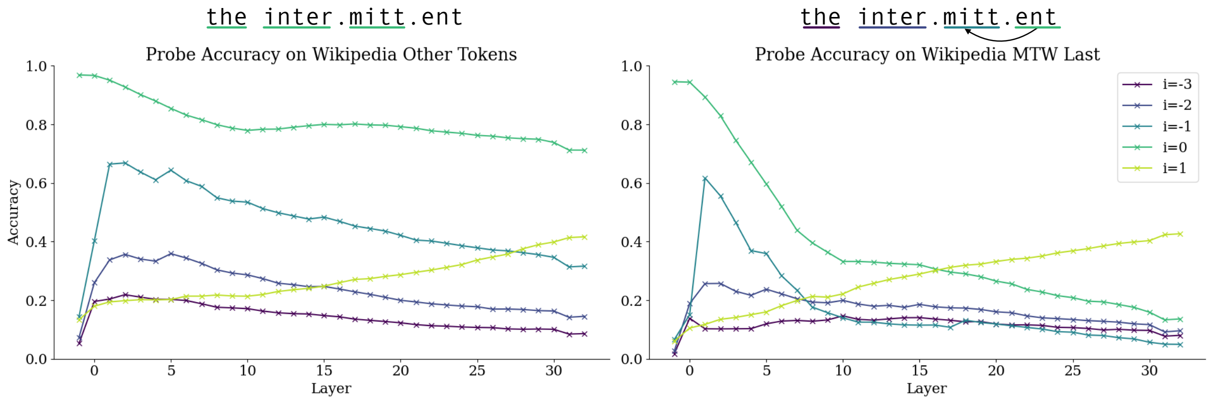

Figure 3: Top-1 test accuracy of probes on last tokens of Wikipedia multi-token words for Llama-2-7b ($n = 80606$). Accuracy on all other tokens shown on the left. We see an erasing effect for multi-token words, similar to the effect seen for COUNTERFACT subjects in Figure 2.

4.1 An Erasure Score

Given some arbitrary sequence of tokens from indices p through q, we want to design an *erasure score* that captures intuitions from Section 3. This score should be higher for sequences exhibiting token erasure (which we hypothesize to be lexical items like [_Cal, g, ary]), and lower for other types of token sequences (e.g., [_go _to, _Cal, g]). We design a metric $\psi_{p,q}$ that uses probe outputs from Section 3 to look for erasure effects betwen layer 1 and layer L.[3]

Concretely, Equation 1 defines the score $\psi_{p,q}$ for a sequence $s_{p,q}$ of length $n = q - p + 1$ as:

$$\frac{1}{1+2n}\left(\delta(q,0) + \sum_{t=p}^{q}\sum_{i=-2}^{-1} \mathbb{1}_{\text{within}}(t,i)\cdot\delta(t,i)\right) \tag{1}$$

where $\delta(t,i)$ denotes the change in probability of the predicted token $t+i$ from layer 1 to layer L, based on probes $p_i^{(\ell)}$ from Section 3.1. We take the softmax of the probe outputs to obtain the probability of a specific token $t+i$ in Equation 2.

$$\delta(t,i) = P_{p_i^{(1)}}(t+i|h_t^{(1)}) - P_{p_i^{(L)}}(t+i|h_t^{(L)}) \tag{2}$$

Finally, if $t+i$ lies outside the boundaries of s, we want the score to decrease. If it is within the boundaries of s, we want a large drop between layers $\delta(t,i)$ to increase the value of $\psi_{p,q}$.

$$\mathbb{1}_{\text{within}}(t,i) = \begin{cases} -1 \text{ if } t+i < p \\ 1 \text{ else} \end{cases} \tag{3}$$

In summary, for every token position $p \le t \le q$ and prediction offset $i \in \{-2,-1\}$, we measure the drop in the predicted probability of the correct token $t+i$ between layer 1 and layer L. The more that the probability of the correct answer *decreases* in early layers, the higher we score that sequence. However, if this "forgetting" occurs for tokens outside of the boundaries of s, we subtract that value from the overall score, effectively penalizing the sequence. This intuition comes from close inspection of probe behavior—for example, Figure 13 shows that there is no "forgetting" effect for $i = -1$ when probing the first token of COUNTERFACT subjects. With this approach, we can also account for cases where s is a subsequence of a larger lexical item: if the token g shows a forgetting effect for _Cal in [_Cal, g, ary], then the sequence [g, ary] would be penalized. Finally, $\delta(q,0)$ additionally rewards sequences in which the last token "forgets itself," as seen in Figures 2 and 3. We then normalize by the total number of δ values considered, in order to account for differing sequence lengths.

4.2 Segmenting Documents

We develop an algorithm built around our erasure score ψ that breaks any given document $d \in \mathcal{D}$ into high-scoring, non-overlapping segments covering all of d (Algorithm 1). Figure 1 shows the top-scoring sequences $s_{p,q}$ calculated in this manner from a Wikipedia excerpt about Thelonious Monk, where unigram scores are excluded for clarity. Not all multi-token words are scored highly via our approach, but the highest-scoring sequences are plausible lexical items that are non-compositional in nature ("dram.atic", "sil.ences", "tw.ists"). We share examples of more documents with complete segmentations in Appendix D.

[3] For both Llama-2-7b and Llama-3-8b we set $L = 9$.

Algorithm 1 Document Segmentation

```
Require: document $d \in \mathcal{D}$ of length $l$
for $n = 1$ to $l$ do                                  ▷ all ngram lengths
    for $p = 0$ to $l - n$ do
        for $q = p + n - 1$ to $l - 1$ do
            assign score $\psi_{p,q}$ to sequence $s_{p,q}$
        end for
    end for
end for
sort $s$ in descending order of $\psi$
$segms \leftarrow \emptyset$
for $s_{p,q}$ in sorted $s$ do
    if $\forall s_{x,y} \in segms, (x > q \vee y < p)$ then
        $segms \leftarrow segms \cup \{s_{p,q}\}$
    end if
end for
return $segms$                                         ▷ non-overlapping segments
```

Token Sequence	n	ct	ψ
lower case	3	2	0.736012
storm	2	4	0.716379
excursion	4	2	0.713134
====... *(72 ‘equals’ signs)*	8	2	0.712982
Mom	3	2	0.706778
acre	3	2	0.629213
Subject	3	2	0.607172
ninth	3	2	0.606669
processing elements	3	2	0.599549
CVC	3	2	0.596735

Table 1: Top ten highest-scoring sequences for Llama-2-7b using a Pile subsample (1658 sequences recovered total). n is the number of tokens in the sequence, and ‘ct’ represents occurrences of this segment. ψ is averaged over all occurrences.

4.3 Model Vocabularies

Finally, we propose a method to “read out” the implicit vocabulary of a model $\mathcal{M}$ given a dataset $\mathcal{D}$. For each document $d \in \mathcal{D}$, we segment d using Algorithm 1. We then average scores ψ for every multi-token sequence that appears more than once in $\mathcal{D}$. As this process is very data-dependent, we show results for both Pile and Wikipedia text. The top 50 sequences for each dataset and model are provided in Appendix E.

With this approach, we are able to recover $\sim$1800 sequences for Llama-2-7b and $\sim$900 for Llama-3-8b using the same five hundred Wikipedia articles. Although recall is quite low (Table 2), we find that 44.9% of sequences recovered for Llama-2-7b on Wikipedia text are either multi-token words or multi-token entities (29.68% for Pile text). For Llama-3-8b, only 5% and 3% of retrieved sequences are multi-token words or entities. However, looking at examples of Llama-3-8b sequences in Appendix E, we can observe other interesting cases, like multi-token expressions (“gold medalists,” “by per capita income,” “thank you for your understanding”) and `LaTeX` commands (as similarly observed by Elhage et al. (2022)). Because Llama-3-8b’s *token* vocabulary is four times larger than Llama-2-7b’s, its *implicit* vocabulary also seems to consist of larger multi-word expressions and chunks of code rather than multi-token words (Appendix E, Table 7).

		MTW		MTE	
llama	data	prec.	recall	prec.	recall
2-7b	wiki	0.306	0.016	0.143	0.016
	pile	0.296	0.017	0.080	0.018
3-8b	wiki	0.044	0.001	0.010	0.000
	pile	0.023	0.001	0.012	0.001

Table 2: Precision and recall for aggregated results of Algorithm 1 run on Llama-2-7b and Llama-3-8b, using either Wikipedia or Pile documents ($|\mathcal{D}| = 500$). MTW refers to all multi-token words in the dataset when split by whitespace; MTE refers to all `spaCy` named entities.

5 Conclusion

In this work, we present preliminary evidence for the existence of an *implicit vocabulary* that allows models to convert from byte-pair encoded tokens to useful lexical items. We posit that the “erasure” effect we observe for Llama-2-7b and Llama-3-8b is a result of model processes that deal with multi-token expressions, and use this insight to propose a new method for “reading out” an LLM’s implicit vocabulary. This is a first step towards understanding the formation of lexical representations in LLMs, and may serve as a useful tool for elucidation of words that a given model “knows.”

Limitations

Evaluation of implicit vocabulary-building methods (Section 4) is challenging due to the lack of a known ground-truth. Our approach is motivated by the desire to understand the inner workings of the model being studied, but we have no authoritative reference that distinguishes between situations where a given sequence gets a high ψ value because it is truly treated as a lexical unit by the model, or where it may be due to an error in our methodology. To quantify our results, we have compared the extracted vocbulary to sequences that we assume to be likely lexical items: multi-token words and spaCy named entities. However, this likely does not cover all cases for which "token grouping" occurs in LLMs.

Another limitation of this work is that we have restricted our analysis to *known* entities. There is also the question of what happens for intermediate cases such as plausible-sounding fictional towns or names of people who are not famous. If ψ correlates with sequence presence in training data, these results could be useful for understanding how familiar an LLM is with a given word or entity.

Finally, our measurements have been run only on the Llama family of models and do not yet extend to non-Llama models of comparable size, or Llama models of larger sizes.

Ethics Statement

In this work, we restrict our analysis to English words, due to our biases as native speakers of English. We hope that this work can also provide valuable insights for other languages, especially low-resource languages, where understanding "what words an LLM knows" may be especially useful.

Acknowledgments

We thank Koyena Pal, David Smith, Bilal Chughtai, Chantal Shaib, Atticus Geiger, and Adrian Chang for helpful discussion and feedback throughout the course of this project. This work was supported in part by Open Philanthropy, and by the National Science Foundation (NSF) grant IIS-1901117.

Experiments were implemented using the nnsight library (Fiotto-Kaufman et al., 2024). Many were run on the Center for AI Safety Compute Cluster. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the sponsors.

References

Khuyagbaatar Batsuren, Ekaterina Vylomova, Verna Dankers, Tsetsuukhei Delgerbaatar, Omri Uzan, Yuval Pinter, and Gábor Bella. 2024. Evaluating subword tokenization: Alien subword composition and oov generalization challenge. *Preprint*, arXiv:2404.13292.

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.

Jaden Fiotto-Kaufman, Alexander R Loftus, Eric Todd, Jannik Brinkmann, Caden Juang, Koyena Pal, Can Rager, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Michael Ripa, Adam Belfki, Nikhil Prakash, Sumeet Multani, Carla Brodley, Arjun Guha, Jonathan Bell, Byron Wallace, and David Bau. 2024. Nnsight and ndif: Democratizing access to foundation model internals. *Preprint*, arXiv:2407.14561.

Wikimedia Foundation. 2022. Wikimedia downloads.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. *ArXiv*, abs/2304.14767.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Preprint*, arXiv:2305.01610.

Bernal Jiménez Gutiérrez, Huan Sun, and Yu Su. 2023. Biomedical language models are robust to sub-optimal tokenization. *Preprint*, arXiv:2306.17649.

Shahar Katz, Yonatan Belinkov, Mor Geva, and Lior Wolf. 2024. Backward lens: Projecting language model gradients into the vocabulary space. *Preprint*, arXiv:2402.12865.

Alex Mallen and Nora Belrose. 2023. Eliciting latent knowledge from quirky language models. *Preprint*, arXiv:2312.01037.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. In *Neural Information Processing Systems*.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.