False Friends Are *Not* Foes: Investigating Vocabulary Overlap in Multilingual Language Models

Julie Kallini, Dan Jurafsky, Christopher Potts, Martijn Bartelds
Stanford University
kallini@stanford.edu

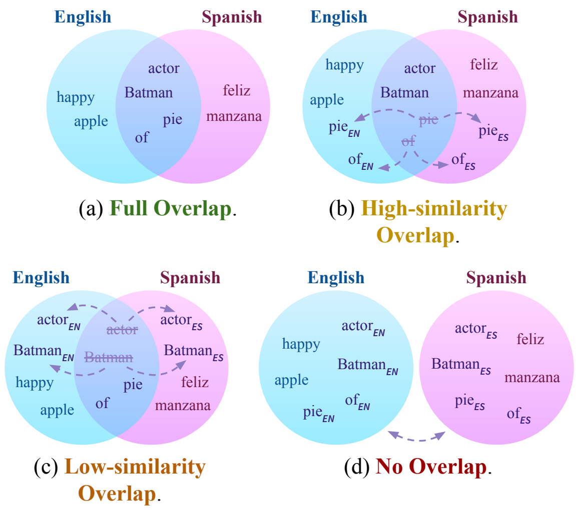
Abstract

Subword tokenizers trained on multilingual corpora naturally produce overlapping tokens across languages. Does token overlap facilitate cross-lingual transfer or instead introduce interference between languages? Prior work offers mixed evidence, partly due to varied setups and confounders, such as token frequency or subword segmentation granularity. To address this question, we devise a controlled experiment where we train bilingual autoregressive models on multiple language pairs under systematically varied vocabulary overlap settings. Crucially, we explore a new dimension to understanding how overlap affects transfer: the semantic similarity of tokens shared across languages. We first analyze our models' hidden representations and find that overlap *of any kind* creates embedding spaces that capture cross-lingual semantic relationships, while this effect is much weaker in models with disjoint vocabularies. On XNLI and XQuAD, we find that models with overlap outperform models with disjoint vocabularies, and that transfer performance generally improves as overlap increases. Overall, our findings highlight the advantages of token overlap in multilingual models and show that substantial shared vocabulary remains a beneficial design choice for multilingual tokenizers.

https://github.com/jkallini/false-friends

1 Introduction

Multilingual tokenizers are commonly trained on the concatenation of corpora from multiple languages (Conneau et al., 2020a; Xue et al., 2021), resulting in subword vocabularies with naturally overlapping tokens across languages. While some of these shared tokens may correspond to semantically aligned units across languages (e.g., cognates, named entities), others may arise from coincidental overlaps or have different meanings (e.g., false friends). Although prior work has demonstrated that token overlap can enhance zero-shot cross-lingual transfer (Pires et al., 2019; Conneau et al., 2020b), others report adverse effects depending on the end task (e.g., Limisiewicz et al., 2023). Some tokenization approaches have aimed to reduce overlap altogether (Chung et al., 2020; Liang et al., 2023). These contradictory studies lead us to ask: *when and how does the presence of overlapping tokens improve cross-lingual transfer?*



(a) **Full Overlap**. (b) **High-similarity Overlap**. (c) **Low-similarity Overlap**. (d) **No Overlap**.

Figure 1: A visualization of the four overlap settings used in our experiments. (a) **Full Overlap**: the two languages share the original tokenizer's native overlapping subwords. These include true cognates and named entities (e.g., *actor*, *Batman*) as well as false cognates or coincidental overlaps (e.g., *pie*, *of*). (b) **High-similarity Overlap**: only tokens with the highest cross-lingual semantic similarity are shared. (c) **Low-similarity Overlap**: only tokens with the lowest cross-lingual semantic similarity are shared. (d) **No Overlap**: the two languages' vocabularies are completely disjoint.

We answer this question by training bilingual autoregressive models on data from six language pairs, each under four controlled vocabulary overlap settings (Figure 1). In contrast to prior work, we distinguish different types of overlap

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 21138–21154

based on semantic similarity of the tokens in the two languages, as semantic alignment has been shown to impact cross-lingual transfer (Cao et al., 2020; Deshpande et al., 2022; Hua et al., 2024), while holding subword segmentation granularity and token frequency distributions fixed. Within pre-trained models, we find that token overlap enables the embedding spaces of the two languages to capture cross-lingual semantic relationships—an effect that is substantially weaker in models with disjoint vocabularies. When testing zero-shot transfer between languages on the XNLI and XQuAD downstream tasks, models with any amount of overlap consistently outperform models with no overlap, and transfer performance generally improves as overlap increases. We find that tokens with shared meanings across languages contribute most to transfer performance, though any overlap is beneficial. Our findings offer practical guidance on the design of future multilingual tokenizers.

2 Background and Related Work

Vocabulary Overlap. Research on cross-lingual transfer has revealed both advantages and challenges of subword overlap in multilingual models. On the positive side, prior work showed that token overlap provides moderate gains for zero-shot transfer in multilingual BERT (mBERT) on language understanding tasks (Pires et al., 2019; Wu and Dredze, 2019; Dufter and Schütze, 2020). Conneau et al. (2020b) more closely examined token overlap using three vocabulary-sharing schemes in bilingual encoders and observed that overlap provided marginal improvements on XNLI, NER, and parsing. K et al. (2020) similarly reported minimal performance differences due to wordpiece overlap in bilingual BERT models.

More recently, Limisiewicz et al. (2023) found that while overlap can benefit sentence-level tasks and NER, it may degrade performance on syntactic tasks. Similarly, Zhang et al. (2023) show that multilingual corpora contain unexpectedly high levels of overlap, largely due to code-switching and shared vocabularies, which may help explain cross-lingual transfer in dense retrieval models. Zhang et al. (2025) extend overlap by merging subwords with different forms but similar meanings into "semantic tokens," preserving downstream performance with smaller vocabularies. Hämmerl et al. (2025) show that similarity- or alignment-weighted overlap correlates with cross-lingual transfer across different scripts. Other related work shows that multilingual tokenizers often over-segment low-resource languages, artificially inflating subword overlap (Rust et al., 2021; Petrov et al., 2023; Ahia et al., 2023). This over-segmentation reduces efficiency and degrades representation quality.

Taken together, these studies paint an unclear picture: while vocabulary overlap can create cross-lingual anchors that facilitate transfer, it may introduce interference across languages that hinders modeling. Moreover, the conditions under which overlap is beneficial remain insufficiently explored. Unlike prior work, we focus on how the semantic similarity of shared tokens affects performance, while carefully controlling for confounders like subword segmentation granularity and token frequency distributions.

Tokenizer Design. Vocabulary overlap has likewise been a central consideration in tokenizer design. Chung et al. (2020) and Liang et al. (2023) use clustering methods to de-emphasize token overlap between lexically distinct languages, citing K et al. (2020) for the thesis that overlap is not the principal factor in multilingual model effectiveness. In contrast, Patil et al. (2022) highlight the importance of overlap for transfer and propose a method to promote token overlap between high- and low-resource languages. At the extreme, byte- and character-level models (e.g., CANINE, Clark et al., 2022; ByT5, Xue et al., 2022; MrT5, Kallini et al., 2025; BLT, Pagnoni et al., 2025; H-Net, Hwang et al., 2025) eliminate subword tokenization altogether. This maximizes vocabulary overlap but comes at a cost to efficiency, presenting unique engineering challenges.

3 Approach: Controlled Overlap Settings

To systematically vary the vocabulary overlap between two languages according to our four experimental settings (see Figure 1), we denote a base tokenizer $\mathcal{T}$ with vocabulary V of size N. We assume that $V = \{0, 1, \dots, N-1\}$, i.e. that each token in V is represented by an integer index from 0 to $N-1$. Two languages L_1 and L_2 have corpora C_1 and C_2, respectively, which we tokenize using $\mathcal{T}$. Let $V_1 = \{\text{unique tokens in } C_1\} \subseteq V$ and $V_2 = \{\text{unique tokens in } C_2\} \subseteq V$. In other words, V_1 and V_2 are the individual vocabularies of L_1 and L_2, respectively. Thus, when tokenizing C_1 and C_2, the *native overlap* of $\mathcal{T}$ is the set $O = V_1 \cap V_2$, and the *effective vocabulary size* of

$\mathcal{T}$ is $N_{\text{eff}} = |V_1| + |V_2| - |O|$.

Given a token sequence $X = [x_1, x_2, \ldots, x_n]$, where $x_i \in V$, from language $\ell \in \{L_1, L_2\}$, we define a modified tokenizer $\mathcal{T}'$ that produces $X' = [x'_1, x'_2, \ldots, x'_n]$, where each

$$x'_i = \begin{cases} x_i + N, & \ell = L_2 \text{ and } x_i \notin O', \\ x_i, & \text{otherwise.} \end{cases}$$

Here, $O' \subseteq O$ denotes the subset of tokens we choose to share under a given setting: for L_1, all tokens remain unchanged, and for L_2, tokens in O' remain unchanged while all others are offset by N. This guarantees that L_1 and L_2 only share O', and $\mathcal{T}'$ has a new effective vocabulary size $N'_{\text{eff}} = |V_1| + |V_2| - |O'|$. The four choices of O' define our four settings, listed below.

Full Overlap. $O' = O$. Since this only renames certain tokens $x_i \notin O$ from L_2, $\mathcal{T}'$ is behaviorally equivalent to $\mathcal{T}$, and $N'_{\text{eff}} = |V_1| + |V_2| - |O|$.

High-similarity Overlap. $O' = O_{\text{hi}}$, where $O_{\text{hi}} \subseteq O$ is the set of tokens whose meanings align closely between L_1 and L_2. Only these tokens remain shared, so $N'_{\text{eff}} = |V_1| + |V_2| - |O_{\text{hi}}|$.

Low-similarity Overlap. $O' = O_{\text{lo}}$, where $O_{\text{lo}} \subseteq O$ is the set of tokens whose meanings differ across L_1 and L_2. Only these tokens remain shared, so $N'_{\text{eff}} = |V_1| + |V_2| - |O_{\text{lo}}|$.

No Overlap. $O' = \emptyset$. Since no tokens are shared, $N'_{\text{eff}} = |V_1| + |V_2|$.

The details for the semantic partitioning of O into O_{hi} and O_{lo} are presented in the next section.

4 Implementation Details

Datasets. We use CCMatrix (Schwenk et al., 2021), a large collection of high-quality web-mined parallel texts, for bilingual model pre-training. This allows us to control for the content and the approximate quantity of data in each language. We train on six language pairs: English–Spanish, English–German, English–Turkish, English–Chinese, English–Arabic, and English–Swahili. English is included in every pair to reflect realistic training scenarios, as English is typically the dominant language in multilingual pre-training datasets. The second language is selected to cover various language families, scripts, and typological distances from English. The pre-training corpus for each pair is constructed by shuffling and interleaving sentences from both languages.

Tokenizer and Overlap Partitioning. Our base tokenizer $\mathcal{T}$ is the multilingual XLM-R tokenizer (Conneau et al., 2020a), which uses SentencePiece (Kudo and Richardson, 2018) with a unigram LM (Kudo, 2018). We found that it offers more effective compression across languages than other tokenizers (see Appendix A). To divide the native overlap O into high- and low-similarity subsets (O_{hi} and O_{lo}), we rank tokens in O by their semantic similarity across languages. For each token $t \in O$, we extract 100 occurrences from C_1 (the CCMatrix corpus for L_1), pass the sentences through XLM-R, and mean-pool the layer-l contextual embeddings of t to obtain a static embedding e_1 (following Bommasani et al., 2020). The layer l is pre-determined by a sweep we conducted on sets of cognates and non-cognates, as detailed in Appendix B. We repeat this for C_2 (the corpus for L_2) to compute e_2. The cosine similarity between e_1 and e_2 serves as the token's cross-lingual similarity score. We rank tokens in O by these scores, assigning the top half to O_{hi} and the bottom half to O_{lo}. For detailed corpus statistics and overlap metrics for each setting, refer to Appendix C.

Models. For each language pair and vocabulary setting, we pre-train a separate model, resulting in 24 bilingual models in total. All models are autoregressive Transformers (Vaswani et al., 2017) with 85M non-embedding parameters, equivalent in size to GPT-2 Small (Radford et al., 2019). We train these models due to their architectural similarity with modern LLMs. See Appendix D for additional architecture and optimization details.

5 Embedding Similarity Analysis

As a first step in analyzing our pre-trained models, we test how sharing semantically similar or dissimilar tokens influences the model's learned representations. We take the 500 most and least similar overlapping tokens for each language pair, ranked using XLM-R as described in the previous section. From a middle layer ($l = 6$) of our own models, we extract contextual embeddings for each token to construct a single static embedding of the token for each language using the same method as before. We then ask whether models learn more similar representations for high-similarity tokens and more distinct ones for low-similarity tokens. Crucially, whether these tokens are shared depends on the overlap setting: in the *High-similarity Overlap* condition, the top 500 tokens are shared; in the

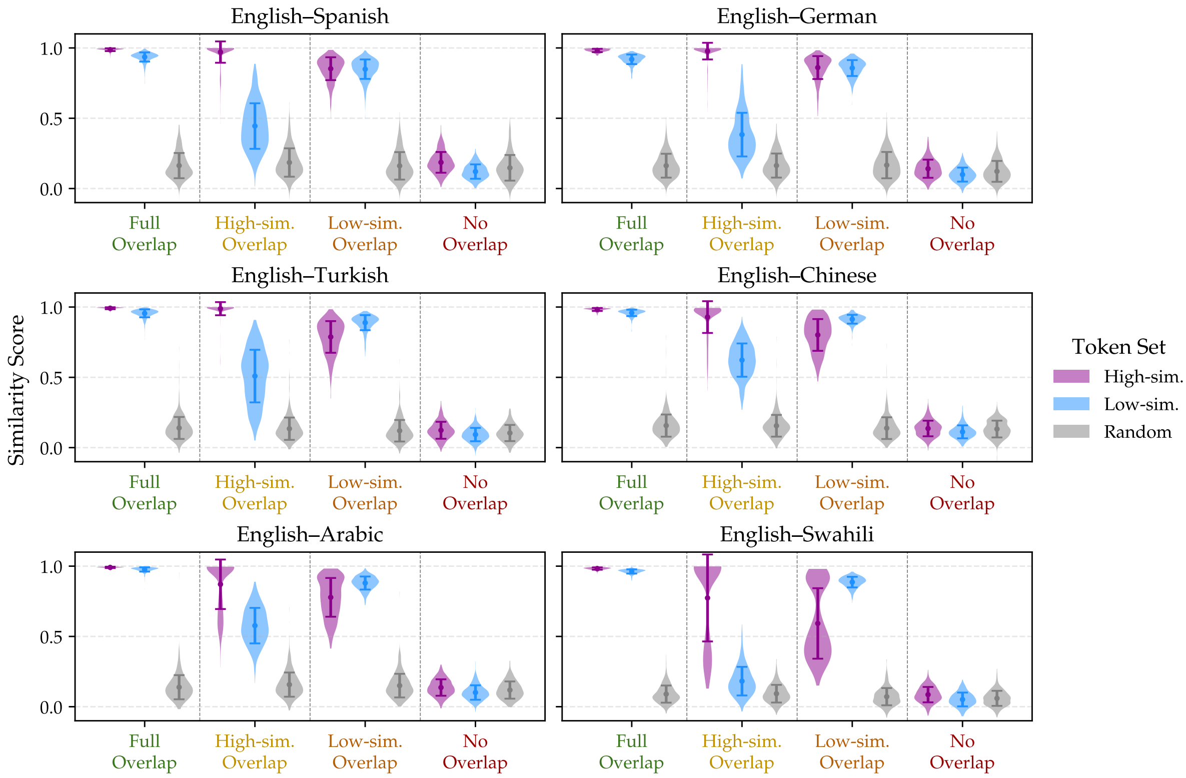


Figure 2: Embedding similarity analysis on pre-trained models for each language pair and vocabulary setting. Cosine similarity is used to measure similarity of tokens in L_1 and L_2 for a given language pair. The high-sim. token set (purple) should have similar meanings; the low-sim. token set (blue) should have dissimilar meanings; the random token set (gray) should not share form or meaning, and are shown as a control for anisotropy.

Low-similarity Overlap condition, the bottom 500 are shared. To control for the high baseline cosine similarities observed in Transformer embeddings due to anisotropy (Ethayarajh, 2019), we additionally measure similarity scores for 500 randomly selected non-overlapping token pairs.

Results. Figure 2 summarizes the results across all language pairs. With the exception of the *Low-similarity Overlap* setting for English–Spanish and English–German, the difference between the high- and low-similarity token sets was statistically significant for every language pair and overlap condition (unpaired t-tests, all Bonferroni-corrected $p < .05$). The effect size (Cohen's d) varied with the overlap condition (see Table 5 for all effect sizes). In the *Full Overlap* and *High-similarity Overlap* settings, high-similarity tokens consistently scored higher than low-similarity tokens, yielding very large effects ($d \in [1.3, 5.1]$). Even in the *No Overlap* setting, the high-similarity tokens scored higher than low-similarity tokens, though effect sizes were smaller ($d \in [0.5, 1.0]$), suggesting that some degree of semantic alignment persists even without shared lexical anchors.

In contrast, the *Low-similarity Overlap* setting revealed a split in results based on language family. For the closely related language pairs English–Spanish and English–German, no significant differences were observed (Bonferroni-corrected $p = 1$, $d \approx 0$). However, for more typologically distant language pairs (English–Turkish, English–Chinese, English–Arabic, English–Swahili), the effect reversed: low-similarity tokens scored higher than high-similarity tokens, with large negative effect sizes ($d \in [-1.6, -1.0]$). These reversals indicate that in the *Low-similarity Overlap* setting, tokens that do not share meaning become aligned in the embedding space simply because they are shared in the vocabulary, producing misleading or inverted similarity effects. This demonstrates that the *type* of overlap—whether it links semantically similar or dissimilar tokens—critically shapes how cross-lingual models align token representations. Effects are especially pronounced for more distant language pairs, where there is less contextual signal available to counteract the bias introduced by shared but semantically unrelated tokens.

Language Pair	Overlap Setting	XNLI Accuracy (%)		XQuAD F1 / EM (%)	
		Test (L_1)	Test (L_2)	Test (L_1)	Test (L_2)
English–Spanish	Full	**78.78**	**74.59**	63.83 / 51.85	**52.84 / 36.47**
	High-sim.	**78.52**	**73.99**	63.42 / 53.03	**48.60 / 31.85**
	Low-sim.	**79.18**	**74.55**	63.52 / 51.93	**51.57 / 36.13**
	No Overlap	76.73	42.67	62.66 / 51.43	7.45 / 0.59
English–German	Full	77.49	**69.44**	62.09 / 50.42	**45.06 / 31.18**
	High-sim.	78.40	**69.98**	62.24 / 51.43	**45.32 / 32.52**
	Low-sim.	78.26	**69.30**	62.34 / 50.92	**41.79 / 27.39**
	No Overlap	78.08	35.01	61.96 / 49.83	5.09 / 0.25
English–Turkish	Full	77.54	**49.46**	61.03 / 49.75	**22.16 / 11.85**
	High-sim.	**78.56**	**56.11**	61.75 / 50.50	**21.20 / 12.69**
	Low-sim.	78.40	**52.32**	62.02 / 51.01	**20.41 / 11.60**
	No Overlap	77.41	37.82	62.71 / 51.18	5.71 / 1.34
English–Chinese	Full	**78.48**	**63.29**	62.07 / 50.42	**26.10 / 16.39**
	High-sim.	77.15	**60.42**	62.75 / 50.50	**23.56 / 16.30**
	Low-sim.	77.13	**55.87**	62.77 / 51.09	**14.24 / 3.70**
	No Overlap	77.03	36.35	62.93 / 51.68	2.70 / 0.42
English–Arabic	Full	77.41	**61.32**	62.52 / 50.25	**29.58 / 17.65**
	High-sim.	77.70	**61.14**	63.31 / 51.51	**28.96 / 16.64**
	Low-sim.	77.60	**49.40**	62.58 / 50.50	**9.46 / 2.27**
	No Overlap	77.72	32.93	61.09 / 50.34	6.14 / 0.92
English–Swahili	Full	75.11	**48.24**	—	—
	High-sim.	74.55	**49.26**	—	—
	Low-sim.	75.23	**43.49**	—	—
	No Overlap	75.69	33.75	—	—

Table 1: Downstream performance across language pairs and vocabulary overlap settings. For XNLI, we report accuracy; for XQuAD, we report F1 and exact match (EM). Settings significantly different from *No Overlap* are in bold (see Table 6 for all p-values).

6 Downstream Task Performance

We further fine-tune and evaluate our models on two downstream tasks, namely, natural language inference (NLI) and question answering (QA), in a standard zero-shot transfer setup. For NLI, we train on English MultiNLI (Williams et al., 2018) and evaluate on XNLI (Conneau et al., 2018). For QA, we train on English SQuAD (Rajpurkar et al., 2016) and evaluate on XQuAD (Artetxe et al., 2020). Fine-tuning hyperparameters and optimization details are provided in Appendix E.

Results. Results for both tasks are shown in Table 1. To compare XNLI accuracies and XQuAD exact match (EM) scores across models, we conducted pairwise McNemar tests (see Table 6). While L_1 (English) evaluation results are reported for completeness, we center the discussion here on L_2 transfer, which is the main focus of this work.

On L_2 transfer, the *No Overlap* models performed substantially worse than all other overlap settings across every language pair for both downstream tasks (all $p < .05$). This confirms that some degree of shared vocabulary is always beneficial for cross-lingual transfer. Comparisons between overlap types show more subtle patterns. *Full Overlap* and *High-similarity Overlap* achieved the strongest transfer performance overall: *Full* was best in six of eleven L_2 evaluations (both XNLI accuracy and XQuAD F1/EM), while *High-similarity* was best in four evaluations. However, differences between these two settings were not significant in seven of the eleven cases (all $p > .05$). By contrast, both *Full* and *High-similarity Overlap* consistently outperformed *Low-similarity Overlap*: *Full* was stronger in ten of eleven evaluations (seven significant; all $p < .05$), and *High-similarity* was stronger in nine of eleven (seven significant; all $p < .05$). This advantage is notable given that high-similarity tokens make up only 10–20% of training and evaluation corpora, whereas low-similarity tokens account for as much as 80% (see Appendix C).

These results show that while any overlap helps, sharing semantically similar tokens is far more impactful. The language pair also matters: for languages closely related to English, such as Spanish and German, *High-* and *Low-similarity Overlap* performed comparably, whereas for more distant languages, *High-similarity Overlap* gave a clearer advantage. In Chinese and Arabic, the use of a different script from English reduces the value of cross-lingual transfer in the *Low-similarity Overlap* setting. Here, semantically aligned tokens have an outsized impact, as they are often English words introduced through code-switching. We provide the full list of overlapping tokens with their similarity scores in our repository.

7 Conclusion

In this paper, we present a detailed study of vocabulary overlap in multilingual language models. Our experimental design isolates the effect of overlap by controlling for token frequency and subword segmentation quality. We also uniquely disentangle how semantically similar or dissimilar vocabulary overlap affect multilingual representations and task transfer. While prior work has raised concerns that highly polysemantic tokens from vocabulary sharing may hinder performance, we find that overlap (1) promotes alignment of the embedding spaces between languages in bilingual models and (2) enables cross-lingual transfer on downstream tasks. Overlapping tokens with the same meaning across languages contribute most, though any overlap proves beneficial. We therefore argue that, rather than reducing overlap, tokenizer development should focus on other determinants of quality, such as per-language compression rates.

8 Acknowledgments

The authors would like to thank Róbert Csordás, Tomasz Limisiewicz, and Ekaterina Shutova for helpful comments at different stages of the project. We would also like to thank the members of the Stanford NLP Group, the Jurafsky Lab Group, and the anonymous reviewers for useful discussions. Julie Kallini is supported by a National Science Foundation Graduate Research Fellowship under grant number DGE-2146755.

9 Limitations

Our study analyzes six language pairs spanning diverse language families. Each language pair requires pre-training four models, which is computationally expensive. While our selection of languages provides meaningful breadth, with additional compute resources, future work could extend the analysis to additional language pairs, particularly more low-resource languages. We also focus on English-centric pairs, reflecting common multilingual pre-training scenarios where English is the dominant language. Exploring overlap effects in non-English pairings would complement our findings. In addition, we use a single, widely adopted tokenizer (XLM-R) to control for tokenizer quality across conditions. Although this choice allows for a clean comparison of overlap settings, future work could examine how overlap interacts with tokenizers of varying quality or design choices to further contextualize our results. Finally, following Dufter and Schütze (2020), future work could explore whether extended training or different parameter budgets further affect cross-lingual generalization under the different overlap settings.

References

Orevaoghene Ahia, Sachin Kumar, Hila Gonen, Jungo Kasai, David Mortensen, Noah Smith, and Yulia Tsvetkov. 2023. Do all languages cost the same? tokenization in the era of commercial language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9904–9923, Singapore. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Steven Cao, Nikita Kitaev, and Dan Klein. 2020. Multilingual alignment of contextual word representations. In *International Conference on Learning Representations*.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online. Association for Computational Linguistics.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020a. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020b. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Ameet Deshpande, Partha Talukdar, and Karthik Narasimhan. 2022. When is BERT multilingual? isolating crucial ingredients for cross-lingual transfer. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3610–3623, Seattle, United States. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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, Minneapolis, Minnesota. Association for Computational Linguistics.

Philipp Dufter and Hinrich Schütze. 2020. Identifying elements essential for BERT's multilinguality. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4423–4437, Online. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv: 2407.21783*.

Katharina Hämmerl, Tomasz Limisiewicz, Jindřich Libovický, and Alexander Fraser. 2025. Beyond literal token overlap: Token alignability for multilinguality. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 756–767, Albuquerque, New Mexico. Association for Computational Linguistics.

Tianze Hua, Tian Yun, and Ellie Pavlick. 2024. mOthello: When do cross-lingual representation alignment and cross-lingual transfer emerge in multilingual models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1585–1598, Mexico City, Mexico. Association for Computational Linguistics.

Sukjun Hwang, Brandon Wang, and Albert Gu. 2025. Dynamic chunking for end-to-end hierarchical sequence modeling. *Preprint*, arXiv:2507.07955.

Karthikeyan K, Zihan Wang, Stephen Mayhew, and Dan Roth. 2020. Cross-lingual ability of multilingual BERT: An empirical study. In *International Conference on Learning Representations*.

Julie Kallini, Shikhar Murty, Christopher D. Manning, Christopher Potts, and Róbert Csordás. 2025. MrT5: Dynamic token merging for efficient byte-level language models. In *The Thirteenth International Conference on Learning Representations*.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 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, Brussels, Belgium. Association for Computational Linguistics.

Els Lefever, Sofie Labat, and Pranaydeep Singh. 2020. Identifying cognates in English-Dutch and French-Dutch by means of orthographic information and cross-lingual word embeddings. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4096–4101, Marseille, France. European Language Resources Association.

Davis Liang, Hila Gonen, Yuning Mao, Rui Hou, Naman Goyal, Marjan Ghazvininejad, Luke Zettlemoyer, and Madian Khabsa. 2023. XLM-V: Overcoming the vocabulary bottleneck in multilingual masked language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13142–13152, Singapore. Association for Computational Linguistics.

Tomasz Limisiewicz, Jiří Balhar, and David Mareček. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5661–5681, Toronto, Canada. Association for Computational Linguistics.

Artidoro Pagnoni, Ramakanth Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason E Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srini Iyer. 2025. Byte latent transformer: Patches scale better than tokens. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9238–9258, Vienna, Austria. Association for Computational Linguistics.

Vaidehi Patil, Partha Talukdar, and Sunita Sarawagi. 2022. Overlap-based vocabulary generation improves cross-lingual transfer among related languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 219–233, Dublin, Ireland. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. In *Advances in Neural Information Processing Systems*, volume 36, pages 36963–36990. Curran Associates, Inc.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Ms, OpenAI.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. 2021. How good is your tokenizer? on the monolingual performance of multilingual language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3118–3135, Online. Association for Computational Linguistics.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, Armand Joulin, and Angela Fan. 2021. CCMatrix: Mining billions of high-quality parallel sentences on the web. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6490–6500, Online. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

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.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

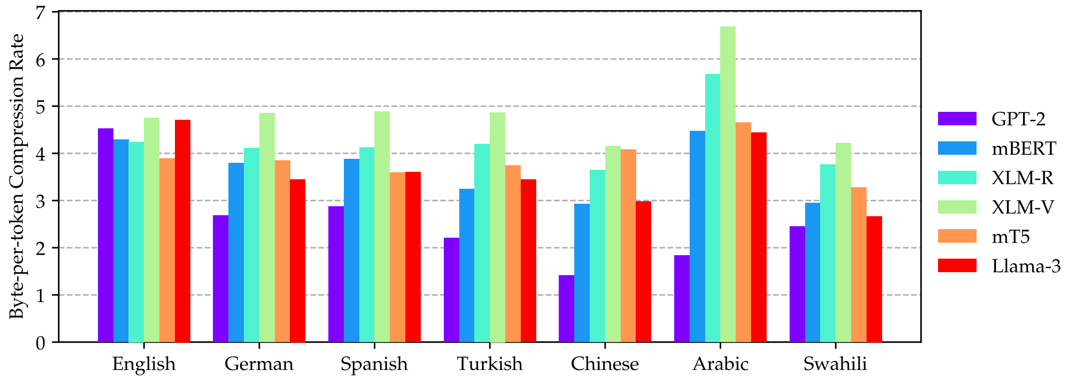
Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Crystina Zhang, Jing Lu, Vinh Q. Tran, Tal Schuster, Donald Metzler, and Jimmy Lin. 2025. Tomato, tomahto, tomate: Do multilingual language models understand based on subword-level semantic concepts? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1821–1837, Albuquerque, New Mexico. Association for Computational Linguistics.

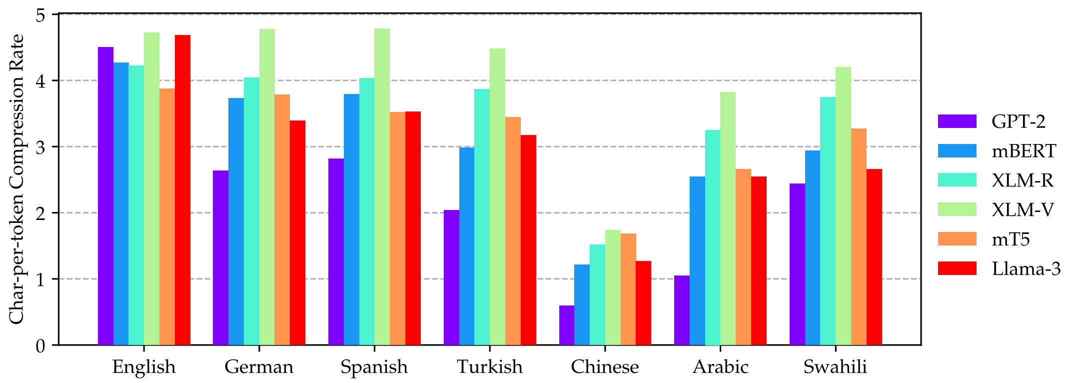
Xinyu Zhang, Kelechi Ogueji, Xueguang Ma, and Jimmy Lin. 2023. Toward best practices for training multilingual dense retrieval models. *ACM Trans. Inf. Syst.*, 42(2).

A Tokenizer Compression Rates

We consider six multilingual tokenizers as candidates for our base tokenizer $\mathcal{T}$: GPT-2 (Radford et al., 2019), mBERT (Devlin et al., 2019), XLM-R (Conneau et al., 2020a), XLM-V (Liang et al., 2023), mT5 (Xue et al., 2021), and Llama 3 (Grattafiori et al., 2024). We compute byte-per-token and character-per-token compression rates for the seven languages involved in our study, using samples from the multilingual C4 corpus (Raffel et al., 2020). As shown in Figure 3, XLM-V achieves the best compression but has an extremely large vocabulary (1M tokens), making it impractical for our setup. XLM-R's compression is competitive with XLM-V's at a much smaller vocabulary size (250k tokens), making it a suitable choice for our controlled experiments.



(a) Byte-per-token compression rates.



(b) Character-per-token compression rates.

Figure 3: Byte-per-token and character-per-token compression rates for English, German, Spanish, Turkish, Chinese, Arabic, and Swahili, for six different tokenizers.

B Layer Selection

We select the Transformer layer that best distinguishes between semantically similar and dissimilar tokens in a controlled setup. We use manually annotated data from the English–Dutch cognate detection dataset of Lefever et al. (2020), as well as English–Dutch parallel texts from CCMatrix. From the cognate detection dataset, we extract a list of both cognates and non-cognates, which we tokenize using XLM-R's tokenizer. We remove any words that are tokenized into more than one token, as well as non-overlapping tokens and tokens that appear fewer than 100 times in the parallel texts. One author then manually verified that no cognates remained in the non-cognate subset. For each remaining token, we sample 100 occurrences per language from the English–Dutch parallel texts. We then pass these tokens through XLM-R's layers $l \in \{1, \ldots, 12\}$ and average the layer-l embeddings to obtain a static embedding per token for each language. We compute the cosine similarity between the static embeddings at each layer and rank the tokens by similarity. To quantify the capacity of each layer to distinguish between cognates and non-cognates, we sweep through every possible threshold n in the ranked list. Specifically, we label the top-n tokens as *predicted cognates* and the remaining tokens as *predicted non-cognates*, measuring the classification accuracy against our gold labels. The highest classification accuracy over all n is taken as the oracle score for layer l. As shown in Figure 4, the highest classification accuracy was obtained using layer 5, and we therefore use layer 5 to rank tokens in O by their semantic similarity across all language pairs.

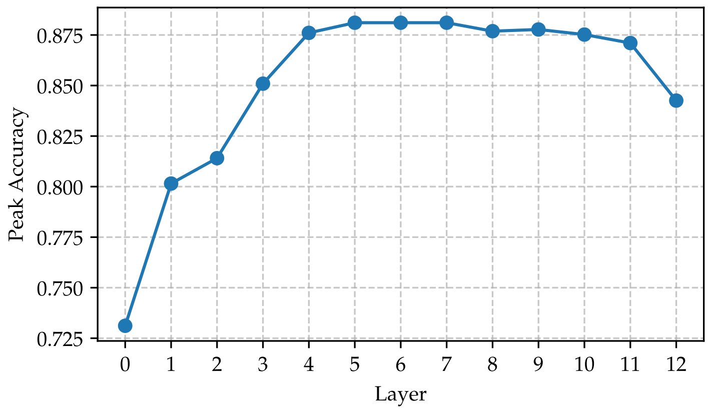


Figure 4: Results of our layer sweep on XLM-R using English–Dutch data from Lefever et al. 2020.

C Overlap Metrics for All Datasets

Below we report corpus statistics *after* applying the four overlap manipulations described in Section 3. For each language pair L_1 and L_2, we start with corpora C_1 and C_2 taken from CCMatrix and tokenize them with the XLM-R SentencePiece tokenizer $\mathcal{T}$. This yields individual language vocabulary sets $V_1 = \{\text{unique tokens in } C_1\}$ and $V_2 = \{\text{unique tokens in } C_2\}$. An overlap setting remaps token indices, producing new language vocabularies V_1' and V_2', with $|V_1'| = |V_1|$ and $|V_2'| = |V_2|$. Their intersection, $O' = V_1' \cap V_2'$ contains the tokens shared under that setting. Thus, the total effective vocabulary size of $\mathcal{T}'$ is $N'_{\text{eff}} = |V_1| + |V_2| - |O'|$. With these definitions in place, we now define two overlap metrics:

1. **Type overlap (IoU).** The Jaccard similarity of the setting-specific vocabularies V_1' and V_2' is

$$J(V_1', V_2') = \frac{|V_1' \cap V_2'|}{|V_1' \cup V_2'|} = \frac{|O'|}{|V_1'| + |V_2'| - |O'|} = \frac{|O'|}{N'_{\text{eff}}}.$$

2. **Frequency-weighted overlap.** To quantify how often the shared tokens are *used* in each corpus, we compute, for $i \in \{1, 2\}$,

$$F_i = \frac{\sum_{t \in O'} \text{count}_i(t)}{\sum_{t \in V_i'} \text{count}_i(t)},$$

where $\text{count}_i(t)$ is the frequency of token t in corpus C_i', where C_i' is the corpus C_i after applying the token remapping under the given setting. Thus F_i is the proportion of running tokens in C_i' that belong to the shared vocabulary O'.

Table 2 presents these statistics in the tokenized CCMatrix pre-training data for every language pair under each overlap condition. We also report frequency-weighted overlap metrics with respect to the XNLI and XQuAD training and test datasets in Table 3. Remarkably, although the *High-* and *Low-similarity Overlap* settings contain the same number of overlapping token types, the latter has substantially higher frequency-weighted overlaps in the pre-training corpora as well as the downstream task datasets.

Language Pair	Setting	$\lvert V_1 \rvert$	$\lvert V_2 \rvert$	$\lvert O' \rvert$	N'_{eff}	IoU (%)	F_1 (%)	F_2 (%)
English–Spanish	Full Overlap	78,469	78,381	73,455	83,395	88.08	99.88	98.98
	High-sim. Overlap			22,103	134,747	16.40	21.47	19.24
	Low-sim. Overlap			22,101	134,749	16.40	77.32	66.80
	No Overlap			0	156,850	0.00	0.00	0.00
English–German	Full Overlap	83,126	83,884	75,922	91,088	83.35	96.73	99.06
	High-sim. Overlap			20,594	146,416	14.07	20.37	18.68
	Low-sim. Overlap			20,592	146,418	14.06	75.82	68.05
	No Overlap			0	167,010	0.00	0.00	0.00
English–Turkish	Full Overlap	65,665	69,703	58,724	76,644	76.62	99.99	86.58
	High-sim. Overlap			13,906	121,462	11.45	19.92	17.20
	Low-sim. Overlap			13,907	121,461	11.45	76.81	43.03
	No Overlap			0	135,368	0.00	0.00	0.00
English–Chinese	Full Overlap	67,754	73,491	57,102	84,143	67.86	99.99	71.09
	High-sim. Overlap			12,598	128,647	9.79	22.59	9.26
	Low-sim. Overlap			12,599	128,646	9.79	73.04	19.20
	No Overlap			0	141,245	0.00	0.00	0.00
English–Arabic	Full Overlap	69,129	68,975	57,084	81,020	70.46	96.11	61.02
	High-sim. Overlap			9,963	128,141	7.78	20.39	9.87
	Low-sim. Overlap			9,963	128,141	7.78	67.56	8.19
	No Overlap			0	138,104	0.00	0.00	0.00
English–Swahili	Full Overlap	45,699	41,956	37,275	50,380	73.99	97.67	79.55
	High-sim. Overlap			4,733	82,922	5.71	20.44	17.35
	Low-sim. Overlap			4,734	82,921	5.71	51.36	39.90
	No Overlap			0	87,655	0.00	0.00	0.00

Table 2: Token statistics for the CCMatrix pre-training corpora: native vocabulary sizes ($\lvert V_1 \rvert$, $\lvert V_2 \rvert$), overlap size ($\lvert O' \rvert$), the resulting effective vocabulary size (N'_{eff}), and percentage-based overlap metrics (IoU, F_1, F_2) reported for every language pair and overlap setting.

D Pre-training Experiment Details

D.1 Model Architectures

All of our models are autoregressive Transformers with a similar architecture to GPT-2 (Radford et al., 2019) with 12 layers, 12 attention heads, $d_{\text{model}} = 768$, and $d_{\text{ff}} = 3072$. The only change we make to the standard GPT-2 architecture is the addition of rotary position embeddings (RoPE, Su et al., 2024), since this is the positional encoding method most often used in modern LLMs. The total non-embedding parameter count for all models is 85M, equivalent to the original GPT-2.

To isolate the effect of vocabulary overlap, we tokenize the data once and vary only which tokens are shared, which necessarily results in different vocabulary sizes across settings. Thus, the total model parameters varies based on the setting and language pair. To minimize unnecessary parameters, we prune the vocabulary to only retain tokens that appear in the CCMatrix corpus. Table 4 reports the resulting vocabulary sizes and total parameter counts for every setting and language pair. For the English–Spanish and English–German pairs, the retained vocabularies are marginally larger than the effective sizes N_{eff} reported in Table 2. This discrepancy occurs because the full CCMatrix corpora—on which the pruning was based—contain more tokens than the 6.6 billion tokens ultimately used for pre-training; consequently, a small subset of the embedding matrix remained unused during training.

Here, we note that no single setting can claim an a priori advantage based solely on vocabulary size. Larger vocabularies benefit from more model parameters but have higher upper bounds on perplexity and receive fewer gradient updates per embedding.

D.2 Optimization

We train with an effective batch size of 64 sequences, each 1024 tokens long, for a per-step token count $2^{16} = 65,536$ tokens. The device batch size is 8 sequences. Each model is trained for a total of 100,000 gradient steps using the AdamW optimizer. The learning rate linearly warms up to $2.5\text{e}{-4}$ during the first 5,000 steps, then follows a cosine decay.

Language Pair	Setting	XNLI			XQuAD		
		Train (L_1)	Test (L_1)	Test (L_2)	Train (L_1)	Test (L_1)	Test (L_2)
English–Spanish	Full Overlap	100.00	100.00	99.78	99.99	100.00	99.79
	High-sim. Overlap	23.89	19.25	17.69	19.85	19.67	16.29
	Low-sim. Overlap	75.16	79.81	69.71	79.06	79.19	72.20
	No Overlap	0.00	0.00	0.00	0.00	0.00	0.00
English–German	Full Overlap	100.00	100.00	99.50	99.98	100.00	99.46
	High-sim. Overlap	22.23	16.85	15.67	17.16	17.02	14.91
	Low-sim. Overlap	77.26	82.65	71.09	82.13	82.32	73.23
	No Overlap	0.00	0.00	0.00	0.00	0.00	0.00
English–Turkish	Full Overlap	99.99	99.99	86.35	99.95	99.95	87.06
	High-sim. Overlap	22.97	17.48	14.90	16.77	16.77	13.61
	Low-sim. Overlap	73.49	78.62	43.74	78.48	78.36	46.09
	No Overlap	0.00	0.00	0.00	0.00	0.00	0.00
English–Chinese	Full Overlap	99.98	99.99	71.08	99.93	99.94	74.23
	High-sim. Overlap	21.92	22.38	8.47	17.92	17.71	4.52
	Low-sim. Overlap	73.46	72.85	18.35	76.38	76.57	18.46
	No Overlap	0.00	0.00	0.00	0.00	0.00	0.00
English–Arabic	Full Overlap	99.96	99.95	61.19	99.93	99.94	61.60
	High-sim. Overlap	21.52	21.54	10.39	18.03	18.32	6.73
	Low-sim. Overlap	70.08	69.84	7.46	73.43	73.33	7.64
	No Overlap	0.00	0.00	0.00	0.00	0.00	0.00
English–Swahili	Full Overlap	97.56	97.34	79.13	—	—	—
	High-sim. Overlap	22.81	18.04	16.03	—	—	—
	Low-sim. Overlap	48.01	51.34	41.00	—	—	—
	No Overlap	0.00	0.00	0.00	—	—	—

Table 3: Frequency-weighted overlap in the XNLI and XQuAD datasets for each language pair and vocabulary overlap setting. Higher values indicate a larger proportion of running tokens that come from the shared set O'.

Because the batch size and number of steps are identical across settings, each model processes 6.6 billion tokens in total. The required number of passes through CCMatrix therefore depends on the parallel corpus size: one epoch for English–Spanish and English–German, 2.1 epochs for English–Chinese, 3.6 epochs for English–Turkish; 2.4 epochs for English–Arabic; and 28.7 epochs for English–Swahili.

Each pre-training job is executed on two NVIDIA RTX A6000 GPUs (48 GB), consuming approximately 96 GPU-hours per model (≈48 wall-clock hours). Training the full suite of 24 models therefore required 48 GPUs and about 2304 GPU-hours in total.

E Fine-tuning Experiment Details

For MultiNLI fine-tuning, we train each model for 5 epochs using a per-device batch size of 64 sequences and a maximum sequence length of 1024 tokens. Optimization is performed with AdamW using a cosine learning rate schedule without warmup. We conduct a hyperparameter sweep over three batch sizes (128, 256, 512) and three learning rates ($1e{-}5$, $5e{-}5$, $1e{-}4$), saving checkpoints every 500 gradient steps. Because the number of epochs is fixed, the total number of steps varies with the batch size. This sweep is performed independently for each language pair and overlap setting, and we select the best model and checkpoint based on validation performance on MultiNLI. Each run is trained on a single NVIDIA RTX A6000 GPU (48GB) and takes approximately 1.5 GPU hours on average. Across 24 models and 9 hyperparameter configurations, the total compute cost is roughly 320 GPU hours.

For SQuAD fine-tuning, we train each model for 7 epochs with a per-device batch size of 16 sequences and a maximum sequence length of 1024 tokens. The optimizer settings and hyperparameter sweep configurations are the same used for MultiNLI, but we save checkpoints every 200 steps. Each run is also trained on a single NVIDIA RTX A6000 GPU (48GB) and takes about 2 GPU hours on average. Across 24 models and 9 hyperparameter configurations, the total compute amounts to roughly 430 GPU hours.

Language Pair	Setting	Vocabulary Size	Total Parameters
English–Spanish	Full Overlap	107,894	167.9M
	High-similarity Overlap	174,271	218.9M
	Low-similarity Overlap	174,271	218.9M
	No Overlap	196,374	235.9M
English–German	Full Overlap	101,813	163.2M
	High-similarity Overlap	163,178	210.4M
	Low-similarity Overlap	163,178	210.4M
	No Overlap	183,772	226.2M
English–Turkish	Full Overlap	76,645	143.9M
	High-similarity Overlap	121,463	178.3M
	Low-similarity Overlap	121,463	178.3M
	No Overlap	135,370	189.0M
English–Chinese	Full Overlap	84,144	149.7M
	High-similarity Overlap	128,648	183.9M
	Low-similarity Overlap	128,648	183.9M
	No Overlap	141,247	193.5M
English–Arabic	Full Overlap	81,020	147.3M
	High-similarity Overlap	128,142	183.5M
	Low-similarity Overlap	128,142	183.5M
	No Overlap	138,106	191.1M
English–Swahili	Full Overlap	50,381	123.7M
	High-similarity Overlap	82,923	148.7M
	Low-similarity Overlap	82,923	148.7M
	No Overlap	87,657	152.4M

Table 4: Vocabulary sizes and parameter counts for each overlap setting. Parameter counts are shown in millions (M).

F Embedding Similarity Analysis Over Training

In Figures 5, 6, and 7 we analyze embedding similarity at training checkpoints from 20k to 100k steps, in 20k increments. Across language pairs, we observe several trends. In the *Full Overlap* setting, the scores for high-similarity tokens gradually separate from low-similarity ones over the course of training. *High-similarity Overlap* shows a strong separation throughout training, with low-similarity tokens becoming more similar over time. In *Low-similarity Overlap*, low-similarity tokens initially have higher similarity scores, but this reverses during training. *No Overlap* shows little change in similarity scores over time.

G Significance Tests

In this section, we present the Cohen's d effect sizes for our embedding similarity analysis (Table 5), as well as the p-values for the pairwise McNemar tests between performance metrics on the XNLI and XQuAD downstream tasks (Table 6).

Language Pair	Full Overlap	High-Sim. Overlap	Low-Sim. Overlap	No Overlap
English–Spanish	2.134	4.156	0.044	1.028
English–German	2.458	5.053	0.049	0.721
English–Turkish	1.766	3.512	-1.151	0.559
English–Chinese	1.358	2.642	-1.350	0.467
English–Arabic	1.264	1.918	-0.992	0.646
English–Swahili	1.706	2.569	-1.639	0.661

Table 5: Cohen's d effect sizes from our embedding similarity analysis. These values compare the cosine similarities between the High-similarity and Low-similarity token sets for each language pair and vocabulary overlap condition.

English–Spanish (L_1)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	<.001 / .381	<.001 / 1.000	<.001 / .788
High-sim. Overlap	—	.206 / .436	.001 / .232
Low-sim. Overlap	—	—	<.001 / .745
English–German (L_1)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.076 / .454	.130 / .734	.253 / .675
High-sim. Overlap	—	.812 / .708	.551 / .211
Low-sim. Overlap	—	—	.752 / .385
English–Turkish (L_1)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.039 / .571	.097 / .365	.810 / .307
High-sim. Overlap	—	.781 / .725	.023 / .631
Low-sim. Overlap	—	—	.062 / .945
English–Chinese (L_1)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.011 / 1.000	.012 / .640	.005 / .340
High-sim. Overlap	—	1.000 / .688	.844 / .393
Low-sim. Overlap	—	—	.879 / .687
English–Arabic (L_1)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.592 / .337	.730 / .890	.574 / 1.000
High-sim. Overlap	—	.871 / .456	1.000 / .401
Low-sim. Overlap	—	—	.850 / .947
English–Swahili (L_1)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.313 / —	.853 / —	.291 / —
High-sim. Overlap	—	.235 / —	.038 / —
Low-sim. Overlap	—	—	.413 / —

(a) L_1 (English) results.

English–Spanish (L_2)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	<.001 / <.001	<.001 / .841	<.001 / <.001
High-sim. Overlap	—	.322 / .002	<.001 / <.001
Low-sim. Overlap	—	—	<.001 / <.001
English–German (L_2)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.366 / .305	.837 / .002	<.001 / <.001
High-sim. Overlap	—	.277 / <.001	<.001 / <.001
Low-sim. Overlap	—	—	<.001 / <.001
English–Turkish (L_2)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	<.001 / .423	<.001 / .867	<.001 / <.001
High-sim. Overlap	—	<.001 / .319	<.001 / <.001
Low-sim. Overlap	—	—	<.001 / <.001
English–Chinese (L_2)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	<.001 / 1.000	<.001 / <.001	<.001 / <.001
High-sim. Overlap	—	<.001 / <.001	<.001 / <.001
Low-sim. Overlap	—	—	<.001 / <.001
English–Arabic (L_2)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.818 / .425	<.001 / <.001	<.001 / <.001
High-sim. Overlap	—	<.001 / <.001	<.001 / <.001
Low-sim. Overlap	—	—	<.001 / .008
English–Swahili (L_2)			
Overlap Setting	High-sim. Overlap	Low-sim. Overlap	No Overlap
Full Overlap	.208 / —	<.001 / —	<.001 / —
High-sim. Overlap	—	<.001 / —	<.001 / —
Low-sim. Overlap	—	—	<.001 / —

(b) L_2 transfer results.

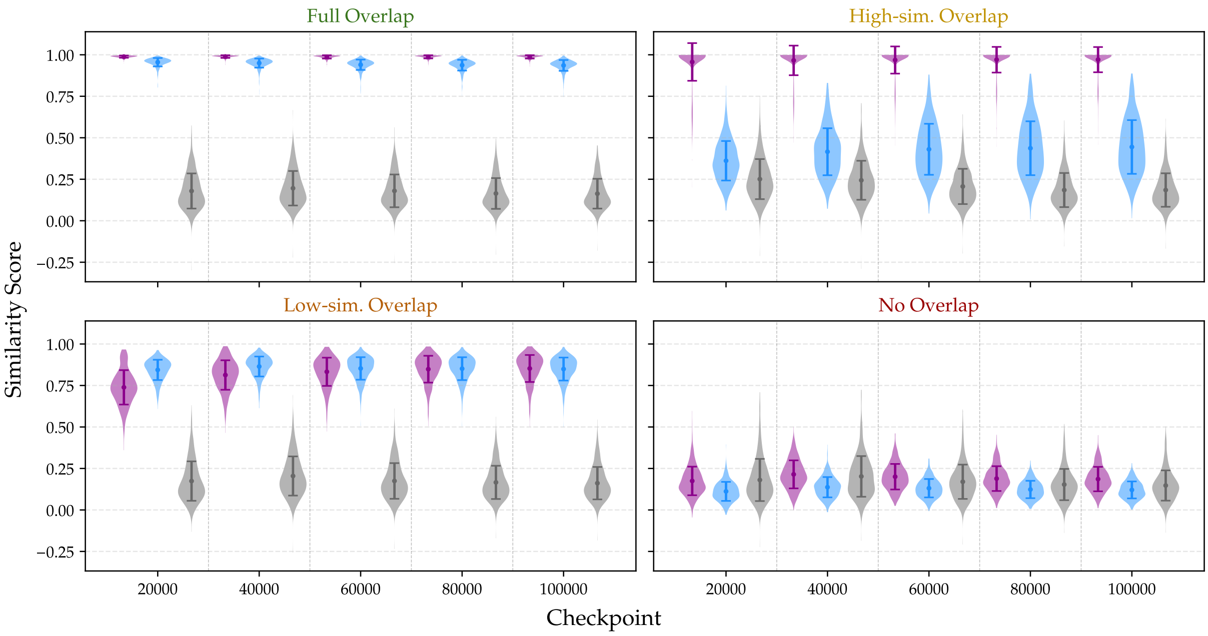
Table 6: McNemar p-values for XNLI / XQuAD across all overlap settings and language pairs. (a) presents results on L_1 (English); (b) presents L_2 transfer results. In each table entry, the first number is XNLI; the second is XQuAD.

H Licenses

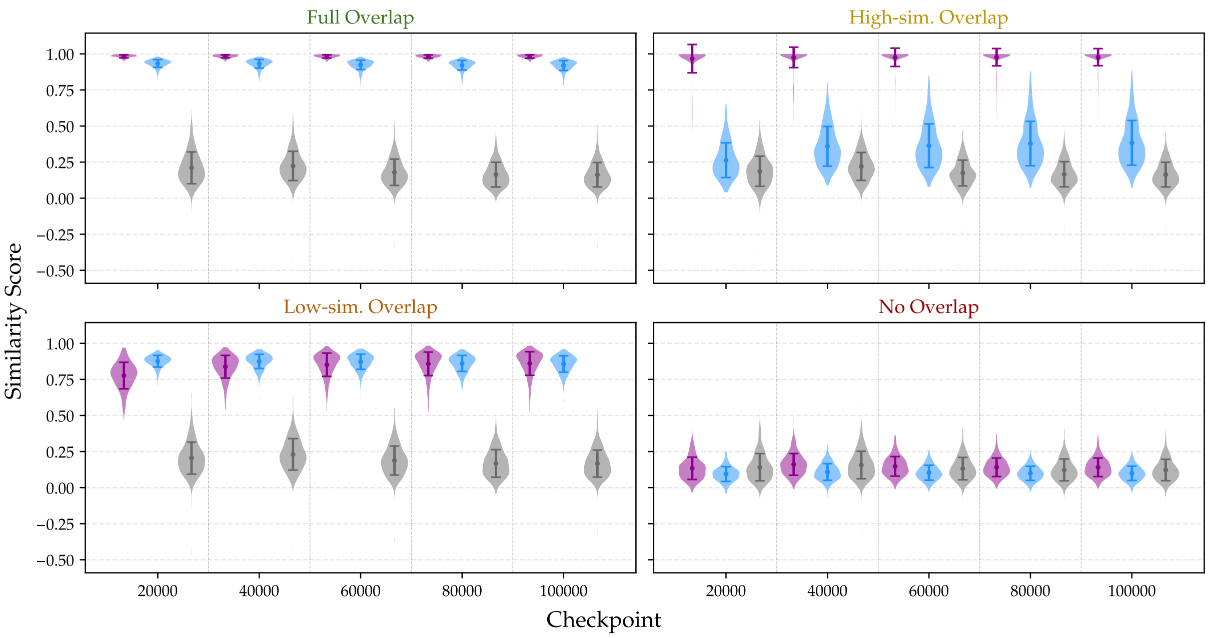
The CCMatrix corpus was released under the BSD license, and XLM-R was released under the MIT license. We will release our code and models under the MIT license. Our use of these artifacts is consistent with their intended use.

I Software Packages

We use the following software libraries in our experiments: HuggingFace Transformers v4.47.0, Datasets v3.2.0, PyTorch v2.5.1, SentencePiece v0.2.0, and Statsmodels v0.14.4.

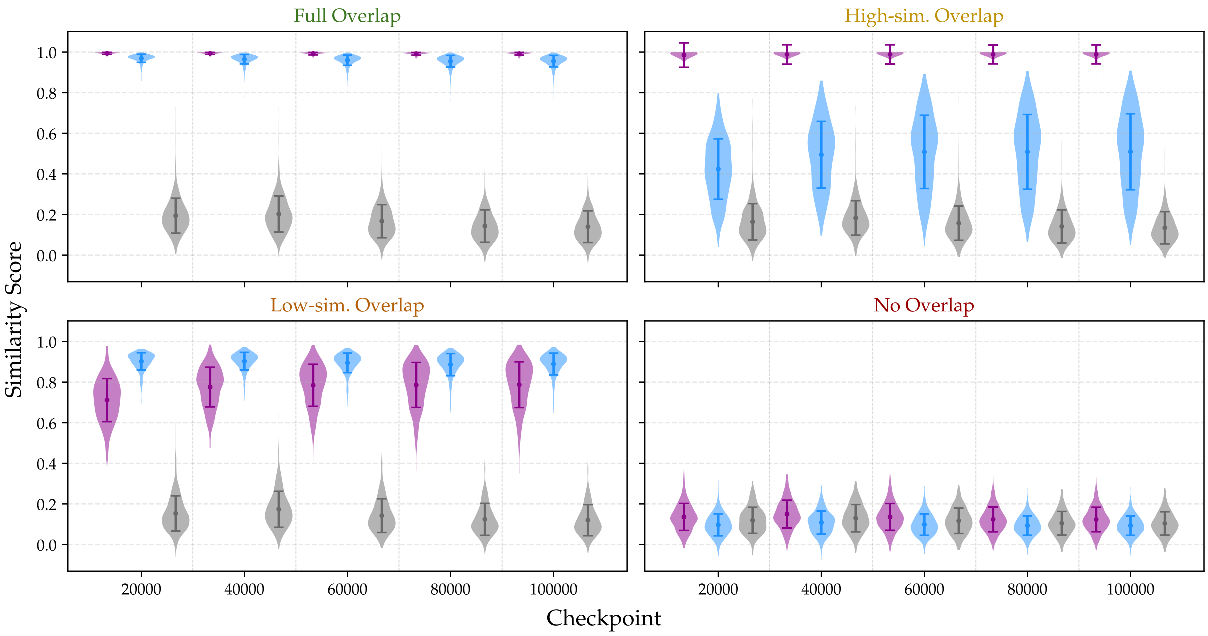


(a) English–Spanish.

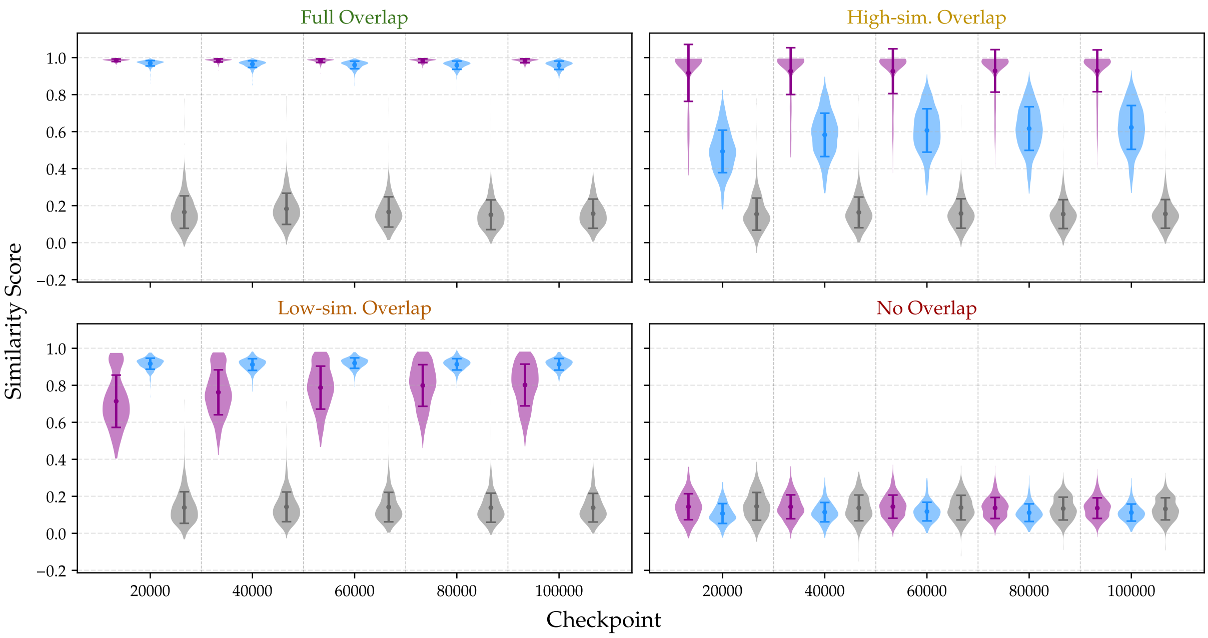


(b) English–German.

Figure 5: Embedding similarity analysis for English–Spanish and English–German over pre-trained model checkpoints.

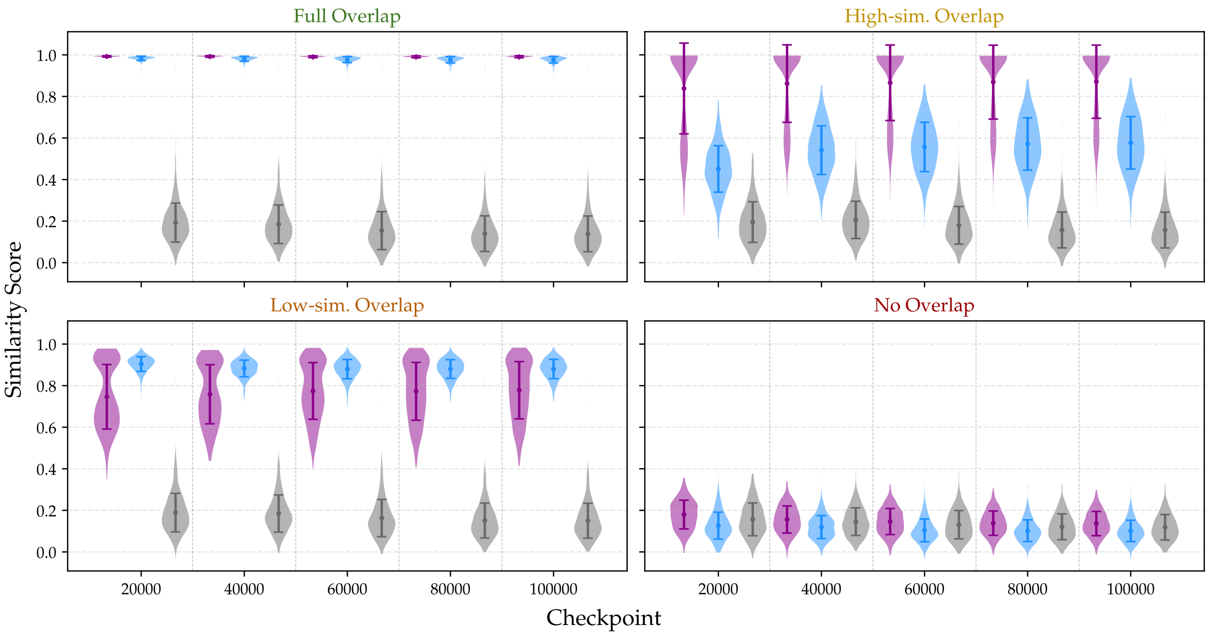


(a) English–Turkish.

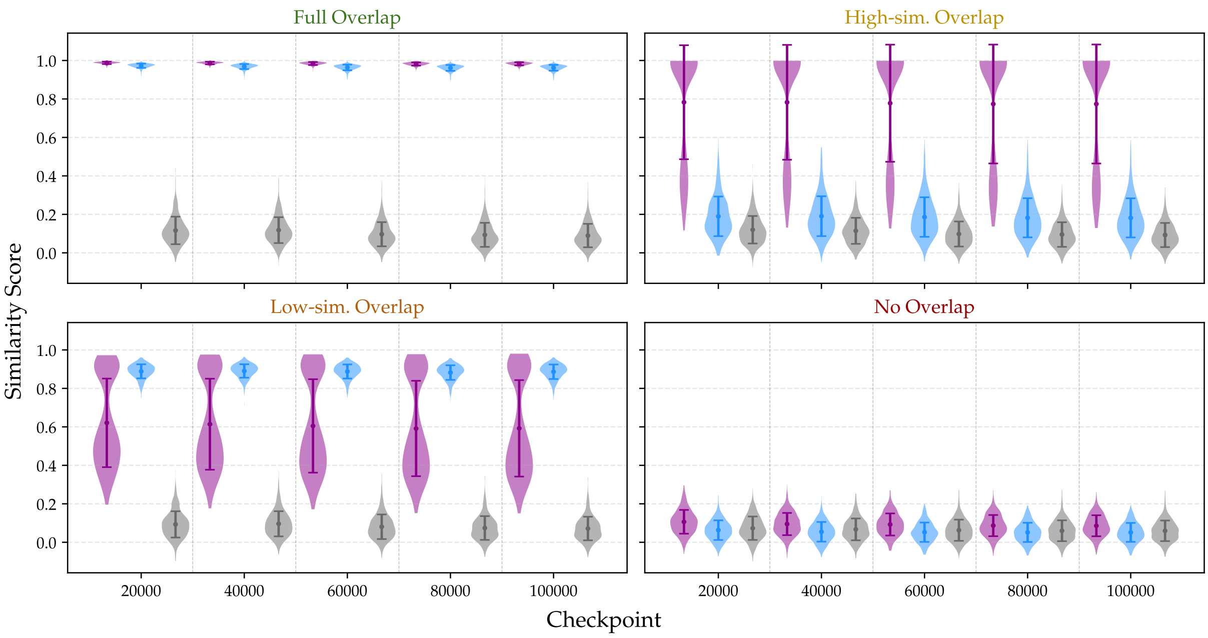


(b) English–Chinese.

Figure 6: Embedding similarity analysis for English–Turkish and English–Chinese over pre-trained model checkpoints.



(a) English–Arabic.



(b) English–Swahili.

Figure 7: Embedding similarity analysis for English–Arabic and English–Swahili over pre-trained model checkpoints.