects. The result is a bias toward the highest-weighted regions of space, here those of the most common small n-gram inputs. This will cause higher reconstruction loss for less common n-grams (i.e. the majority of prompts), since they must ”overcome” the biases.

3. Sparse Auto-Encoders

The motivation for training SAEs is often presented as feature discovery. This is achieved by reconstructing the hidden representations through a sparse hidden basis, often called features. We show that SAEs memorize and organize themselves around the most common input n-grams, contributing to the observed correlation between them (Figure 4).

Memorization. Suppose the most common n-gram inputs cause a training imbalance. Then we would expect to see (and observe) that with larger n-gram frequency, the reconstruction MSE decreases (Figure 2) and fewer features activate (Figure 3). Generally, we observe these correlations weaken with the SAE layer. In later layers, attention has likely consolidated information from other tokens, making the most common representations involve prior tokens. For example, many common words require multiple tokens to represent. We have observed evidence for this by noting that unigrams are most commonly activated in early layers and bigrams in later layers.

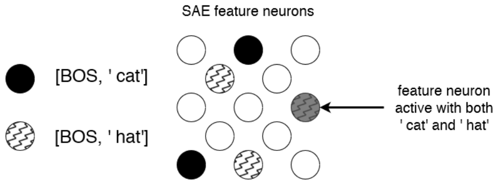


Figure 4. Illustrating experimental results, an individual feature neuron is activated when one of its associated n-grams is present. The most common tokens will occupy a full feature while less common tokens will share a feature. To maximize reconstruction, this sharing occurs between semantically similar tokens.

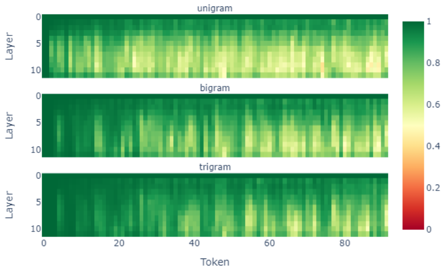


Figure 5. Measuring cosine similarity of hidden representations and a patched version which only includes the last n tokens in GPT-2 small. Trigrams are generally an adequate approximation across the network.

Token Reconstruction Features. Suppose some SAE is represented by a set of features $\mathbb{F}$. Based on the prior experimental results and imbalance theory, we hypothesize that each common n-gram $\mathbf{x}$ maps to a subset of $\mathbb{F}$ which activates $A^p(\mathbf{x})$. The set of increasingly most common n-grams approaches a cover of $\mathbb{F}$. (Figure 4)

Then, an SAE feature activates when one of the corresponding n-grams appears in $\mathbf{x}$. We can show this experimentally by predicting which input tokens will activate a given feature. In RES-JB layer 8, of the 76% of features activated by a unigram, 39% matched the top unigram activation, and 66% matched at least one. The 24% of features not activated by a unigram illustrate:

1. In later layers, some common SAE inputs may result from non-local information that more likely occurs in longer sequences. Experimental evidence shows that a minority of layer 8 GPT-2 features do not respond to any of 212K most-common ($n \leq 6$)-grams. A qualitative characterization of these features reveals these features exhibit more interesting semantic behavior.

2. This method operates under the assumption that some n tokens prior to row vector i are sufficient to mostly describe the SAE inputs, i.e. $A^p(\mathbf{x})_i \approx A^p(\mathbf{x}_{i-n})_i$. We show this to generally be the case in Figure 5.

4. Tokenized SAEs

To resolve the abovementioned issues, we propose a new method that separates token reconstruction features from the dictionary. This is achieved by adding a separate path to the SAE, only concerned with providing a base reconstruction of tokens. Concretely, we add a per-token bias, corresponding to a lookup table (Equation 2). Let index row vector $\mathbf{a}_i$ correspond to input token $\mathbf{t} = \mathbf{x}_i$. We initialize $W_{lookup}(\mathbf{t})$ with $A^p([\text{BOS}, \mathbf{t}])_1$ and add it as follows:

$$\begin{aligned} \mathbf{f}(\mathbf{a_t}) &= \text{ReLU}(W_{\text{enc}}(\mathbf{a_t} - b_{dec}) + b_{enc}) \quad (1) \\ \hat{\mathbf{a}}_t &= W_{\text{dec}}\mathbf{f}(\mathbf{a_t}) + b_{dec} + W_{lookup}(\mathbf{t}) \quad (2) \end{aligned}$$

This lookup table has no impact on the encoding thus computing feature activations requires no change in setup. However, token information is necessary for the reconstruction. We provide further details in Appendix A.

Results. The experiments in this section are all performed on layer 8 of GPT-2 small. This is sufficiently deep in the model that we would expect complex behavior to have arisen. Furthermore, a breadth of differently-sized pretrained SAEs exist that can be used for comparison. We use the *added* cross-entropy (Equation 3) to measure the impact on the model prediction and *normalized* MSE (Equation 4) to measure reconstruction [3].

$$CE_{added}(\mathbf{x}) = \frac{(CE_{patched}(\mathbf{x}) - CE_{clean}(\mathbf{x}))}{CE_{clean}(\mathbf{x})} \quad (3)$$

$$\text{NMSE}(\mathbf{x}) = \frac{||\mathbf{x} - \text{SAE}(\mathbf{x})||_2}{||\mathbf{x}||_2} \quad (4)$$

The strength of Tokenized SAEs is their ability to maintain good reconstruction in hyper-sparse circumstances, significantly outperforming gated SAEs (Rajamanoharan et al., 2024), shown in Figure 6.

5. Feature Comparison

Quantitave. We quantify the number of uninteresting features by sampling each possible unigram (pre-pended with BOS) and measuring the number of features that activate

[3] The proposed SAEs were trained on consumer hardware and could be undertrained. Furthermore, we find that Tokenized SAEs suffer from a very high number of dead features. We suspect solving this will yield further improvements. Lastly, experimental results reveal that adding the per-token bias to gated SAEs doesn't improve their reconstruction or sparsity. We currently offer no hypothesis for this.

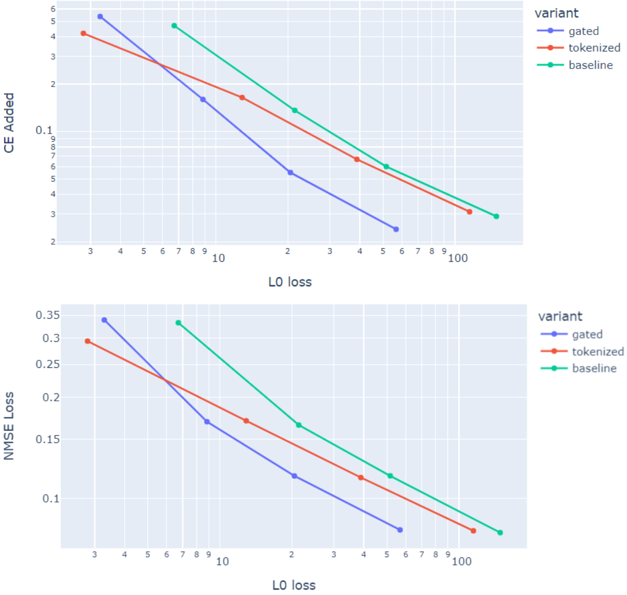


Figure 6. The cross-entropy added and normalized MSE compared to the L0 norm. This shows Tokenized SAEs outperform its baseline version by a significant margin. Specifically, it achieves the same reconstruction while being about 25% sparser. In hyper-sparse regimes, it also outperforms Gated SAEs.

strongly for it. Features that strongly correspond to very few tokens are highly likely to be feature reconstruction tokens. We display which distribution they follow in Figure 7.

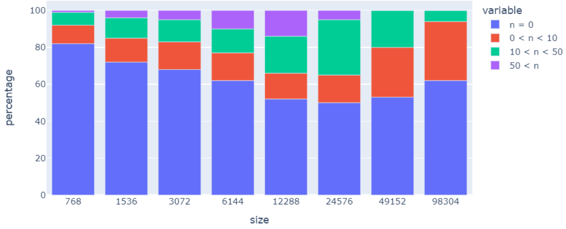


Figure 7. Approximate categorization of features by the number of tokens they activate on (above a threshold of 5). In smaller SAEs, there is no bandwidth to represent individual or small sets of tokens. In medium-sized SAEs, we see features representing small sets of tokens. As size increases, it starts representing specific tokens strongly.

We perform the same experiment on the Tokenized SAEs from Figure 6. We find that the number of features that activate on any single unigram is below 5% for all of them.

Qualitative. We performed a brief blind study on three layer 8 GPT-2 SAEs: Tokenized ($L_0 = 12$), Baseline ($L_0 = 22$), and RES-JB (Lin & Bloom, 2024). The results are shown in Table 1.

	RES-JB	Tokenized	Baseline
Interpretability	7.7	7.2	6.8
Complexity	3.5	6.9	1.5

Table 1. We observe 10 features from multiple SAEs and note their mean complexity and interpretability. For complexity, we follow the following scoring convention: 1-3 unigrams, 4-6 simple patterns, and 6-10 complex semantic behavior. The scoring of interpretability occurred as follows: 1-4 no discernable pattern, 5-7 noisy pattern, 8-10 clear pattern.

This implies that while our SAE features are slightly less interpretable due to the lack of trivial unigram features, their complexity is significantly higher. Appendix B includes a list of cherry-picked features to corroborate these subjective findings. In summary, we find that features generated by Tokenized SAEs tend to be more semantically meaningful and contain fewer uninteresting features.

6. Future Work

Tokenized SAEs have a wide possible range of extensions. This section outlines three promising ideas from most likely to work towards most speculative.

1. Incorporate n-gram Statistics. This paper only considers including unigrams as a reconstruction baseline. This can be extended towards any common n-gram in the training data. We believe this to be mostly an engineering challenge; it requires efficiently making a sparse, multi-token lookup table for combinatorically more n-grams.

2. Further Architectural Changes. As stated in section 4, naively adopting this setup to the novel Gated SAE architectures leads to slightly worse results. It would be interesting to see how this can be fixed.

3. Including Previous SAE Features. Token bases are known to be sparse and are therefore a natural fit for tokenized SAEs. However, with some modifications, we could also use a previous SAE as a sparse basis. This would change the role of SAEs from reconstructing towards ”diff-ing” the residual stream in similar bases.

Additionally, a more thorough study into the quality of Tokenized SAE features is still to be performed. This should be done on both the dictionary and the lookup table. The former is related to the incorporated non-local context and the latter is related to the token reconstruction. Exactly characterizing this token reconstruction similarity in latent representations is undoubtedly useful. Lastly, as model sizes grow, the proposed technique will certainly deteriorate. However, we are excited to see if this technique can still yield more interesting features in such cases.

References

Cunningham, Hoagy, e. a. Sparse autoencoders find highly interpretable features in language models., 2023.

Lin, J. and Bloom, J. Announcing Neuronpedia: Platform for accelerating research into Sparse Autoencoders. March 2024.

Rajamanoharan, S., Conmy, A., Smith, L., Lieberum, T., Varma, V., Kramár, J., Shah, R., and Nanda, N. Improving dictionary learning with gated sparse autoencoders, 2024.

Ren, Jiawei, e. a. Balanced mse for imbalanced visual regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Stocksieker, S., Pommeret, D., and Charpentier, A. Boarding for iss: Imbalanced self-supervised: Discovery of a scaled autoencoder for mixed tabular datasets, 2024.

Templeton, A., Conerly, T., Marcus, J., Lindsey, J., Bricken, T., Chen, B., Pearce, A., Citro, C., Ameisen, E., Jones, A., Cunningham, H., Turner, N. L., McDougall, C., MacDiarmid, M., Freeman, C. D., Sumers, T. R., Rees, E., Batson, J., Jermyn, A., Carter, S., Olah, C., and Henighan, T. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

Yang, Y., Zha, K., Chen, Y.-C., Wang, H., and Katabi, D. Delving into deep imbalanced regression. In *Proceedings of the 38th International Conference on Machine Learning (ICML 2021)*, 2021.

A. SAE Setup

A.1. General

We perform our experiments using the following hyperparameters (Table 2). The setup is intentionally kept as simple as possible to avoid confounding factors. Specifically, no resampling or ghost gradients are used. The only notable difference between the baseline SAEs and our tokenized SAEs is the initialization of the encoder, which is not set to the transpose of W_{dec} but rather Kaiming normal initialized. The reason is that the identity-like structure is no longer required. We have not considered alternative initialization strategies. Beyond a baseline implementation, we also compare to a highly-tuned Gated-SAE combined with the improved loss sparsity term from Templeton et al. (2024). We have found this to be the current state-of-the-art.

A.2. Lookup Table Initialization

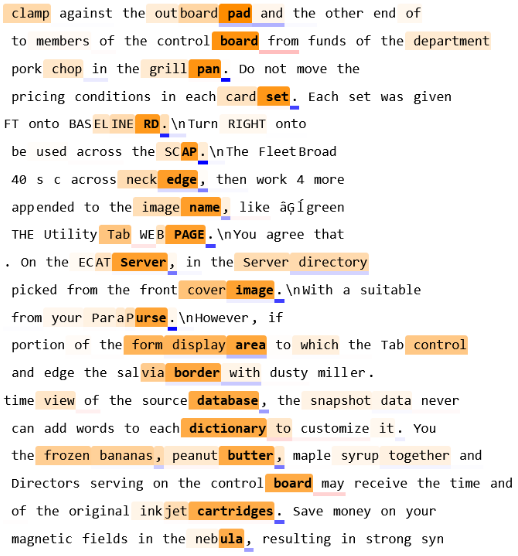
Maintaining the same topic, the lookup table is initialized with an approximate ”clean” token reconstruction. Specifically, we use the latent representation of unigrams for the reconstruction. These unigrams are considered clean because they lack context. To reduce model breakage, the BOS token is pre-pended. Attention heads typically ”sink” attention either to a BOS token or to themselves, and this setup preserves that behavior.

A.3. Hyperparameters

Data	
dataset	C4
tokens	600M
buffer size	128K
batch size	4096
SAE	
location	resid_pre layer 8
expansion	16
features	12288
context	256
sparsity	[8.0; 64.0]
learning rate	10^{-4}
scheduler	cosine annealing
optimizer	AdamW
sparsity loss	including W_{dec}
sparsity warmup	first 5%

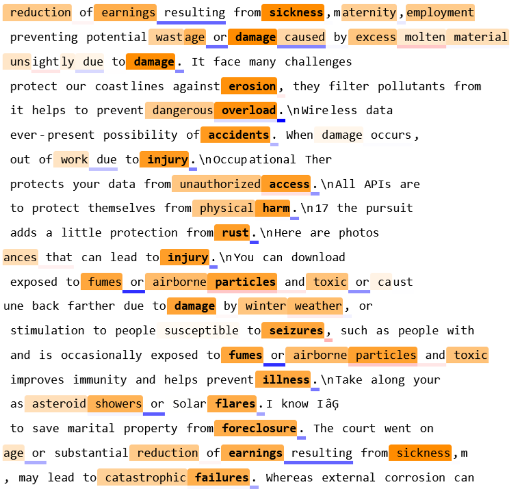
Table 2. Dataset and model parameters

B. Cherry-Picked Features



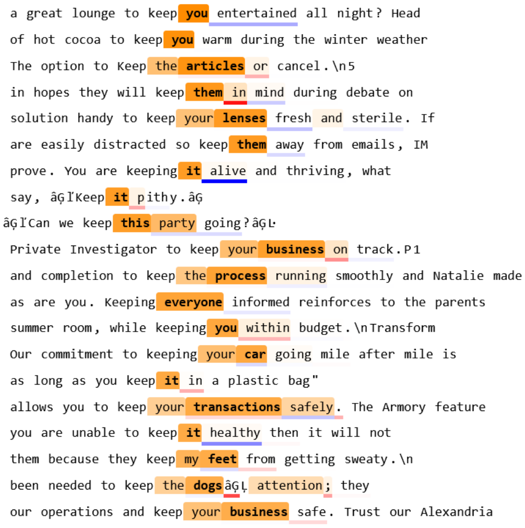
clamp against the outboard **pad** and the other end of
to members of the control **board** from funds of the department
pork chop in the grill **pan**. Do not move the
pricing conditions in each card **set**. Each set was given
FT onto BASELINE **RD**.\nTurn RIGHT onto
be used across the SC**AP**.\nThe FleetBroad
40 s c across neck **edge**, then work 4 more
appended to the image **name**, like âĢÍgreen
THE Utility Tab WEB **PAGE**.\nYou agree that
. On the ECAT **Server**, in the Server directory
picked from the front cover **image**.\nWith a suitable
from your ParaP**urse**.\nHowever, if
portion of the form display **area** to which the Tab control
and edge the salvia **border** with dusty miller.
time view of the source **database**, the snapshot data never
can add words to each **dictionary** to customize it. You
the frozen bananas, peanut **butter**, maple syrup together and
Directors serving on the control **board** may receive the time and
of the original inkjet **cartridges**. Save money on your
magnetic fields in the neb**ula**, resulting in strong syn

Figure 8. An end of sentence feature, boosting ”.”, ”,”, and ”and” tokens.



reduction of earnings resulting from **sickness**,maternity,employment
preventing potential wastage or **damage** caused by excess molten material
unsightly due to **damage**. It face many challenges
protect our coastlines against **erosion**, they filter pollutants from
it helps to prevent dangerous **overload**.\nWireless data
ever-present possibility of **accidents**. When damage occurs,
out of work due to **injury**.\nOccupational Ther
protects your data from unauthorized **access**.\nAll APIs are
to protect themselves from physical **harm**.\n17 the pursuit
adds a little protection from **rust**.\nHere are photos
ances that can lead to **injury**.\nYou can download
exposed to fumes or airborne **particles** and toxic or caust
une back farther due to **damage** by winter weather, or
stimulation to people susceptible to **seizures**, such as people with
and is occasionally exposed to **fumes** or airborne particles and toxic
improves immunity and helps prevent **illness**.\nTake along your
as asteroid showers or Solar **flares**.I know IâĢ
to save marital property from **foreclosure**. The court went on
age or substantial reduction of **earnings** resulting from sickness,m
, may lead to catastrophic **failures**. Whereas external corrosion can

Figure 9. A health hazard feature.



```
a great lounge to keep you entertained all night? Head
of hot cocoa to keep you warm during the winter weather
The option to Keep the articles or cancel.\n5
in hopes they will keep them in mind during debate on
solution handy to keep your lenses fresh and sterile. If
are easily distracted so keep them away from emails, IM
prove. You are keeping it alive and thriving, what
say, âĢľKeep it pithy.âĢ
âĢľCan we keep this party going?âĢĿ
Private Investigator to keep your business on track.P1
and completion to keep the process running smoothly and Natalie made
as are you. Keeping everyone informed reinforces to the parents
summer room, while keeping you within budget.\nTransform
Our commitment to keeping your car going mile after mile is
as long as you keep it in a plastic bag"
allows you to keep your transactions safely. The Armory feature
you are unable to keep it healthy then it will not
them because they keep my feet from getting sweaty.\n
been needed to keep the dogsâĢļ attention; they
our operations and keep your business safe. Trust our Alexandria
```

Figure 10. A direct object feature.

C. Blind Study Notes

C.1. Joseph

```
"faculty" + "alumni" | 7 | 2
quantitative measures  | 9 | 7
"Che"/"sche"/"ische"/"arche" | 7 | 4
" to" | 9 | 1
tokens inside trigram compounds | 9 | 6
"L" | 8 | 1
adjective + to | 7 | 3
"make it" | 8 | 3
numbers | 6 | 4
nouns, not specific | 7 | 4
```

C.2. Tokenized

```
tokens after places where hyphens should've occurred | 7 | 9
politics, specifically voting | 7 | 8
single character token + hyphen | 8 | 5
split tokens in company names | 5 | 7
comma + nearby context | 8 | 4
wholesome context | 8 | 9
"e" in compound word | 7 | 4
"miss"/"missing" skip-gram | 6 | 5
time pressure | 7 | 9
end of sentence feature | 9 | 9
```

C.3. baseline

```
"International" | 8 | 1
"A+" | 8 | 1
"light" | 7 | 1
"interesting", "intriguing" | 7 | 2
"Prov" | 7 | 2
```

```
"J" | 7 | 1
"ahead", "free", "proceed" | 5 | 2
"=" | 8 | 1
"Alban" + random tokens | 4 | 2
"with the" | 7 | 2
```

D. Neuronpedia feature Study

Index	Term	Type
0	numbers	Unigram collection
1	“Pier”	**Unigram**
2	“weeks”/”months”/”years”	Unigram collection
3	Token after sorry/apologize	Bigram collection
4	separator/time	Attention
5	“in”	**Unigram**
6	Adjectives related to famousness	Unigram collection + attention
7	recipe(s)	**Unigram**
8	Causality (by a/due to)	Bigrams + attention
9	“Ļ”	**Unigram**
10	“Ļ” (again, look it up)	**Unigram**
11	Not sure	Attention
12	“told”	**Unigram**
13	solved, addressed, resolved	Unigram collection
14	“example”	**Unigram**
15	Really not sure…	*nan*
16	“With”	**Unigram**
17	“Ste”	**Unigram**
18	numerics in brackets (references)	Bigram collection
19	“s” after number (20s)	Bigram collection
20	Anglo + Alred + Pf	Unigram collection

Table 3. A qualitative study into the first 21 features of Joseph Blooms GPT-2 resid_pre SAE on layer 8. We show that more than half of the features represent uninteresting reconstructions.