What If Chinese Were Latinized? A Counterfactual Study of Script, Tokenization, and Language Modeling

Anonymous ACL submission

Abstract

If Chinese had been Latinized, as the Latinxua Sin Wenz movement proposed, what would it have meant for Chinese NLP? Homophones and tone marking complicate the answer. We convert Chinese corpora to syllable-separated Pinyin and train superBPE tokenizers from scratch. The resulting Pinyin vocabularies not only exhibit expected homophone collisions, but also show $\tilde{1}0\%$ higher higher fertility than characters. Under matched-token compute, we further pretrain a Llama-style model on Chinese characters and on Pinyin; the Pinyin model is worse on aggregate across per-character perplexity, homophone disambiguation, and overall ZhoBLiMP minimal-pair accuracy. The broader implication is that writing systems are not merely neutral encodings but also part of the linguistic representation for language models, impacting their downstream performances. Latinization never actually happened for Chinese, but other languages did, such as Hangul and latin script replaced character in Korean and Vietnamese; the cost we measure is a guide to where modelling effort for those languages should concentrate.

1 Introduction

In the 1920s–50s, the **Latinxua Sin Wenz** (拉丁化新文字, "Latinized New Script") movement proposed abolishing Chinese characters in favor of a Latin alphabet. It was championed by figures from Qu Qiubai to Lu Xun and over 600 writers and scholars (DeFrancis, 1950).[1] Although the movement ultimately failed, the question remains: *how would Chinese NLP be different if Chinese had been Latinized?*

[1]Latinxua differs from earlier Latinizations (Wade–Giles, Gwoyeu Romatzyh) as a *complete* writing system rather than a pronunciation aid, with no tone marks and explicit word spacing; the Dungan language (Dyer, 1967), a Cyrillic-script Chinese variety in Central Asia, is a real fully alphabetic Chinese variety. See DeFrancis (1984); Norman (1988); Seybolt and Chiang (1979); Zhou (1965); Cheng (1986) for broader histories.

Figure 1: Overview of the counterfactual. **Top:** example homophone collisions under Latinization. **Bottom:** one tokenizer metric and two model-evaluation metrics under matched-token compute.

We frame this as a script-level counterfactual: we hold the underlying Chinese content fixed and replace the character script with a written Pinyin representation. Two issues make the answer non-obvious. First, Chinese is highly **homophonous** (hundreds of distinct morphemes share the pronunciation *shi*, for instance), and characters distinguish these orthographically; therefore Latinization risks introducing lexical ambiguity. Second, Latinization introduces **phonetic and tonal information** that former character-based input does not explicitly provide (*shī, shí, shǐ, shì* differ by one mark) by replacing the character forms.

Our experiment is illustrated in Figure 1. We construct our corpus based on today's standard Latin transcription of Mandarin under three orthographic conventions (toneless, number-toned, and diacritic) and train matched superBPE (Liu et al., 2025a) tokenizers from scratch on each. Under both matched-token budget and matched-source control, we pretrain Llama-style language models on Chinese-Origin and Pinyin-Diacritic, and evaluate the models on held-out perplexity, a

homophone-interference probe, and ZhoBLiMP (Liu et al., 2025b), a Chinese minimal-pair acceptability benchmark.

What we find. At the tokenizer level, Pinyin-Diacritic exhibits widespread homophone collisions (up to 41:1) and roughly 10% higher fertility. At the model level, the Latinized model is worse on aggregate across per-character perplexity, homophone disambiguation, and overall ZhoBLiMP accuracy. These results suggest that Chinese characters are not a redundant visual layer but preserve lexical distinctions. More broadly, scripts shape what linguistic information is available to models, motivating script-aware multilingual NLP and future tests of character-derived information in languages such as Korean.

2 Related Work

Chinese NLP defaults on character, but several research explores how to incorporate additional lexical, phonological, or visual information: glyph-based models use visual features of characters (Meng et al., 2020), while ChineseBERT augments character representations with both glyph and pinyin embeddings (Sun et al., 2021). These approaches treat pinyin as an auxiliary signal that supplements the character channel. We instead replace characters with Pinyin entirely, asking what is lost when the orthographic information is removed. Methodologically, we follow controlled pretraining studies such as Kallini et al. (2024), which use synthetic transformations of English to probe what structures language models can learn. To our knowledge, no prior NLP work uses Latinized Chinese as a counterfactual experimental framework.

3 Tokenizer Analysis

3.1 Setup

Corpus and variants. From cleaned Chinese Wikipedia text, we construct three Pinyin orthographic variants that differ only in how tone is represented: **Pinyin-Toneless** (tones removed, e.g., *zhong guo ren min*), **Pinyin-Toned** (tones as numbers, *zhong1 guo2 ren2 min2*), and **Pinyin-Diacritic** (tones as diacritics, *zhōng guó rén mín*); the character baseline (Chinese-Origin) is unchanged. Full preprocessing, segmentation, and splits are in Appendix A.

Mapping	Chinese	Pinyin-Diacritic
Shared 1:1	32,025	32,025
Shared 2:1	3,722	1,861
Shared 3:1	1,020	340
Shared 4:1	580	145
Shared $>$ 4:1	3,027	362

Table 1: Cross-script token overlap between the 64K tokenizers. Rows report mapping multiplicity. For example, Shared 2:1 denotes two distinct Chinese tokens mapping to one Pinyin-Diacritic token.

Tokenizer matrix. We train four vocabulary sizes per variant (8K, 16K, 32K, 64K), yielding 16 superBPE tokenizers. Tokenizer training uses a fixed 100K-line subset of the training pool; fertility evaluation uses the full held-out split.

SuperBPE training. We train superBPE (Liu et al., 2025a) with two-stage pre-tokenization (transition at K=2048); two-stage rationale, regexes, and robustness to alternative K are reported in Appendix B.

Result. The vocabulary composition of the four tokenizers is shown in Appendix C.

3.2 Cross-Script Token Overlap

Do the same lexical units appear as tokens in both the character-based and Latinized vocabularies? Aligning the 64K Chinese-Origin and Pinyin-Diacritic tokenizers, we map every Chinese token to its Pinyin-Diacritic form and group by the resulting mapping multiplicity (Table 1): a 1:1 mapping is a bijective match, while N:1 mappings ($N \geq 2$) constitute homophone collisions.

Homophone collisions. Table 1 shows that homophone collisions are widespread. We use *collapse* to describe such N:1 mappings: for example, the Pinyin-Diacritic form *quán lì* collapses 全力 ("with full effort"), 权利 ("rights"), and 权力 ("power") into a single surface form. The largest collision has multiplicity 41:1; the full distribution and collision examples are shown in Figure 2. Additional details and overlap results for other Pinyin tokenizers are reported in Appendix D.

Semantic dispersion of collisions. Some collision groups erase grammatical or referential distinctions encoded in orthography, such as *tā de* = 他的/她的/它的 ("his/her/its"). Other groups merge lexical items that are much farther apart semantically, such as *qí shì* = 骑士/歧视 ("knight/discrimination"). Table 2 illustrates that

several groups have dispersion close to a random-pair baseline, indicating substantial semantic mixing within particular homophone classes. The Chinese lexical items and glosses underlying these representative groups are listed in Appendix E. Additional results for other Pinyin tokenizers are reported in Appendix F.

Pinyin-Diacritic token	N	BGE	text2vec
tā de	4	0.3154	0.3219
àn jiàn	2	0.6486	0.5867
diàn shì	2	0.6639	0.7092
yù shì	2	0.6634	0.7626
gōng shì	6	0.4934	0.5556
xíng shì	5	0.4682	0.5103
qí shì	4	0.6436	0.6659
1:1 baseline	1	0	0
All	All	0.4028	0.4345
Random baseline	10K pairs	0.6503	0.6237

Table 2: Semantic dispersion of representative N:1 homophone collision groups. Distances are average pairwise cosine distances among the collapsed Chinese tokens, computed with `BAAI/bge-small-zh-v1.5` and `shibing624/text2vec-base-chinese`. The *1:1 baseline* denotes zero dispersion for a bijective mapping. The *All* row aggregates all collision groups after retaining source tokens with at least two Chinese characters. The *random baseline* is sampled from 10K random pairs of the full Chinese vocabulary.

3.3 Tokenization Fertility

We next evaluate tokenization fertility, measured as tokens per aligned corpus line, on the held-out split corresponding to each tokenizer (Table 3). A perhaps surprising result is that Chinese-Origin has lower fertility than all three Latinized variants: despite homophone collisions, Pinyin-Diacritic requires roughly 10% more tokens per aligned line. This indicates that Latinization does not automatically make Chinese more compact for tokenization, possibly because superBPE can merge continuous Chinese character sequences into stable lexical chunks, whereas Pinyin must allocate merge capacity across syllable spacing, tone-marked variants, and Latin-script boundary fragments.

4 Pretraining and Downstream Evaluation

In §3 we investigated the effect of a Latinized Chinese tokenizer vocabulary. A natural next question

Tokenizer	Tokens per line
Chinese SuperBPE (64K)	312.82
Pinyin-Toned SuperBPE (64K)	343.02
Pinyin-Toneless SuperBPE (64K)	327.49
Pinyin-Diacritic SuperBPE (64K)	343.01
Qwen-7B	414.69

Table 3: Tokens per line on the held-out zh-wiki test set corresponding to each tokenizer. Qwen-7B is evaluated on Chinese-Origin text as a production-tokenizer reference.[2]

is whether these representational differences translate into measurable downstream consequences for language model behavior. We therefore conduct a controlled *light pretraining* study, training small language models from scratch under matched-token and matched-source settings for both writing systems. Our goal is to isolate the effect of script choice on learnability.

4.1 Experimental Setup

Script variants and model. We restrict the comparison to **Chinese-Origin** and **Pinyin-Diacritic**. Pinyin-Diacritic preserves tone information and is closest to the standard tone-marking convention of Pinyin. All pretraining runs use the same 134M-parameter Llama-style decoder-only architecture and matching 32K superBPE tokenizers trained in §3, so that the primary difference is the underlying script representation. Full architecture, corpus, seed, and training-schedule details are in Appendix G.

Matching. We report three settings: **Chinese**, the Chinese-Origin baseline; **Pinyin-token**, the Pinyin-Diacritic model under the same matched-token budget; and **Pinyin-source**, a matched-source Pinyin-Diacritic control. The matched-token comparison fixes the number of model-token predictions, matching the dominant training compute and the effective cost paid during training. Because Pinyin has higher fertility, Pinyin-token sees fewer passes over the underlying source corpus than Chinese. Pinyin-source instead has equal source-corpus exposure, making it a strictly more favorable control than Pinyin-token. All settings use three independent seeds and report mean $\pm$ standard deviation.

4.2 Downstream Evaluation

Figure 3 shows the loss log of the three diagnostic runs. We probe three aspects of script-induced behavior, each motivated by a representational sig-

[2]We use the tokenizer from the Hugging Face `Qwen/Qwen-7B` base model repository.

Probe	Chinese	Pinyin-token	Pinyin-source
Pinyin overlap*	**96.11**$\pm$0.34	88.85$\pm$0.72	88.85$\pm$1.06
Semantic control	**97.33**$\pm$0.35	90.80$\pm$0.78	91.13$\pm$0.76
Random control	**98.63**$\pm$0.23	93.90$\pm$0.46	93.57$\pm$0.31

Table 4: Accuracy on the Homophone-interference lexical probe. *Pinyin overlap** excludes items collapsed by Pinyin-Diacritic conversion. Construction and scoring details are given in Appendix H.

nature observed in §3. When a benchmark contrast collapses under Pinyin-Diacritic conversion and input becomes indistinguishable, we mark the subset after removing collapsed items with an asterisk (*).

Held-out perplexity. We evaluate script-comparable perplexity by summing the negative log-likelihood under each model's own tokenization of the held-out zh-wiki test split and normalizing by the number of source Chinese characters. Per-character perplexity is 9.14 ± 0.01 for Chinese-Origin, compared with 10.53 ± 0.03 for Pinyin-token and 10.36 ± 0.03 for Pinyin-source. Thus, under matched-token compute, Latinization increases perplexity by roughly 15.1%.

Homophone-interference lexical probe. We next ask whether the homophone collisions observed at the tokenizer level make lexical choice harder for Pinyin models. We construct a 1,000-item probe from held-out zh-wiki text where the gold word is paired with a distractor that shares the same Pinyin after tone marks are removed, as **Pinyin Overlap**. Additionally, we include two non-homophone controls in the same sentence contexts: **Semantic Control**, matched by semantic distance to the gold word, and **Random Control**. A structural limitation appears before scoring: 229 of the 1,000 same-Pinyin items collapse. The results in Table 4 show both a hard surface-collapse effect and a smaller residual effect on non-collapsed pronunciation-overlapping pairs. Pronunciation overlap therefore adds a modest disambiguation cost beyond the general lexical-completion gap.

ZhoBLiMP. We evaluate ZhoBLiMP (Liu et al., 2025b), a Chinese BLiMP-style minimal-pair benchmark over 15 linguistic phenomena. A model is correct if it assigns higher likelihood to the acceptable sentence than to its minimally different counterpart. For readability, Table 5 groups related phenomena. Both Pinyin-Diacritic variants are above chance overall but remain below Chinese-Origin. The gaps are not evenly distributed. They are smaller for *order-sensitive* phenomena such as topicalization, relativization, and passive, where Pinyin preserves sentence order. By contrast, larger drops appear when the contrast is carried by character-distinguishable lexical or functional items, such as gendered pronouns and licensing expressions (*Anaphor** and *Functional licensing*).

Phenomenon class	Chinese	Pinyin-token	Pinyin-source
Functional licensing	**80.0**$\pm$3.1	50.7$\pm$3.0	52.0$\pm$5.2
Constructional	**81.6**$\pm$1.2	71.4$\pm$1.8	71.3$\pm$1.5
Anaphor*	**57.3**$\pm$1.8	49.9$\pm$2.0	45.6$\pm$2.3
Argument structure	**68.9**$\pm$1.0	63.9$\pm$0.8	64.3$\pm$1.6
Order-sensitive	**77.4**$\pm$2.2	72.5$\pm$1.0	71.9$\pm$2.3
Unstable categories	45.1$\pm$4.4	**48.4**$\pm$1.0	48.2$\pm$1.8
Overall	**68.4**$\pm$0.7	61.2$\pm$0.8	61.8$\pm$1.3

Table 5: ZhoBLiMP accuracy aggregated by phenomenon class. Class scores are macro-averages over assigned phenomena. *Anaphor** excludes items collapsed by Pinyin-Diacritic conversion, while *Unstable categories* aggregates five categories near or below chance for at least one model. Class definitions and the full per-phenomenon breakdown are given in Appendix I.

5 Discussion and Conclusion

What the script is for. Across our tokenizer and evaluation results, the Latinization counterfactual changes how linguistic information is made accessible to the model. The cost from Latinization has two components: *representational collapse*, where Latinization removes the contrast from the input entirely (他/她 both map to *tā*), and a residual *likelihood-discrimination cost* on non-collapsed items that remain surface-distinguishable but are harder to score reliably.

Beyond the Chinese case. Chinese was never Latinized in Pinyin as its primary writing system, but comparable script choices are real elsewhere. Vietnamese is a tonal language written with Latin letters and tone diacritics, while Korean Hangul encodes sound in alphabetic units arranged into syllable blocks. These cases show that writing systems are not neutral encodings for language models, but part of the linguistic representation that shapes tokenization and recoverability. Our experiment may motivate future matched-script studies, such as testing Hanja-derived information in Korean. For multilingual LLMs, cross-lingual fairness should also account for how scripts make linguistic information more or less accessible to models.

Limitations

Our study has several limitations.

First, our Pinyin conversion pipeline relies on automated segmentation and character-to-Pinyin mapping with `jieba` and `pypinyin`. Segmentation errors and imperfect polyphone disambiguation may propagate into tokenizer training and evaluation inputs.

Second, our counterfactual depends on orthographic choices such as syllable spacing and tone marking. Because Pinyin-Diacritic uses explicit syllable spacing while Chinese-Origin remains ordinary unspaced character text, our results combine script form with a particular segmentation convention. Although we compare three Pinyin variants at the tokenizer level, future work should include controls such as spaced Chinese or character-separated Chinese to better isolate script form from spacing effects.

Third, our pretraining experiments use 134M-parameter decoder-only Transformers trained on Chinese Wikipedia. We report three seeds and both matched-token and matched-source controls, but the absolute gaps may change with larger models, longer training, or broader corpora. Larger models may reduce non-collapsed likelihood-discrimination costs, although exact representational collapse remains unrecoverable from the input alone.

Fourth, Chinese Wikipedia is encyclopedic and may not represent informal, conversational, or code-mixed Chinese. Testing other registers is left to future work.

Ethics Statement

This work uses publicly available Chinese Wikipedia text and fully automated Pinyin conversion; no human subjects or sensitive data are involved. Chinese Wikipedia text is used under the Wikimedia content license, including Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0) and the GNU Free Documentation License (GFDL). We do not infer protected attributes or make claims about individual users. We used AI writing assistance for language polishing, wording suggestions, and revision support; all research ideas, experimental design, analyses, and final text were reviewed, edited, and approved by the authors, who take full responsibility for the content of the paper. We study a counterfactual writing system as a controlled probe for tokenization and language modeling, not as a proposal for orthographic reform. We do not foresee specific risks of misuse beyond those associated with any small-scale Chinese language model. All preprocessing code, tokenizers, pretraining configurations, and trained checkpoints will be released upon publication to support replication.

References

Chin-Chuan Cheng. 1986. Contradictions in chinese language reform. *International Journal of the Sociology of Language*, 1986(59):87–96.

J. DeFrancis. 1950. *Nationalism and Language Reform in China*. Princeton University Press.

John DeFrancis. 1984. *The Chinese Language: Fact and Fantasy*. University of Hawaii Press, Honolulu.

S.R.K. Dyer. 1967. *Soviet Dungan: The Chinese Language of Central Asia : Alphabet, Phonology, Morphology*. Reprint series / Asian Studies Research Institute, Indiana University. Asian Studies Research Institute, Indiana University.

Mozhi Huang. 2014. pypinyin: A python library for converting chinese characters to pinyin.

Julie Kallini, Isabel Papadimitriou, Richard Futrell, Kyle Mahowald, and Christopher Potts. 2024. Mission: Impossible language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14691–14714, Bangkok, Thailand. Association for Computational Linguistics.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025a. Superbpe: Space travel for language models. *Preprint*, arXiv:2503.13423.

Yikang Liu, Yeting Shen, Hongao Zhu, Lilong Xu, Zhiheng Qian, Siyuan Song, Kejia Zhang, Jialong Tang, Pei Zhang, Baosong Yang, Rui Wang, and Hai Hu. 2025b. A systematic assessment of language models with linguistic minimal pairs in chinese. *Preprint*, arXiv:2411.06096.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2020. Glyce: Glyph-vectors for chinese character representations. *Preprint*, arXiv:1901.10125.

J. Norman. 1988. *Chinese*. ACLS Humanities E-Book. Cambridge University Press.

P.J. Seybolt and G.K.K. Chiang. 1979. *Language Reform in China: Documents and Commentary*. M. E. Sharpe.

Junyi Sun. 2012. jieba: Chinese text segmentation.

Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, Xiang Ao, Qing He, Fei Wu, and Jiwei Li. 2021. Chinese-BERT: Chinese pretraining enhanced by glyph and Pinyin information. 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 2065–2075, Online. Association for Computational Linguistics.

E. Zhou. 1965. *Reform of the Chinese Written Language*. Foreign Languages Press.

A Conversion Pipeline Details

Source corpus and cleaning. We use Chinese Wikipedia as the source corpus. After extracting article text with WikiExtractor, we clean the corpus by converting traditional characters to simplified Chinese, removing citation markers and wiki-style internal links, truncating reference and external-link sections, deleting selected markup symbols, and normalizing whitespace. We map common Chinese punctuation to Western equivalents while preserving numbers, URLs, existing Latin text, and other non-Chinese content.

Pinyin conversion. For Pinyin conversion, we segment the cleaned Chinese text with `jieba` (Sun, 2012). The segmentation provides word-level input to `pypinyin` (Huang, 2014), which improves polyphone disambiguation. The Chinese baseline (Chinese-Origin) is not written with inserted word-boundary spaces and remains ordinary character-based Chinese text. For the three Pinyin variants, each Chinese character is converted to its Pinyin syllable when possible, while non-Chinese content is preserved. Pinyin syllables are separated by spaces; multi-character words are not pre-concatenated, leaving multi-syllable merges to the tokenizer.

Splits. The full corpus contains 1,337,505 aligned lines. We split using shuffled line indices with seed 42, assigning 90% of lines (1,203,754) to the training pool and 10% (133,751) to the evaluation split. Tokenizer training uses a 100,000-line subset sampled from the training pool with a fixed random seed; the same sampled line indices are used for all four representations and all vocabulary sizes. Fertility evaluation uses the full 10% evaluation split, with the Chinese test lines used directly for Chinese-Origin and converted into the corresponding Pinyin form for the Latinized variants.

Table 6 shows an example conversion across the three Pinyin variants.

Variant	Output
Original	中华人民共和国
Toned	zhong1 hua2 ren2 min2 gong4 he2 guo2
Toneless	zhong hua ren min gong he guo
Diacritic	zhōng huá rén mín gòng hé guó

Table 6: Example conversion of 中华人民共和国 ("People's Republic of China") under different orthographic conventions.

B superBPE Tokenizer Training Details

Two-stage pre-tokenization. A key difference between Pinyin text and ordinary English text is that spaces in our representation mark syllable boundaries rather than ordinary word boundaries. Because each Chinese character is converted into a Pinyin syllable and spaces are inserted between syllables, standard whitespace pre-tokenization would prevent BPE from learning cross-syllable units (e.g., merging `zhong` and `guo` into `zhong guo`). We therefore use superBPE (Liu et al., 2025a), which proceeds in two stages controlled by a transition point K. The first stage uses a whitespace-constrained pre-tokenization pattern `\S+|\s+`, which separates non-whitespace spans from whitespace spans; early merges learn characters, syllable-internal structures, and local orthographic units. The second stage switches to a more permissive pattern `[^\p{L}\p{N}\s]+|[\r\n]+`, which mainly isolates punctuation and line breaks and no longer treats ordinary spaces as hard boundaries, allowing cross-syllable sequences such as `zhong guo` and `ren min` to enter the same learnable span. The second stage inherits the first K merges from the first stage and continues until the target vocabulary size is reached; thus K denotes the number of early merges inherited from ordinary BPE, not the final vocabulary size.

Choice of K. We use $K = 2048$ as a fixed pre-superBPE warmup budget across all four corpus variants and all four vocabulary sizes (8K, 16K, 32K, 64K), yielding 16 superBPE tokenizers in total. This choice is motivated by the size of the Pinyin syllable inventory: possible Pinyin syllables number on the order of a few hundred without tones and roughly 1.3K–1.5K when tone distinctions are included, with additional redundancy from boundary variants. $K = 2048$ is therefore large enough to cover most syllable-level units before the superBPE stage while preserving substantial vocabulary capacity for cross-space merges.

Robustness to K. To check whether the choice of $K = 2048$ affects our conclusions, we compare it against $K = 1000$, $K = 3200$, and $K = 6400$ on the 64K Chinese-Origin and Pinyin-Diacritic tokenizers. For each metric, Table 7 reports the largest absolute percentage deviation from the $K = 2048$ baseline across these alternative settings (preserving the sign), and Table 8 reports the baseline values. Most deviations are below 2%, and the largest observed deviation is 6.84% for the number of independent Pinyin-Diacritic tokens, suggesting that the conclusions in the main text are not driven by the specific choice of $K = 2048$.

Metric	Side	Max dev.	K
Vocab (unique)	Chinese	+0.05%	1000
	Diacritic	+1.46%	1000
Vocab (raw)	Chinese	+0.00%	1000
	Diacritic	+0.01%	6400
Overlap shared	Chinese	+1.69%	1000
	Diacritic	+1.94%	1000
Overlap independent	Chinese	-3.14%	1000
	Diacritic	+6.84%	6400
Fertility (tok/char)	Chinese	-0.02%	6400
	Diacritic	+0.25%	6400

Table 7: Maximum percentage deviation from the $K = 2048$ baseline across alternative transition points $K \in \{1000, 3200, 6400\}$ for the 64K Chinese-Origin and Pinyin-Diacritic tokenizers. *Vocab (unique)* and *Vocab (raw)* are total unique whitespace-insensitive token contents and total raw vocabulary entries, respectively. Each row selects the alternative setting with the largest absolute deviation, while the displayed percentage keeps its sign.

Metric	Chinese	Diacritic
Vocabulary total unique	61,143	42,653
Vocabulary total raw entries	63,499	63,754
Overlap total shared	40,374	34,733
Overlap independent	20,768	7,919
Fertility tokens/original char	0.5448	0.5974

Table 8: Baseline values at $K = 2048$ for the metrics used in Table 7.

C Vocabulary Composition Results

Table 9 classifies the vocabulary entries of the Chinese-Origin and Pinyin-Diacritic superBPE tokenizers across vocabulary sizes; Table 10 reports the analogous results for Pinyin-Toneless and Pinyin-Toned superBPE.

D Additional Cross-Script Token Overlap Results

Using the same overlap protocol as in Section 3.2, we additionally report the 64K cross-script overlap results for Pinyin-Toneless and Pinyin-Toned superBPE. Overlap counts exclude whitespace-only tokens whose normalized content is empty; vocabulary composition tables count these as one whitespace-insensitive *Other/Unknown* item, explaining the one-token difference in totals. Each Chinese token is converted into the corresponding Latinized form and matched against the target Latinized vocabulary. Rows are grouped by mapping multiplicity: N:1 denotes N distinct Chinese tokens collapsing to one Latinized token.

Tables 11 and 12 show that cross-script overlap varies substantially across Latinization schemes. Pinyin-Toneless produces more N:1 mappings than both Pinyin-Toned and Pinyin-Diacritic, as removing tone information causes more distinct Chinese tokens to collapse into the same Latinized surface form. The maximum observed collision order is 77:1 for Pinyin-Toneless, compared with 41:1 for both Pinyin-Toned and Pinyin-Diacritic.

Figure 2 plots the full Pinyin-Diacritic collision distribution.

E Homophone Collision Examples

Table 13 lists representative collision groups from the 64K Pinyin-Diacritic vs. Chinese-Origin alignment. Each Pinyin-Diacritic token below corresponds to multiple semantically distinct Chinese tokens that collapse into a single Latinized form even with diacritic tones preserved.

F Additional Semantic Dispersion Results

We report per-group semantic dispersion for representative N:1 collision groups against the 64K Chinese-Origin superBPE, for Pinyin-Diacritic (Table 2), Pinyin-Toneless (Table 14), and Pinyin-Toned (Table 15). Distances are average pairwise cosine distances among the collapsed Chinese tokens, computed using two Chinese embedding models. The ALL row aggregates all collision groups after retaining source tokens with at least two Chinese characters. The random baselines are sampled from 10K random pairs of the full Chinese vocabulary.

	Chinese-Origin superBPE				Pinyin-Diacritic superBPE			
Token Category	**8K**	**16K**	**32K**	**64K**	**8K**	**16K**	**32K**	**64K**
Base inventory								
Single Latin letters (A–Z, a–z)	52 (94)	52 (104)	52 (117)	52 (132)	52 (163)	52 (232)	52 (301)	52 (342)
Single CJK Characters	2378 (2418)	3061 (3181)	3769 (4021)	4515 (4937)	0 (0)	0 (0)	0 (0)	0 (0)
Digits	155 (185)	247 (295)	312 (394)	558 (696)	294 (420)	294 (587)	314 (826)	397 (1215)
Punctuation	131 (131)	201 (202)	308 (309)	471 (472)	58 (59)	62 (65)	70 (73)	94 (97)
Pure Latin Combinations	274 (289)	713 (785)	1541 (1753)	3194 (3780)	1644 (3256)	1692 (4078)	2174 (5480)	3341 (8003)
Multi-syllable / multi-character tokens								
2-Syllable / 2-Char	3443 (3483)	7589 (7722)	15000 (15320)	27294 (28053)	2546 (3327)	6044 (8418)	11851 (17460)	21468 (33200)
3-Syllable / 3-Char	556 (556)	1946 (1949)	5345 (5367)	13200 (13293)	259 (296)	1226 (1404)	3637 (4361)	9390 (11658)
4-Syllable / 4-Char	95 (97)	442 (446)	1684 (1698)	5696 (5739)	50 (54)	256 (287)	974 (1107)	3393 (3991)
5-Syllable / 5-Char	23 (23)	100 (100)	389 (389)	1340 (1343)	23 (23)	61 (64)	247 (262)	867 (963)
6+-Syllable / 6+-Char	40 (42)	121 (125)	386 (395)	1335 (1348)	32 (34)	102 (106)	302 (319)	1026 (1098)
Mixed tokens								
Mixed (Latin + CJK)	5 (5)	12 (12)	49 (49)	157 (161)	0 (0)	0 (0)	0 (0)	0 (0)
Other								
Other / Unknown	242 (265)	593 (652)	1595 (1734)	3331 (3545)	206 (241)	469 (623)	1346 (1632)	2627 (3187)
Total	**7394 (7588)**	**15077 (15573)**	**30430 (31546)**	**61143 (63499)**	**5164 (7873)**	**10258 (15864)**	**20967 (31821)**	**42653 (63754)**

Table 9: Vocabulary composition of Chinese-Origin superBPE and Pinyin-Diacritic superBPE tokenizers across four vocabulary sizes. Each cell reports unique token contents after whitespace removal, with raw vocabulary entries shown in parentheses.

G Pretraining Architecture and Schedule

Model architecture. All pretraining runs use a Llama-style causal language model with 12 Transformer layers, hidden size 768, 12 attention heads, intermediate size 2048, maximum context length 1024, SiLU activations, RMSNorm, and rotary positional embeddings. The vocabulary size is 32,001 for both tokenizers, including a shared EOS/PAD token with id 32,000, and we do not tie input and output embeddings. This yields 134,107,392 trainable parameters per model. All reported pretraining runs were conducted on a NVIDIA A100 GPU and required approximately 52.04 A100 GPU-hours. Apart from the script-specific tokenizer and input representation, the architecture is held fixed across runs.

Pretraining corpus. Tokenizers are trained on the fixed tokenizer-training subset described in §3. LM pretraining uses a separate split from the 90/10 tokenizer-analysis split described in Appendix A. The same processed zh-wiki corpus is repartitioned into 1,310,755 training lines and 13,375 lines each for validation and test, with each line presented in its corresponding script representation.

For Chinese-Origin, the tokenized training split contains 434,846 1024-token rows before drop-last batching. With per-device batch size 16 and gradient accumulation 4, one effective training pass uses 6,794 optimizer updates, corresponding to 445,251,584 effective training tokens. For Pinyin-Diacritic, the same source content yields 476,236 1024-token rows before drop-last batching. One effective Pinyin-Diacritic training pass uses 7,441 optimizer updates, corresponding to 487,653,376 effective training tokens. This difference reflects the higher fertility of the Pinyin-Diacritic representation.

Matched-token training schedule. The primary comparison uses a matched token budget. Each matched-token run is trained for 27,176 optimizer updates with block size 1024, per-device batch size 16, and gradient accumulation 4. This corresponds to 65,536 tokens per update and 1,781,006,336 total training tokens per model. Under this fixed-token setting, Chinese-Origin receives approximately 4.0 full effective passes over its training split, while Pinyin-Diacritic receives approximately 3.65 effective passes over the same underlying source content. This setting holds token-level compute fixed and is therefore our primary compute-matched comparison.

Matched-source control. Because Pinyin-Diacritic has higher fertility, the matched-token setting exposes it to slightly fewer source-data passes. We therefore train Pinyin-Diacritic matched-source controls. Each matched-source run is trained for 29,764 optimizer updates, corresponding to four full effective Pinyin-Diacritic passes and 1,950,613,504 total training tokens. We report these models separately as **Pinyin-source**.

Seeds and aggregation. We train three independent seeds, 42, 43, and 44, for each reported model condition: **Chinese**, **Pinyin-token**, and **Pinyin-**

	Pinyin-Toneless superBPE				Pinyin-Toned superBPE			
Token Category	**8K**	**16K**	**32K**	**64K**	**8K**	**16K**	**32K**	**64K**
Base inventory								
Single Latin letters (A–Z, a–z)	52 (168)	52 (237)	52 (309)	52 (343)	52 (161)	52 (234)	52 (299)	52 (342)
Single CJK Characters	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Digits	423 (556)	423 (735)	425 (985)	486 (1392)	296 (422)	296 (587)	315 (825)	398 (1217)
Punctuation	67 (68)	67 (70)	74 (77)	95 (98)	58 (59)	62 (65)	70 (73)	94 (97)
Pure Latin Combinations	1447 (2410)	1453 (2947)	1626 (3817)	2757 (5965)	1652 (3268)	1699 (4085)	2179 (5488)	3357 (8005)
Multi-syllable tokens								
2-Syllable	3325 (3901)	7099 (8995)	12817 (17313)	20900 (30400)	2545 (3326)	6043 (8417)	11851 (17462)	21466 (33203)
3-Syllable	364 (388)	1537 (1680)	4880 (5490)	12888 (14811)	258 (295)	1224 (1403)	3639 (4362)	9392 (11660)
4-Syllable	58 (63)	324 (353)	1263 (1392)	4312 (4807)	50 (54)	256 (287)	973 (1106)	3393 (3990)
5-Syllable	29 (30)	77 (79)	333 (344)	1068 (1145)	23 (23)	61 (64)	248 (263)	867 (963)
6+-Syllable	36 (39)	108 (113)	387 (400)	1251 (1309)	32 (34)	102 (106)	303 (320)	1026 (1098)
Mixed tokens								
Mixed (Latin + CJK)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)
Other								
Other / Unknown	204 (245)	479 (647)	1359 (1682)	2844 (3471)	198 (233)	461 (615)	1338 (1624)	2619 (3180)
Total	**6001 (7868)**	**11608 (15856)**	**23197 (31809)**	**46602 (63741)**	**5164 (7875)**	**10256 (15863)**	**20968 (31822)**	**42663 (63755)**

Table 10: Vocabulary composition of Pinyin-Toneless superBPE and Pinyin-Toned superBPE tokenizers across four vocabulary sizes. Each cell reports the number of unique whitespace-insensitive token contents, with the raw number of vocabulary entries shown in parentheses.

Mapping	**Chinese**	**Pinyin-Toneless**
Shared 1:1	28,045	28,045
Shared 2:1	6,572	3,286
Shared 3:1	2,937	979
Shared 4:1	1,484	371
Shared $>$ 4:1	6,369	596

Table 11: Cross-script token overlap between the 64K Chinese-Origin superBPE and Pinyin-Toneless superBPE vocabularies.

Mapping	**Chinese**	**Pinyin-Toned**
Shared 1:1	32,006	32,006
Shared 2:1	3,720	1,860
Shared 3:1	1,017	339
Shared 4:1	584	146
Shared $>$ 4:1	3,025	362

Table 12: Cross-script token overlap between the 64K Chinese-Origin superBPE and Pinyin-Toned superBPE vocabularies.

source. All main tables report mean $\pm$ sample standard deviation across these three seeds. Gaps are computed as the difference between the Chinese-Origin mean and the corresponding Pinyin-Diacritic mean.

Training loss curves. Figure 3 plots token-level training and validation losses for the three seed-42 pretraining runs: Chinese-Origin matched-token, Pinyin-Diacritic matched-token, and Pinyin-Diacritic matched-source. Because these curves are token-level losses under different tokenizers, they are used only as optimization diagnostics and are not directly comparable across scripts.

H Homophone-interference Lexical Probe Details

We construct the homophone-interference probe from held-out zh-wiki sentences containing a target word that belongs to a Pinyin-Toneless collision group identified from Appendix D. For each item, the distractor is sampled from the same Pinyin-Toneless group as the target word, so the two candidates share the same toneless pronunciation. This includes both pairs that remain identical under Pinyin-Diacritic conversion, such as same-tone homophones, and pairs that differ once tone marks are restored.

Models are evaluated with *candidate-plus-suffix* scoring. Given the same left context, each model scores the candidate together with the right context, and the prediction is correct if the gold target yields the higher score. This ensures that both candidates are evaluated with the same available sentence context.

We also construct two non-homophone controls using the same contexts and gold targets. One uses a non-homophone distractor matched by semantic distance to the gold word, while another control uses a random non-homophone distractor. Neither control contains Pinyin-Diacritic surface collisions.

I Full ZhoBLiMP Breakdown

Table 16 provides the full per-phenomenon breakdown of ZhoBLiMP results summarized in Table 5.

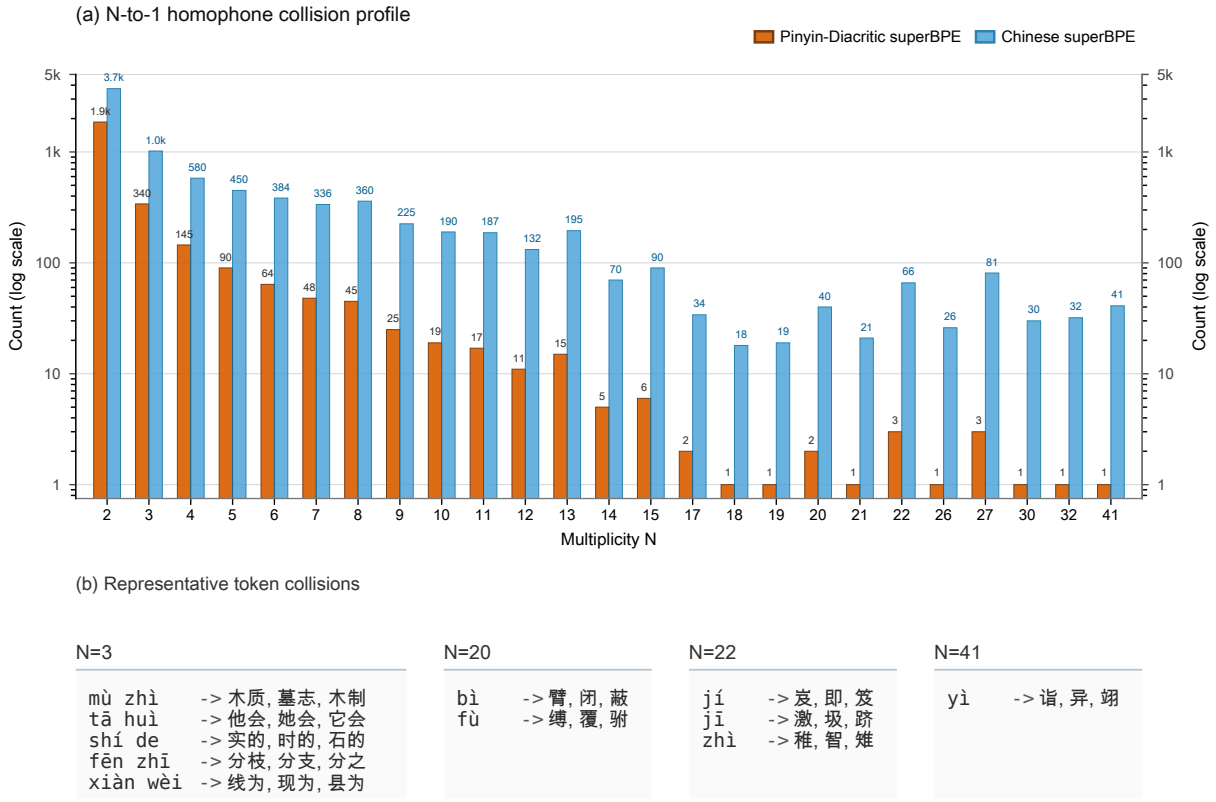

Figure 2: Distribution of homophone collisions between the 64K Chinese-Origin superBPE and Pinyin-Diacritic superBPE vocabularies. Orange bars report the number of Pinyin-Diacritic tokens with multiplicity N; blue bars report the corresponding number of Chinese source tokens.

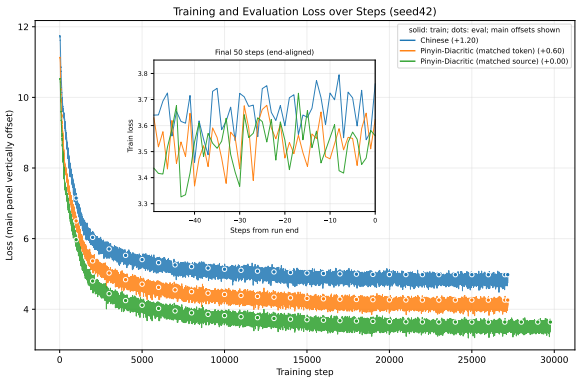

Figure 3: Token-level training and validation loss for the three seed-42 pretraining runs. The main panel uses vertical offsets for readability, and the inset shows the final 50 training steps end-aligned.

Within the unstable cluster, *Question* shows a small aggregate advantage for Pinyin-Diacritic, but subtype analysis indicates that this category is strongly affected by lexical directionality (e.g., directional preferences between *nándào* and *dàodǐ*), so we treat the broad *Question* average as a mixed diagnostic rather than as evidence of a uniform Pinyin-Diacritic advantage.

J Implementation Details and External Packages

Pinyin conversion and preprocessing. We convert Chinese sentences to tone-marked Pinyin using `pypinyin.lazy_pinyin(..., style=Style.TONE, neutral_tone_with_five=False)`. For ZhoBLiMP-style evaluation, we use the Hugging Face dataset implementation `chinese-babylm-org/zhoblimp` with the train split and `all_items=true`, yielding 35,400 minimal pairs.

Semantic-distance models. Semantic-dispersion analyses and semantic-control construction use Hugging Face `sentence-transformers`[3]. Chinese tokens are embedded with `BAAI/bge-small-zh-v1.5`[4] and `shibing624/text2vec-base-chinese`[5]. Embeddings are L2-normalized, and pairwise

[3] https://www.sbert.net/

[4] https://huggingface.co/BAAI/bge-small-zh-v1.5

[5] https://huggingface.co/shibing624/text2vec-base-chinese

Pinyin	Chinese tokens	Glosses
quán lì	全力, 权利, 权力	with full effort, rights, power
tā de	他的, 她的, 它的, 牠的	his, her, its, its (for animals)
àn jiàn	案件, 按键	legal case, button
diàn shì	电视, 殿试	television, palace examination
yù shì	预示, 浴室	to foreshadow, bathroom
qí shì	其事, 骑士, 棋士, 歧视	that matter, knight, chess player, discrimination
gōng shì	公事, 公式, 公示, 公视, 工事, 攻势	official business, formula, public notice, Public TV Service, engineering works, offensive
xíng shì	刑事, 型式, 形势, 形式, 行事	criminal, type, situation, form, conduct affairs

Table 13: Selected homophone collision groups in the 64K Pinyin-Diacritic vocabulary. Each Pinyin token maps to multiple Chinese tokens that Chinese-Origin represents as separate vocabulary entries.

Pinyin-Toneless token	N	BGE	text2vec
`ta de`	5	0.3930	0.4085
`an jian`	3	0.5895	0.6556
`dian shi`	3	0.6108	0.6449
`yu shi`	8	0.6051	0.6227
`gong shi`	11	0.5304	0.5649
`xing shi`	8	0.5039	0.5786
`qi shi`	13	0.5927	0.5931
1:1 baseline	1	0	0
All	All	0.4831	0.4991
Random baseline	10K pairs	0.6505	0.6224

Table 14: Semantic dispersion of representative N:1 collision groups between the 64K Chinese-Origin superBPE and Pinyin-Toneless superBPE vocabularies. Rows follow the same format as Table 2.

Pinyin-Toned token	N	BGE	text2vec
`ta1 de`	4	0.3154	0.3219
`an4 jian4`	2	0.6486	0.5867
`dian4 shi4`	2	0.6639	0.7092
`yu4 shi4`	2	0.6634	0.7626
`gong1 shi4`	6	0.4934	0.5556
`xing2 shi4`	5	0.4682	0.5103
`qi2 shi4`	4	0.6436	0.6659
1:1 baseline	1	0	0
`All`	All	0.4026	0.4343
Random baseline	10K pairs	0.6500	0.6231

Table 15: Semantic dispersion of representative N:1 collision groups between the 64K Chinese-Origin superBPE and Pinyin-Toned superBPE vocabularies. Rows follow the same format as Table 2.

semantic distance is computed as $1 - \cos(\mathbf{x}, \mathbf{y})$. For generated probe/control datasets, we use `shibing624/text2vec-base-chinese` with normalized embeddings and batch size 128. The semantic-dispersion script retains source tokens with at least two CJK characters, samples 10,000 random baseline pairs, and uses random seed 20260430.

Tokenizer implementations. We train our tokenizers with `superBPE` (Liu et al., 2025a). The implementation is based on `PythonNut/superbpe`, which depends on the `alisawuffles/tokenizers-superbpe` fork[6]. In our local fork, we modified the BPE trainer by changing the multi-word-token skip threshold from more than 4 words to more than 32 words. The `alisawuffles/tokenizers-superbpe` commit used is `757f2a5`.

For the public-tokenizer comparison in Table 3, we use `AutoTokenizer.from_pretrained("Qwen/Qwen-7B", trust_remote_code=True)` from Hugging Face[7] and compute token counts with `add_special_tokens=False`.

Software versions. Unless otherwise stated, experiments use `sentence-transformers==5.1.2`, `transformers==4.57.6`, `tokenizers==0.22.2`, `pypinyin==0.55.0`, `datasets==4.5.0`, `torch==2.8.0`, `numpy==2.0.2`, `scikit-learn==1.6.1`, and `tqdm==4.67.3`.

[6] https://github.com/alisawuffles/tokenizers-superbpe

[7] https://huggingface.co/Qwen/Qwen-7B

Phenomenon	**Chinese**	**Pinyin-token**	**Pinyin-source**	Δ_{tok}	Δ_{source}
FCI licensing	88.49±0.77	48.82±6.09	51.93±9.66	+39.67	+36.56
NPI licensing	71.61±6.96	52.55±0.66	52.09±2.62	+19.05	+19.52
Anaphor	55.00±0.42	33.24±1.36	30.41±1.52	+21.76	+24.59
Anaphor*	57.33±1.79	49.86±2.04	45.61±2.29	+7.47	+11.72
Control/raising	84.36±2.87	74.94±1.87	69.97±3.20	+9.42	+14.39
Verb phrase	85.60±0.69	73.49±1.44	75.27±1.53	+12.11	+10.33
BA construction	74.90±1.00	65.90±2.51	68.80±3.53	+9.00	+6.10
Passive	79.24±2.37	72.27±1.10	72.36±0.94	+6.97	+6.88
Argument structure	68.94±1.00	63.89±0.81	64.30±1.64	+5.05	+4.64
Nominal expression	50.37±4.76	45.86±2.05	44.47±3.93	+4.52	+5.90
Topicalization	80.47±2.20	76.50±0.96	74.64±2.46	+3.97	+5.83
Relativization	72.36±5.84	68.70±3.44	68.81±4.84	+3.66	+3.55
Ellipsis	35.48±1.86	36.41±1.38	36.22±1.26	-0.93	-0.74
Question	58.72±1.48	65.32±1.75	67.90±4.33	-6.59	-9.18
Classifier	44.37±4.71	44.89±1.84	47.67±5.17	-0.52	-3.30
Quantifiers	36.72±15.59	49.78±2.71	44.83±3.75	-13.05	-8.11
Overall	68.40±0.66	61.19±0.83	61.76±1.27	+7.21	+6.64

Table 16: Full ZhoBLiMP breakdown. Δ_{tok} and Δ_{source} are percentage-point gaps computed as Chinese mean minus the corresponding Pinyin-Diacritic mean. *Anaphor* reports all anaphor items; *Anaphor** excludes the items that collapse under Pinyin-Diacritic conversion.