Same Meaning, Different Tokens: Tokenization-Induced Shifts in Representations and Predictions

Anthony Ragazzi **Eugene Santos, Jr.**
Thayer School of Engineering, Dartmouth College, Hanover, NH, USA

Abstract

Tokenization, the process where an input string is segmented into a token sequence, is the initial step to many language modeling architectures. It has the power to define the space in which the entire model will operate. Despite its importance, its effects on the internal representations within large language models is not fully understood. This work builds towards understanding such effects. We first study the similarity of hidden layer representations in models with different tokenizers but otherwise identically trained. In addition we study the effects of semantic preserving perturbations on output distribution and surprisal. We show that although meaning is preserved, changes in tokenization can lead to shifts in hidden layer representation, output distribution, and surprisal. We identify tokenization as an early, discrete choice that can systematically shape how interventions on surface form translate into changes in internal representations and next-token predictions.

1 Introduction

Language modeling, particularly transformer Vaswani et al. (2017) based approaches, have produced excellent results on a variety of language modeling tasks, including machine translation Gumma et al. (2025), software engineering Yang et al. (2024), and medical reasoning Jeong et al. (2024). Due to their rapid adoption and near ubiquity in language modeling applications, these models have been the subject of extensive research. Substantial work has improved many aspects of this architecture, one of which is developing tokenizers such as Byte Pair Encoding (BPE) Sennrich et al. (2016), WordPiece Devlin et al. (2019), and SentencePiece Kudo and Richardson (2018) for different settings and constraints. The reason for this is clear, while large language models (LLMs) produce output strings, they operate on discrete token sequences, and changing the way the text segmented changes the representational basis in which the model operates. In order to obtain a complete understanding of these models, we need to understand the causal effect that the differences in tokenization can have on the models, particularly the effect of interventions on hidden representations and next token prediction.

It is insufficient to simply design tokenizers and study the outcomes on benchmark datasets. To understand the true effects that tokenization has we need a framework for understanding its effects that are internal to the model. Recent work has moved in this direction. Particularly, Altıntaş et al. (2025) has recognized the importance of isolating tokenizers for study, but evaluated their models using performance based metrics, not metrics concerning the internal representations to the model. Another approach, Rauba et al. (2024), introduced distribution based perturbation analysis, which analyzes the distribution of generated answers as a result of perturbations in input sequence. Other approaches move from output based metrics to internal distributions, as in Lesci et al. (2025), which defined tokenization bias as the effect of tokenization choice on model behavior, and studied whether a token's inclusion in a vocabulary affected the probability of its corresponding character string. While the others studied one type of tokenization bias, a more comprehensive understanding is essential.

Much of the recent work in this area is focused on output distributions rather than internal mechanisms. While these studies have merit, analyzing the outputs can only provide limited information on the model's inner workings. Recently, work has been done indicating that LLMs lack robustness to word level perturbations Alahmari et al. (2025); Wang et al. (2025). This is an area of concern, since in many real-world applications data can be noisy, including typos and other categories of word perturbation. In order to predict and improve robustness in real-world situations, the effects of changes in input sequence, whether through

string perturbations or tokenization differences, further investigation is required. This paper aims to do so. We will study both the internal representations of two models trained identically except for their tokenizers, as well as analyze distributional shifts and changes in surprisal due to perturbations in input. We build towards a goal of characterizing the effects input token sequence on model behavior. Our contributions are summarized here:

1. We isolate tokenizer choice by training matched models and measuring representational divergence via centered kernel alignment.
2. We apply an intervention based evaluation where semantics preserving edits change tokenization, and we quantify output distribution shifts and surprisal gaps.
3. We interpret tokenization as a mediator that maps surface interventions to internal changes, motivating a framework to study model stability

The paper is organized as follows: The methods used in our analysis are outlined in section 2. The details of our implementation are outlined in section 3. Our results are presented in section 4 along with a discussion of our findings. Finally, we conclude in section 5 and look towards future improvements.

2 Methods

We conducted two experiments in this paper. Both are ordered towards isolating the input token sequence, and analyzing the effects any differences might have on a model. Our first experiment trains two identical models with identical training data, except that they use different tokenizers. Here, this experiment is labeled "CKA Similarity", in reference to the analysis we do. The other experiment is designed for larger models, where training would be infeasible due to hardware and cost limitations. Instead, we compare within the same model, but perturb the input sequence so that semantics are preserved but tokenization is changed. We refer to this experiment as "Token Perturbations".

2.1 CKA Similarity

Here we pretrain two RoBERTa models of identical structure with identical training data. The only difference between the two is that we use two different tokenizers, BPE and WordPiece. Let $\mathcal{M}_B$ and $\mathcal{M}_W$ be the two models which use the BPE and WordPiece tokenizers, respectively. We wish to compare the sentence representations of $\mathcal{M}_B$ and $\mathcal{M}_W$ at each hidden layer. To do so, we extract the hidden states of a batch of n input sentences, and we apply mean pooling to obtain a representation of fixed length d for each sentence. We can then obtain a sentence representation matrix for model M at hidden layer i

$$S_{M_i} = [s_1, s_2, ..., s_n]^\top \tag{1}$$

where $S \in \mathbb{R}^{n \times d}$. Then we may apply our similarity metric, linear Centered Kernel Alignment (CKA) Kornblith et al. (2019). We chose this due to its invariance to orthogonal transformations and isotropic scaling, allowing for a focus on structure rather than exact feature alignment. Let $S_{B,i}$ be the sentence representation matrix from $\mathcal{M}_B$ at hidden layer i, and $S_{W,j}$ be the sentence representation matrix from $\mathcal{M}_W$ at hidden layer j. The linear CKA of $S_{B,i}$ and $S_{W,j}$ is given by

$$CKA(S_{B,i}, S_{W,j}) = \frac{\|S_{W,j}^\top S_{B,i}\|_F^2}{\|S_{B,i}^\top S_{B,i}\|_F^2 \|S_{W,j}^\top S_{W,j}\|_F^2} \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. We do this for all combinations of hidden layers $h_i \in \mathcal{M}_B$ and $h_j \in \mathcal{M}_W$.

2.2 Token Perturbations

In this part, we start with a standard English sentence x and a model $\mathcal{M}$. We obtain x' by applying perturbations that are approximately semantics preserving. These perturbations include compound words (e.g. health care becomes healthcare), expanding contractions (e.g. don't becomes do not) and punctuation removals (e.g. U.S.A. becomes USA). We ensure that these changes result in changes in tokenization, and compute two metrics to understand the effects that the perturbations have on output distribution. First, we measure surprisal using either negative log likelihood (NLL) for causal models Smith and Levy (2013) or negative pseudo log likelihood (NPLL) as presented in Salazar et al. (2020) for masked models. We obtain a token representation $\tau = \{t_1, t_2, ..., t_n\}$ of sentence x and compute NLL with the following

$$NLL(\tau) = -\sum_{i=1}^{|\tau|} log(P(t_i|\tau_{<i})) \tag{3}$$

Similarly, we compute NPLL according to

$$NPLL(\tau) = -\sum_{i=1}^{|\tau|} log(P(t_i|\tau_{\setminus i})) \tag{4}$$

We do this individually for sentences x and x', and then compute the perturbation induced surprisal gap

with the following

$$\begin{aligned} \Delta NLL &= NLL(\tau') - NLL(\tau) \\ \Delta NPLL &= NPLL(\tau') - NPLL(\tau) \end{aligned} \tag{5}$$

This surprisal gap can be used to quantify each model's sensitivity to the applied perturbations. In addition to surprisal, we also study the effects each perturbation had on the model's output probability distribution. To do so, we calculate top-k Jensen-Shannon Divergence (JS_k) between the two output distributions of the original and perturbed sentences, where top-k refers to the k most probable tokens from the two distributions. Choosing the top k tokens rather than the entire vocabulary allowed for increased computational efficiency while still capturing the majority of the probability mass Holtzman et al. (2019). This approach will become increasingly necessary as this project is scaled to larger models.

Let P_t be an output probability distribution over vocabulary $\mathcal{V}$ for model $\mathcal{M}$ and input token t evaluated on sentence x. Let Q_t be an output probability distribution over vocabulary $\mathcal{V}$ for model $\mathcal{M}$ and input token t evaluated on sentence x' The top-k truncated probability distributions are given by

$$\begin{aligned} p_t &= [p_{(1)}, p_{(2)}, ..., p_{(k)}, p_{rem}] \\ q_t &= [q_{(1)}, q_{(2)}, ..., q_{(k)}, q_{rem}] \end{aligned} \tag{6}$$

where each $p_{(i)}, q_{(j)}$ is the i^{th} largest probability values in their respective distributions, and $p_{rem}, qrem$ is the cumulative probability of the remaining $|\mathcal{V}|-k$ tokens, ensuring probability sums to one. Then, for each token t compute JS_k for p_t, q_t,

$$JS_k(p_t \| q_t) = \frac{1}{2} KL(p_t \| m) + \frac{1}{2} KL(q_t \| m), m = \frac{1}{2}(p_t + q_t) \tag{7}$$

Where KL represents Kullback-Leibler divergence. We compute the mean JS_k for all the tokens in the sentence, allowing us to understand distributional shifts between the two different input sentences.

3 Implementation Details

The implementation details of the two experiments are presented in the following.

3.1 CKA Similarity

To fairly compare the hidden layer representations of two models differing only in tokenizer, we needed to train the models from scratch. To do so, we trained two small RoBERTa models, each containing 4 hidden layers, each with dimension 256, and 4 attention heads per layer. The intermediate size was set to 1024 and we used a max sequence length of 128. The models were trained with a masked language model objective on the wikidata Vrandečić and Krötzsch (2014) and bookcorpus Zhu et al. (2015) datasets with vocabulary size 30,000. We used a global batch size of 512 and peak learning rate of 5×10^{-4}. The only difference between the two was the tokenizer, one using BPE and the other WordPiece.

3.2 Token Perturbations

We began with a set of 100k standard English sentences coming from the High Quality English Sentences Dataset Tseng (2024). For each sentence we applied up to 5 perturbations. Of the 100k sentences, 66,573 sentences received a single perturbation, while 25,016, 13,804, 4,818, and 2,650 sentences were perturbed two, three, four, and five times, respectively. Note that the same original sentence, if it can be perturbed more than once, was included in multiple categories. When computing JS_k, we set $k = 100$. The models used were the two RoBERTa models already presented, GPT2 Radford et al. (2019), LLaMa 3.2-3B Grattafiori et al. (2024), and Mistral-7B-v0 Jiang et al. (2023).

4 Results and Discussion

First, we analyze our two small RoBERTa models, trained identically with the exception of the tokenizer, one of which was trained with a WordPiece tokenizer and another a BPE tokenizer. As shown in Fig. 1, the cross layer linear CKA indicates that, while early layers have high similarity, later layers are more dissimilar. This suggests that the tokenizer may act as a bias that can steer the representation learning.

Training models that are identical in all aspects besides their tokenizer gives us an isolated environment to study their tokenizer's effects. However, for large models such a study is infeasible. To better understand the importance of tokenization in larger models, a different approach is required. We instead compare an input sentence with a sentence that is approximately equivalent semantically, but perturbed $n \in \{1, 2, 3, 4, 5\}$ times. We measured mean change in surprisal and JS divergence in the top 100 tokens, the results of which are displayed in Fig. 2. Note that in Fig. 2, the only original sentences that were considered were those that appeared in all $n = \{1, 2, 3, 4, 5\}$ perturbation datasets. This was done to remove any bias that could come with differences in sentence structure and content.

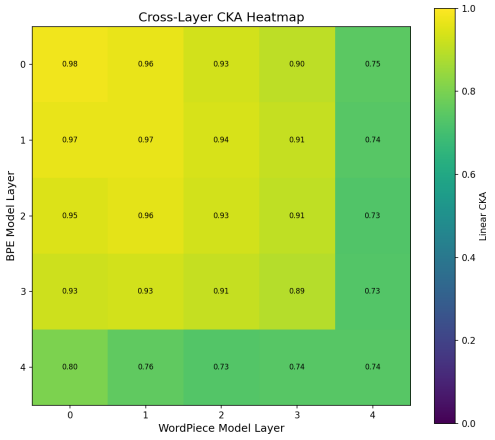

Figure 1: CKA similarity between the layers of RoBERTa models trained on BPE and WordPiece tokenizers, layer 0 being the input layer

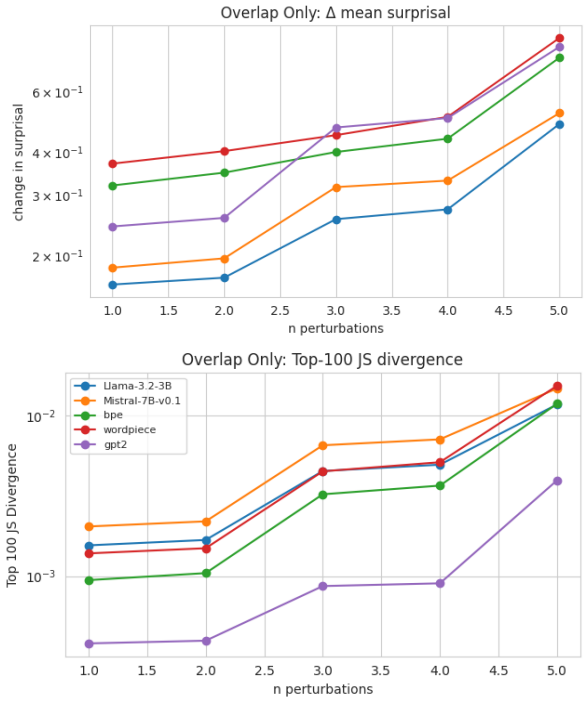

Figure 2: JS divergence and change in mean surprisal for $n \in \{1, 2, 3, 4, 5\}$ perturbations

These results show that both change in surprisal and top-100 JS divergence generally increase with the number of perturbations across models. These results indicate that even if an intervention does not alter the meaning of an input sentence, the model's output distribution still changes. Therefore, the models have a sensitivity to non-semantic factors.

Both experiments aimed to isolate tokenizer, either through training identical models on different tokenizers, or by providing the same inputs that semantically were approximately equal yet resulted in different input token sequences. In both cases, the model was not fully invariant to tokenization, despite the meaning of the inputs being stable. These results indicate a sensitivity to tokenization, where discrete segmentation differences can propagate to measurable shifts in downstream tasks, which can shape how well a model maintains stability.

5 Conclusions

This work presented two complementary studies of tokenizer importance. First, we trained two small RoBERTa models under identical conditions, differing only in the tokenizer. We analyzed the hidden layers using CKA similarity and showed that internal representations diverged, indicating that the tokenizer can act as an inductive bias shaping learned representations. Second, using semantic preserving perturbations of an input sentence, we observed a relationship between number of perturbations and next token distribution as well as increasing shifts in surprisal, indicating that edits can preserve meaning yet still induce systematic changes in model predictions.

Future work is twofold. First, scale up the empirical perturbation study to involve larger models, additional families of tokenizers, and analysis over more datasets and perturbation types to ensure that the observed trends are consistent. Additionally, we must extend the hidden representation similarity study not only through larger models but also relating tokenizer similarity to hidden representation similarity. The second direction we will take is to formalize the results into a framework that can characterize a model's input sensitivity, which is highly influential in applications such as knowledge extraction and explainability. We must be able to more clearly represent how strong an intervention is to the resulting change in the model's representations and next token distributions. Ultimately, we aim to measure tokenizer effects so that we can predict and improve model stability. The reported experiments showed that further study towards this goal is warranted.

References

Alahmari, S. S., Hall, L., Mouton, P. R., and Goldgof, D. (2025). Large language models robustness against perturbation: S. alahmari et al. *Scientific Reports*.

Altıntaş, G. S., Ehghaghi, M., Lester, B., Liu, F., Zhao, W., Ciccone, M., and Raffel, C. (2025). Toksuite: Measuring the impact of tokenizer choice

on language model behavior. *arXiv preprint arXiv:2512.20757*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gumma, V., Chitale, P. A., and Bali, K. (2025). Towards inducing long-context abilities in multilingual neural machine translation models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7158–7170.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2019). The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Jeong, D. P., Garg, S., Lipton, Z. C., and Oberst, M. (2024). Medical adaptation of large language and vision-language models: Are we making progress? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12143–12170.

Jiang, Y., Li, X., Zhu, G., Li, H., Deng, J., Han, K., Shen, C., Shi, Q., and Zhang, R. (2023). 6g non-terrestrial networks enabled low-altitude economy: Opportunities and challenges. *arXiv preprint arXiv:2311.09047*.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. (2019). Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMlR.

Kudo, T. and Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 conference on empirical methods in natural language processing: System demonstrations*, pages 66–71.

Lesci, P., Meister, C., Hofmann, T., Vlachos, A., and Pimentel, T. (2025). Causal estimation of tokenisation bias. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28325–28340.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Rauba, P., Wei, Q., and van der Schaar, M. (2024). Quantifying perturbation impacts for large language models. *arXiv preprint arXiv:2412.00868*.

Salazar, J., Liang, D., Nguyen, T. Q., and Kirchhoff, K. (2020). Masked language model scoring. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 2699–2712.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 1715–1725.

Smith, N. J. and Levy, R. (2013). The effect of word predictability on reading time is logarithmic. *Cognition*, 128(3):302–319.

Tseng, A. (2024). High-quality english sentences dataset. https://huggingface.co/datasets/agentlans/high-quality-english-sentences. Accessed: 2026-03-04.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Vrandečić, D. and Krötzsch, M. (2014). Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Wang, H., Ma, G., Yu, C., Gui, N., Zhang, L., Huang, Z., Ma, S., Chang, Y., Zhang, S., Shen, L., Wang, X., Zhao, P., and Tao, D. (2025). Are large language models really robust to word-level perturbations? *Transactions on Machine Learning Research*.

Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. (2024). Sweagent: Agent-computer interfaces enable automated software engineering. *Advances in Neural Information Processing Systems*, 37:50528–50652.

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.