Vocab Diet: Reshaping the Vocabulary of LLMs via Vector Arithmetic

Yuval Reif **Guy Kaplan** **Roy Schwartz**
The Hebrew University of Jerusalem
{yuval.reif,guy.kaplan3,roy.schwartz1}@mail.huji.ac.il

Abstract

Large language models (LLMs) often encode word-form variation (e.g., *walk* vs. *walk**ed***) as linear directions in the embedding space. However, standard tokenization algorithms treat such variants as distinct words with different vocabulary entries—quickly filling the size-capped token vocabulary with surface-form variation (e.g., *walk*, *walk**ing***, ***W**alk*), at the expense of diversity and multilingual coverage. We show that many of these variations can be captured by *transformation vectors*—additive offsets that yield the appropriate word representation when applied to a *base form* embedding, in both the input and output spaces. Building on this, we propose a compact reshaping of the vocabulary: instead of assigning unique tokens to each surface form, we compose them from shared *base form* and *transformation* vectors (e.g., *walked* is *walk*+*past tense*). Our approach is lightweight—keeping the pretrained backbone frozen and only training small adaptation modules. We apply it across five languages and multiple LLMs in both pretraining and post-hoc adaptation, freeing 10–40% of vocabulary slots to be reallocated where tokenization is inefficient. Importantly, we do so while also expanding vocabulary coverage to out-of-vocabulary words, and with minimal impact on downstream performance. Our findings motivate a rethinking of vocabulary design, towards a representation that better matches the underlying structure of language and the practical needs of multilingual coverage.[1]

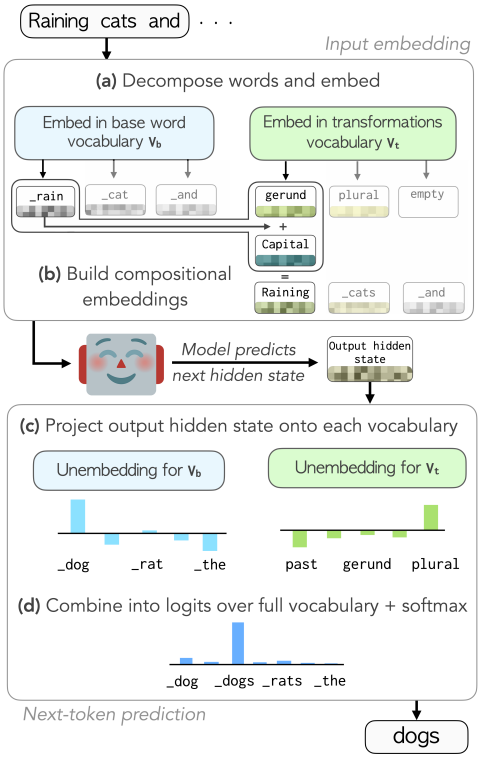

Figure 1: **Compositional vocabulary for LLMs. Top:** Input tokens are represented by **(a)** decomposing them into base words ($\mathcal{V}_b$) and transformations ($\mathcal{V}_t$), and **(b)** feeding the composite embeddings to the model. For example, "*cats*" becomes *cat* + *plural*. **Bottom:** The next token is predicted by **(c)** computing logits independently over base words and transformations, and **(d)** combining them into next-token probabilities. Our approach works seamlessly both as a lightweight adaptation of pretrained LLMs and when pretraining from scratch, creating a more compact vocabulary that supports a wider array of words.

1 Introduction

Modern large language models (LLMs) typically rely on subword tokenization algorithms like byte-pair encoding (BPE; Sennrich et al., 2016). Such methods allocate tokens to frequent words and split less frequent ones into sequences of sub-word tokens—minimizing the number of tokens needed to represent typical textual data. Recent models use ever-larger vocabularies, often exceeding 100k tokens (Grattafiori et al., 2024; OpenAI, 2024; Yang et al., 2024). While recent work calls for scaling up the vocabulary even further (Tao et al., 2024; Huang et al., 2025), the computational cost of supporting large vocabularies forces developers to cap

[1]Code is available at https://vocabdiet.github.io.

Findings of the Association for Computational Linguistics: ACL 2026, pages 32334–32352

its size (Dagan et al., 2024; Wijmans et al., 2025). Vocabulary design is therefore a resource allocation problem: every slot added to one language or domain comes at the expense of coverage and efficiency elsewhere (Foroutan et al., 2025).

Standard tokenization, while effective, often leads to a disproportionate allocation of the vocabulary (§3). Common words occupy multiple token slots for their various forms (e.g., *walk*, *walks*, *walking*), leaving less room for uncommon words and multilingual coverage. This approach ultimately hurts both performance and inference costs (Petrov et al., 2023; Ahia et al., 2023; Ali et al., 2024). More fundamentally, it ignores a striking property of LLMs: their tendency to encode relationships between words as simple *linear directions* (Park et al., 2024; Marks and Tegmark, 2024). Our central question is whether this structure can be leveraged to build more compact and expressive vocabularies under a fixed size—allowing for more efficient tokenization across domains.

We begin by investigating how LLMs represent word form variation. Building on the idea of vector arithmetic in embedding space (Mikolov et al., 2013b), we examine whether common word-form transformations—including morphological inflection (*walk**ed***), derivation (*walk**able***) and capitalization (***W**alk*)—can be captured as consistent *transformation vectors* added to a *base form* word embedding (§4). Focusing on five morphologically diverse languages, we identify token pairs of base- and surface-form words exemplifying the same relation using UniMorph (Batsuren et al., 2022). We then compute the average offset vector for each relation, and use these as *transformation* vectors. Our results show that adding these vectors to *base form* embeddings yields representations that the model interprets similarly to the expected surface form (Ghandeharioun et al., 2024). Interestingly, this holds even when the target word is not represented as a single token in the vocabulary,[2] indicating that LLMs process and interpret word forms compositionally (§5).

Building on these insights, we propose a compact restructuring of the vocabulary, building word embeddings from shared components (Figure 1): a *base form* vector for the core lexical item and a *transformation* vector for encoding word-form variation. Rather than assigning a unique token embedding to each surface form, we remove inflected forms from the model's embedding tables. Instead, we introduce a small set of *transformation* embeddings—enabling us to represent the discarded words compositionally (e.g., *walked* as *walk*+*past tense*) in both input and output.

We study two practical regimes: lightweight post-hoc fine-tuning and compositional pretraining from scratch. In the post-hoc setting (§6), we only fine-tune the *transformation* embeddings and train LoRA adapters on the final $k = 8$ transformer blocks, leaving all other parameters frozen. Across five models and five languages, our method removes up to 10% of the vocabulary tokens while largely maintaining performance over a suite of downstream tasks. In pretraining proof-of-concept experiments (§7), we show that compositional vocabularies are even more effective when trained from scratch: they remove 41% of BPE vocabulary entries and obtain comparable performance, creating substantial room for new tokens.

In summary, we introduce compositional structure into language model vocabularies, enabling more efficient use of a fixed vocabulary budget through shared building blocks, which reduce redundancy in token allocation while also expanding lexical coverage. Our experiments demonstrate that LLMs can naturally operate over these representations, and establish compositional vocabularies as a competitive alternative to standard surface-form tokenization for future language models.

2 Background: Token Allocation in Language Model Vocabularies

Tokenization bridges natural language and model representations: it decomposes text into sequences of tokens from a fixed vocabulary, where each token is an atomic string unit for which the model learns specialized, single-vector embeddings. These vocabularies are almost universally built using byte-pair encoding (BPE; Sennrich et al., 2016), which iteratively merges the most frequent token pairs—from characters to subwords to words—in an attempt to optimally compress the text using a predetermined vocabulary size.

As LLM vocabularies grow larger (e.g., Gemma Team, 2024; Aryabumi et al., 2024), there is growing recognition that vocabulary resources can be better allocated. Recent studies point to stark imbalances in token allocations across languages, negatively impacting both model cost (Petrov et al., 2023; Ahia et al., 2023) and performance (Ali et al.,

[2] E.g., a word like "walkable" is split into `[_walk, able]`.

2024; Limisiewicz et al., 2023; Toraman et al., 2022). These findings motivate the development of techniques for post-hoc vocabulary expansion to reduce costs for a specific language or domain (Han et al., 2025; Nakash et al., 2025; Liu et al., 2024b; Minixhofer et al., 2024).

Another line of research advocates for scaling up the vocabulary together with model size, to unlock performance gains in the model's main language (Tao et al., 2024; Huang et al., 2025; Liu et al., 2025). Still, expansion is ultimately bounded by memory and compute constraints (Dagan et al., 2024; Wijmans et al., 2025), underscoring the importance of carefully reconsidering how the token vocabulary is allocated.

3 Word Structure and Redundancy in Vocabulary Design

One underexplored source of inefficiency in current vocabulary design is the treatment of morphologically related word forms as independent tokens. In high-resource languages like English, this often results in large clusters of surface variants—*walk, walks, walking, walked*—each assigned a separate token, despite their shared meaning and structure.

To quantify this redundancy, we examine the English whole-word tokens in the GPT-4 tokenizer (OpenAI, 2024)—the base tokenizer for many recent LLMs (Grattafiori et al., 2024; Yang et al., 2024; OLMo et al., 2024). We use UniMorph's English lexicon (Batsuren et al., 2022) to identify tokens that are English words,[3] finding 24.6k such tokens (Figure 2, left side).[4] Ignoring case (e.g., equating "walk" with "Walk") reduces this to 17.7k unique types. Further accounting for inflectional and derivational relations reduces this to just 14.3k *base forms*, a total reduction of 42%.

Rather than assigning each word form a distinct, independently learned token, what if we could model these processes as *transformations* applied to a compact set of base words? Our analysis shows that, beyond reconstructing every in-vocabulary word, these tokens can further represent 98k out-of-vocabulary words (Fig. 2, right), which are currently represented using multiple tokens.

[3]We only consider tokens that start with a leading space as whole word tokens; tokens without it can sometimes occur mid-word (like "ask" in "task", compared to "_ask").

[4]Out of 100k tokens, there are 41.3k tokens with a leading space in the vocabulary that are composed of English letters. Roughly 60% are identified as valid English words. The rest are either code-related terms, sub-words, or proper nouns. The other 60k tokens are either sub-words or non-English tokens.

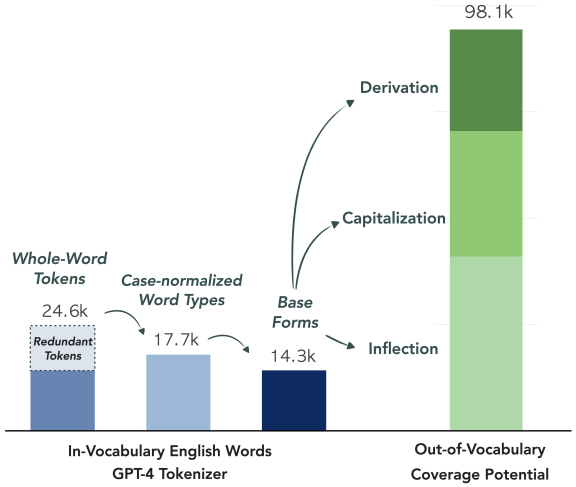

Figure 2: **Structure in LLM vocabularies and potential for compositional design. Left:** Many in-vocabulary English word tokens in the GPT-4 tokenizer are surface variants of other tokens—differing only by case, inflection, or derivation—reducing from 24k tokens to just 14k base-form words. **Right:** The existing set of base forms and transformations can be used to compose over 98k currently out-of-vocabulary words, highlighting the inefficiencies of current vocabularies and the potential of a compositional design.

Altogether, this motivates a structured vocabulary design that composes word forms from shared blocks, yielding vocabularies that are simultaneously more compact and more expressive while scaling effectively across domains and languages.

4 Composing Words from Base Forms and Transformations

We propose a compositional representation approach in which each surface form is constructed from a base word and a set of transformation vectors. Formally, let $\mathcal{V}_{\text{orig}}$ denote the model's original token vocabulary. We define a subset $\mathcal{V}_b \subset \mathcal{V}_{\text{orig}}$ as the base-word vocabulary, consisting of canonical lexical forms (e.g., *walk*) and any auxiliary tokens (e.g., punctuation, sub-words, code segments, words in non-target languages). We also introduce a transformation vocabulary $\mathcal{V}_t$, which consists of a small number of vectors corresponding to morphological operations such as inflection or derivation, or other word-level processes like capitalization.

In our scheme, a word w is represented by a base $b_w \in \mathcal{V}_b$ and a set of transformations $T(w) \subset \mathcal{V}_t$:

$$\mathbf{e}_w = \mathbf{e}_{b_w} + \sum_{t_i \in T(w)} \mathbf{e}_{t_i} \quad (1)$$

where $\mathbf{e}_{b_w}$ and $\mathbf{e}_{t_i}$ are rows from embedding ma-

trices E_b and E_t, respectively. We define the *compositional vocabulary* $\mathcal{V}$ as all words that can be constructed from $(b_w, T(w))$ combinations. For base words and auxiliary tokens, $T(w) = \varnothing$.

This decomposition applies both at input and output: At input, we replace direct lookup with Eq. 1. At output, we replace the model's large unembedding matrix U with two separate matrices for *base forms* and *transformations*: U_b and U_t. Given an output state $\mathbf{h}$, we score each candidate next-token w by separately projecting $\mathbf{h}$ onto U_b and U_t, and summing the relevant dot-products:

$$\text{logit}(w) = \mathbf{h} \cdot \mathbf{u}_{b_w} + \sum_{t_i \in T(w)} \mathbf{h} \cdot \mathbf{u}_{t_i} \qquad (2)$$

where $\mathbf{u}_{b_w}$ and $\mathbf{u}_{t_i}$ are the corresponding columns of U_b and U_t for w's components. To obtain the final next-token probabilities in the post-hoc setting, we apply a single softmax over the logits of all entries in $\mathcal{V}$ (as computed by Eq. 2). Importantly, our method is agnostic to whether w is originally in-vocabulary (IV) or out-of-vocabulary (OOV), as long as its base form is IV.

Vocabulary decomposition map. To apply this framework, we construct a mapping $w \mapsto (b_w,\ T(w))$ from surface forms to their base forms and matching transformations. We use UniMorph (Batsuren et al., 2022), a multilingual word form database, to identify base forms and their inflected and derived forms. Transformation labels are drawn from UniMorph's standardized tags (e.g., `V;PST`) with added rules for capitalization. Then, to build a decomposition map for a given tokenizer's vocabulary $\mathcal{V}_{\text{orig}}$, we iterate over its tokens, identify base forms, and map all related surface forms—whether in-vocabulary or not—to their base and transformation sets.

Notably, the decomposition map could also be built using sources other than morphological annotations, such as unsupervised morphological segmentation (Creutz and Lagus, 2002, 2007; Smit et al., 2014; Abdelali et al., 2016) or LLM-based morphological analyses (Pranjić et al., 2024). In this work, UniMorph serves as a clean experimental scaffold for testing whether models interpret and use these compositions correctly. Importantly, it is not a requirement of our framework itself.

Computing the transformation vectors. To initialize the transformation vectors themselves (i.e., the entries in E_t and U_t) in already-trained models, we revisit the idea of vector arithmetic in embedding space (Mikolov et al., 2013b). Let O be an embedding matrix of $\mathcal{V}_{\text{orig}}$, and let $b(w) : \mathcal{V}_{\text{orig}} \mapsto \mathcal{V}_{\text{b}}$ be a function that maps a word to its base form. For each transformation t, we extract the set $R(t) = \{(w, b(w)) \mid t \in T(w)\}$ of word pairs in $\mathcal{V}_{\text{orig}}$ that exemplify t (e.g., *walk* and *walked* for t = past tense).[5] We then compute the average offset of their respective embeddings:

$$\mathbf{o}_t = \frac{1}{|R(t)|} \sum_{w \in R(t)} (\mathbf{o}_w - \mathbf{o}_{b(w)}) \qquad (3)$$

We compute this separately for all $t \in \mathcal{V}_{\text{t}}$ in both the embedding and unembedding spaces, yielding *transformation* vectors for input and output. While prior work analyzed such linearity in the embeddings of LLMs (Park et al., 2024, 2025), to the best of our knowledge, our work is the first to leverage this for end-to-end language modeling.

5 Do LLMs Understand Compositional Word Representations?

We now turn to our first core question: can LLMs that were pretrained with standard vocabularies interpret our compositional embeddings—sums of *base form* and *transformation* vectors—as intended?

Recent work has shown LLMs build up and resolve the meanings of input tokens across their early layers, a process referred to as *detokenization* (Kaplan et al., 2025; Feucht et al., 2024; Gurnee et al., 2023). This was particularly observed for multi-token words or in-vocabulary words split into multiple tokens (e.g., due to typos). Building on this, we feed models with compositional inputs and inspect whether the embedding and early layer representations have successfully resolved into the intended surface form meanings. To interpret these internal representations, we follow Kaplan et al. and use Patchscopes (Ghandeharioun et al., 2024), a prompting method to probe the contents of a hidden state using natural language.

Languages and models. We experiment with five morphologically-diverse languages: English, Arabic, German, Russian and Spanish. For English, we use three LLMs: `Llama-3-8B` (Grattafiori et al., 2024), `Qwen2.5-7B` (Yang et al., 2024), and `OLMo-2-7B` (OLMo et al., 2024). As coverage of

[5] To obtain a "clean" signal for *transformations*, we only use w that demonstrate a *single* transformation ($|T(w)| = 1$).

whole-word tokens in these models' vocabularies for other languages is narrow,[6] we use models with dedicated tokenizers for them: `ALLaM-7B` for Arabic (Bari et al., 2025) and `EuroLLM-9B` for the three other languages (Martins et al., 2025).[7] In experiments for a specific model and language pair, we construct the vocabulary decomposition and transformation vectors (§4) only for that language, ignoring words in other languages.

Examining word representations. For each model and language pair, we iterate over all words w that could be composed from the *base forms* and *transformations* extracted from its vocabulary (§4). Next, given a surface form w, we replace the token embedding for w with its compositional representation e_w (Eq. 1), and feed it to Patchscopes to generate its textual description.[8] We then evaluate whether the Patchscopes interpretation of the compositional embedding e_w matches the target word w (*embed*). We also examine whether the model successfully *detokenizes* compositional embeddings in its early layers: we feed e_w to the model without any context, extract the resulting hidden states at the first $k = 10$ layers, and report whether the Patchscopes interpretation matches the target word w in at least one layer (*detok*).

English results. We begin by examining English words that exist as single tokens in `Llama-3-8B`'s original vocabulary $\mathcal{V}_{\text{orig}}$ (Table 1, *in-vocab*). We observe that most inflectional transformations—such as verb tense (past, present participle) and number (plural)—as well as capitalization, are often correctly resolved by the model already at the embedding layer (*embed*), and almost always at early internal layers (*detok*). For example, $\mathbf{e}_{\text{walk}} + \mathbf{e}_{\textit{past}}$ is interpreted by Patchscopes as "*walked*". In contrast, derivations (e.g., *walk→walkable*), which rarely occur as single tokens in the vocabulary, are seldom recognized by the model and often resolve as the base word instead. This suggests that models learn weaker linear structure for rare relations, or that *transformation* vectors built using small sample sizes show weaker generalization.

[6]This restricts both the base-word lexicon, and the number of existing *base-inflected* pairs for extracting transformations.

[7]All models have vocabularies of 100k or more tokens, except `ALLaM` with 64k (but roughly 32k are for Arabic).

[8]Following Kaplan et al. (2025), we use the Patchscopes prompt "`[X], [X], [X], [X],`", where we replace the placeholder token (`[X]`) with a hidden state $\mathbf{h}$ and let Patchscopes generate text. We expect Patchscopes to generate the intended word form if $\mathbf{h}$ indeed captures it. For languages other than English, we add the prefix "`In {language_name}:`".

Transformation	In-vocab.			Out-of-vocab.		
	embed	*detok*	*N*	*embed*	*detok*	*N*
Inflection						
Plural (N)	92%	96%	0.8k	30%	56%	3.4k
Plural (N) & Present Singular (V)	87%	91%	1.6k	43%	75%	2.1k
Present Singular (V)	90%	91%	0.1k	64%	82%	0.3k
Past (V)	71%	81%	0.6k	9%	29%	2.9k
Past Participle (V)	64%	93%	14	14%	38%	21
Gerund (V)	83%	93%	0.2k	17%	34%	3.2k
Superlative (ADJ)	71%	94%	31	5%	29%	0.4k
Comparative (ADJ)	40%	83%	30	3%	36%	0.4k
Capitalization	80%	89%	6.0k	72%	85%	8.4k
Derivation						
-y	24%	47%	17	2%	12%	1.5k
-er	8%	17%	12	0%	6%	2.6k
-al	25%	25%	8	0%	9%	0.7k
un-	0%	33%	3	0%	2%	3.3k
re-	67%	67%	3	0%	10%	1.8k
-ic	100%	100%	2	4%	21%	0.4k
All derivatives	31%	45%	51	0%	3%	31.4k

Table 1: Accuracy of Patchscopes interpretations for compositional input representations (i.e., *base form* + *transformation* embeddings) of in-vocabulary and out-of-vocabulary English words in `Llama-3.1-8B`. We report successful resolution both at the embedding layer (*embed*), and after detokenization in early layers (*detok*). N indicates the number of surface forms evaluated per category. Compositional embeddings of capitalization and inflectional forms are very often resolved correctly—even for many out-of-vocabulary words, which never occur as single input vectors during pretraining. Derivatives remain challenging—likely because they rarely occur as in-vocabulary words.

We next examine out-of-vocabulary words, i.e., English words that can be composed using the *base forms* and *transformations* but are *not* found as a single token in the original vocabulary (Table 1, *out-of-vocab*). Using our decomposition map, we construct single-vector representations for these words and feed them to the model. Surprisingly, many of these are resolved as the intended word form already at the embedding layer, with Patchscopes generating the full, multi-token word, especially for inflections and capitalization. Similarly to in-vocabulary results, we observe higher successful resolution rates for early-layer detokenization, while representing out-of-vocabulary derivations compositionally generally fails. We observe similar results for English in other models.[9]

Multilingual results. We repeat the same experiment on each of the other languages. Since each language has different types and number of inflectional and derivational processes,[10] we aggregate

[9]See Appendix A.1 for further English results, and Appendix B.1 for analysis of *transformation* errors and geometry.

[10]We treat each UniMorph tag as its own *transformation*.

	Language	Capitalization		Noun Inflection		Adjective Inflection		Verb Inflection		Derivation	
		In-Vocab.	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*
ALLaM	Arabic	—	—	77% (1.8k)	14% (3.6k)	69% (0.5k)	23% (1.0k)	41% (1.0k)	14% (2.7k)	—	—
EuroLLM	German	95% (0.2k)	74% (0.4k)	—	—	21% (0.3k)	7% (1.3k)	82% (0.3k)	36% (1.2k)	—	—
	Russian	97% (66)	88% (0.7k)	63% (0.6k)	21% (4.2k)	100% (50)	89% (94)	83% (6)	30% (10)	—	—
	Spanish	97% (1.0k)	90% (2.8k)	76% (0.7k)	46% (1.9k)	79% (0.5k)	60% (1.1k)	67% (0.8k)	35% (6.9k)	37% (65)	14% (0.4k)
Llama-3	English	80% (6.0k)	72% (8.4k)	89% (2.4k)	35% (5.6k)	56% (61)	4% (0.9k)	76% (0.9k)	16% (6.4k)	20% (41)	0% (12.8k)

Table 2: Accuracy of Patchscopes interpretations for compositional input embeddings across languages. Numbers in parentheses indicate sample sizes. "—" indicates cases where no suitable *base-inflection* pairs were found in the vocabulary or where there are no UniMorph entries for that category. For detokenization results, see Appendix A.2.

results over five categories: adjective inflection, verb inflection, noun inflection, derivation and capitalization. Our results (Table 2) show that LLMs can correctly interpret compositional word representations across diverse languages and morphological structures. Surprisingly, some *transformation* vector types (e.g., adjective or verb inflections) work better for out-of-vocabulary representation than in English, hinting that models learn stronger linear encodings of morphological structure when the token vocabulary is more limited—a phenomenon we further analyze in §8. Overall, our results show that LLMs can naturally interpret compositional word embeddings across languages.

Analysis of composition failures. Across languages and models, we observe a consistent gap between inflectional transformations (often resolved) and derivational transformations (rarely resolved). To characterize these failures, we analyze whether the number of in-vocabulary exemplar pairs used to estimate each *transformation* vector (Eq. 3) helps explain composition failures. We find that the number of exemplars mainly matters for *generalization*: *transformation* vectors estimated from many pairs are much more likely to resolve for multi-token surface forms, while in-vocabulary success is overall insensitive to exemplar count once a usable signal is available (see Appendix B.2).

6 Compositional Language Modeling

We have shown that *transformation* vectors capture meaningful operations in the input space of LLMs, and that these can be successfully composed with base word embeddings. We next investigate whether models can use compositional vocabularies effectively in end-to-end language modeling.

6.1 Implementation and Experimental Setup

Given a model's vocabulary decomposition map (§4), we apply our compositional vocabulary framework and restructure the input and output embedding matrices. We replace the model's input embedding of any surface form w with compositions of the corresponding *base form* and *transformation* embeddings (Eq. 1). For next-token prediction, we compute logits through summation of *base form* and *transformation* logits (Eq. 2). Importantly, any word not in the decomposition map maintains its original embedding and unembedding throughout training and inference, without modifications.

Fine-tuning the transformation vectors. After initialization (Eq. 3), we train the *transformation* vectors jointly within the model: we treat the *transformation* embedding and unembedding matrices E_t and U_t as trainable weights (introducing fewer than 0.001% additional parameters), and freeze all other model parameters, including the embeddings and unembeddings of *base forms*. We use knowledge distillation loss (Hinton et al., 2015) to fine-tune the *transformation* vectors using two-stage distillation: We first freeze the output unembeddings and only train the *input transformations*, using the predictions of the original, unmodified model as targets. Next, we freeze the input embeddings and only train the *output transformations*, this time using the (frozen) model resulting from the first stage as the distillation target—ignoring all words $w \notin \mathcal{V}_{\text{orig}}$ in the loss. In both stages, we train on a fixed, small sample of the `FineWeb-Edu` corpus (Penedo et al., 2024).[11] See Appendix C.1.

Lightweight LoRA adaptation. To allow lightweight adaptation to the reshaped output vocabulary, we add LoRA adapters to the final $k = 8$ model layers, keeping all other internal layers frozen. We use LoRA $r = \alpha = 256$.

Filtering the decomposition map. Our results in §5 indicate some out-of-vocabulary surface forms fail to be interpreted by the model as their

[11] We use a sequence length of 256 and train on ∼5M tokens.

intended word when given as compositions. We therefore filter out surface words with failed detokenization from the decomposition map, and fall back to using their original tokenization and embeddings in both input and output. We also exclude all derivational transformations due to their weak resolution rates. See analysis in Appendix B.3.

Downstream tasks. We evaluate our compositional vocabulary models on a diverse suite of standard benchmarks. As a baseline, we compare performance to the original, unmodified models. For English, the benchmarks cover knowledge, reading comprehension, and commonsense: *MMLU* (Hendrycks et al., 2021), *ARC* (Clark et al., 2018), *HellaSwag* (Zellers et al., 2019), *Winogrande* (Sakaguchi et al., 2021), *TriviaQA* (Joshi et al., 2017), *SQuAD* (Rajpurkar et al., 2016), *BoolQ* (Clark et al., 2019), *PIQA* (Bisk et al., 2020) and *COPA* (Kavumba et al., 2019). For other languages, we use *XNLI* (Conneau et al., 2018), *XQuAD* (Artetxe et al., 2020) and *Global MMLU* (Singh et al., 2025). See Appendix D.

6.2 Results

We report our results for English on `Llama-3-8B` in Table 3,[12] and results for other languages in Table 4. Our compositional language modeling approach results in minimal degradation compared to the baseline models across languages, indicating that LLMs can leverage compositional vocabularies with only lightweight adaptation.

We further inspect reductions in vocabulary size after applying our framework. For English, our approach removes roughly 10k surface-form tokens from `Llama3` and `OLMo2` each, and 7.8k from `Qwen2.5`.[13] This frees a meaningful number of vocabulary slots for reallocation: recent work has shown that adding even several hundred dedicated tokens to the vocabulary can greatly improve tokenization efficiency and downstream behavior for a language or expert domain (Ahia et al., 2023; Liu et al., 2024a; Nakash et al., 2025). Notably, our method has a marginal effect on decoding speed—only a 0.8% reduction compared to standard prediction (see Appendix B.4).

[12] See Appendix A.3 for results on other English models.

[13] In other languages, absolute reductions are smaller (0.6k–3k) but correspond to 38–45% of whole-word tokens in the target languages, as these tokenizers devote far fewer whole-word entries to non-English languages to begin with.

Category	Task	Baseline	End-to-end	Δ
Knowledge	MMLU (Acc.)	65.2	64.9	-0.3
	ARC (Acc.)	53.6	52.5	-1.1
Reading Comprehension	BoolQ (Acc.)	83.2	83.3	+0.1
	TriviaQA (EM)	66.5	63.3	-3.3
	SQuAD (EM)	22.1	20.0	-2.1
Commonsense	Hellaswag (Acc.)	60.6	59.5	-1.1
	Winogrande (Acc.)	78.1	78.6	+0.5
	PIQA (Acc.)	80.3	79.1	-1.2
	COPA (Acc.)	93.0	92.0	-1.0
Average		**66.9**	**65.9**	-1.0

Table 3: Downstream performance of English compositional-vocabulary models (*End-to-end*) and their original, unmodified version (*Baseline*) for `Llama-3.1-8B`. Our framework remains competitive with the baseline despite extensive changes to the model's input and output representation mechanisms—highlighting the intrinsic ability of LLMs to process and predict words compositionally.

		XNLI	Δ	XQuAD	Δ	GMMLU	Δ
ALLaM	Arabic	44.1	-0.3	42.7	-3.2	59.9	+0.2
EuroLLM	German	46.5	+0.6	51.3	-1.6	54.6	-0.7
	Russian	40.1	-4.5	37.4	-3.6	54.4	-0.3
	Spanish	43.3	-0.6	48.3	-4.1	55.2	-0.9

Table 4: Multilingual downstream performance of compositional-vocabulary models, along with absolute performance difference from the baseline model (Δ).

Reallocating freed vocabulary slots. To make the practical gains concrete, we simulate token reallocation based on our results for the `Llama-3.1-8B` tokenizer. After evicting the 10k English surface words that can be represented compositionally, we add 2.5k new language-specific BPE tokens for each of: Arabic, Russian, German, and Spanish. We measure compression using bytes-per-token (BPT) on held-out `FineWeb-2` (Penedo et al., 2025) text. After reallocation, BPT improves from 4.40 to 4.81 on average across languages (Table 5).

In the next section, we show that vocabularies can also be built compositionally from the outset, with even greater vocabulary-allocation efficiency, by pretraining a model with a compositional vocabulary from scratch (§7).

7 Compositional Vocabulary Pretraining

To demonstrate that compositional vocabularies can also serve as a *design choice* when training new language models, we reshape English and Spanish BPE vocabularies into compositional ones, and pretrain small baseline and compositional models from scratch. For English, we reshape the 50k-token `GPT-2` tokenizer (Radford et al., 2019),

Language	Baseline	Reallocated	Δ (%)
Arabic	4.62	5.46	+18.0
Russian	5.59	5.85	+4.8
German	3.59	3.86	+7.5
Spanish	3.80	4.07	+7.0
Average	4.40	4.81	+9.3

Table 5: Tokenization efficiency, measured in bytes-per-token (*higher is better*), before and after reallocating token slots with our approach. Starting from the `Llama-3.1-8B` tokenizer, we replace 10k English surface forms that are represented compositionally with 2.5k new, non-overlapping BPE tokens for each language, keeping the total vocabulary size fixed.

		BPB ↓		Bytes/tok. ↑	
Language	Vocab. red.	Base	Comp.	Base	Comp.
English	41.6%	1.08	1.09	–	–
Spanish	41.8%	1.00	1.11	4.77	4.92

Table 6: Pretraining results for baseline (*Base*) and compositional (*Comp.*) models based on the same BPE vocabulary. Lower bits-per-byte (BPB) is better; higher bytes-per-token indicate more efficient tokenization. For Spanish, we further extend the compositional model to previously out-of-vocabulary word compositions, resulting in better compression.

while restricting the compositional model to predict exactly the same surface-form vocabulary as the BPE baseline (i.e., we do not extend to out-of-vocabulary words). For Spanish, we train a 32k-token BPE vocabulary[14] and then reshape it, this time allowing the compositional model to generate out-of-vocabulary surface forms via compositions. For each language and vocabulary, we pretrain a `nanoGPT-124M` model (Jordan et al., 2024) on 1B tokens,[15] comparing a baseline model against an otherwise-identical compositional model.

In contrast to our post-hoc setup, the compositional model predicts tokens in a factorized space, where a surface word w is predicted by first sampling from a *base form* distribution, and then predicting *transformations* conditioned on a chosen *base*:

$$p(w \mid \mathbf{h}) = p(b_w \mid \mathbf{h})\, p(T(w) \mid b_w, \mathbf{h}) \quad (4)$$

We further include a space-prefix *transformation* (e.g., "*_walking*" vs. "*walking*").[16] We measure performance using bits-per-byte (BPB) on a held-out set, as it is well-defined across different vocabularies and tokenizers.[17] To measure tokenization efficiency in Spanish, we use average bytes-per-token (higher is better). See Appendix C.2 for the exact training and modeling details.

We report our results in Table 6. In both languages, our approach frees roughly 42% of vocabulary entries compared to the original tokenizers. English shows comparable performance under this more compact parameterization, whereas Spanish shows a small BPB gap alongside more efficienct tokenization—while using an overall much smaller vocabulary.

Together, these results show that compositional vocabularies can be trained from scratch effectively, offering compact vocabularies and improved tokenization efficiency for future language models.

8 Morphology in Embedding Space Scales Inversely with Vocabulary Size

Having established that models implicitly learn compositional word representations, and with recent calls to scale vocabularies even further, a natural question emerges: how does vocabulary size affect the way models encode linguistic structure?

To study this question, we evaluate the extent of compositional word representations across models with varying vocabulary sizes. For each model, we decompose its vocabulary and measure the average Patchscopes interpretation accuracy for each *transformation* vector we extract (as in §5). We also separate models by their embedding architecture (*untied* vs. *tied*). We track each model's English vocabulary size (the subset of tokens present in English UniMorph), and plot the results in order of increasing vocabulary size. The English vocabulary size of these models spans 8k–44k tokens with total vocabulary sizes of 32k–256k tokens, representing varied scales of vocabulary design. See full model details in Appendix B.5.

Our results (Figure 3) reveal a general inverse relationship: models with compact English vocabularies (8–10k words, e.g., `Llama2`, `Mistral`) tend to encode morphology through consistent vector offsets that generalize across words. In contrast, large-vocabulary models (∼40k words, e.g., `Falcon3`, `Gemma2-9B`) tend to represent inflected forms of the same type as individual lexical units, rather than through a shared linear translation of

[14] We train the Spanish tokenizer on 10B bytes from the Spanish subset of FineWeb-2 (Penedo et al., 2025)

[15] We use FineWeb (English) and FineWeb-2 (Spanish).

[16] Modern BPE vocabularies include prefix whitespace characters when merging tokens, creating many near-duplicates.

[17] BPB normalizes negative log-likelihood by the number of UTF-8 bytes in the evaluation text.

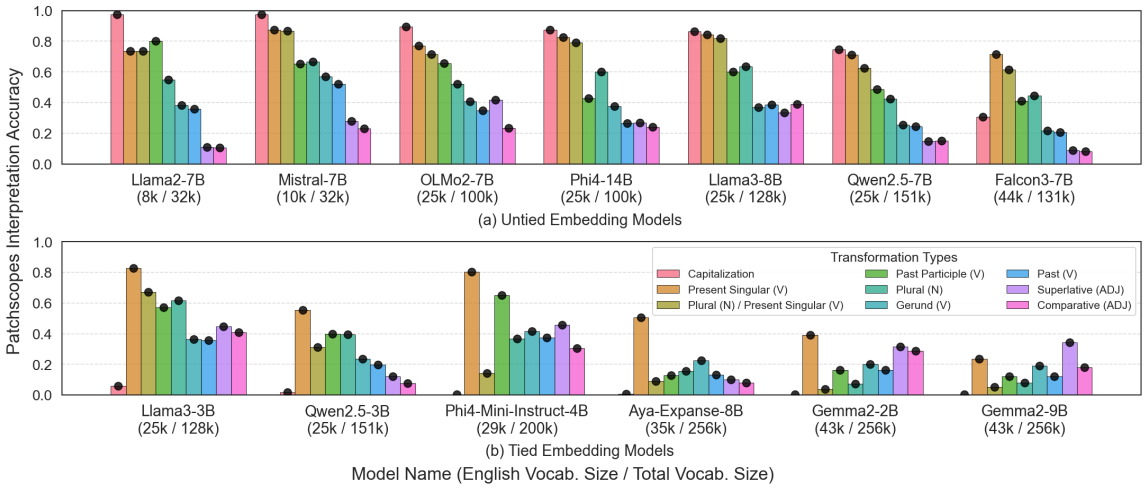

Figure 3: **Linear representation of morphology in embeddings weakens as vocabulary size increases.** Accuracy of Patchscopes interpretations of compositional word representations across models, in order of increasing English vocabulary size (English tokens present in UniMorph), separated by embedding architecture. Scaling vocabulary size leads models to represent inflected forms as individual lexical units, rather than with consistent vector offsets.

their base forms, with weight tying further amplifying this trend. Overall, these results suggest that vocabulary scaling trades morphological compositionality in embedding space for lexical memorization.[18]

9 Related Work

Incorporating morphology into representations A longstanding goal in NLP has been to integrate morphological knowledge into models. Early work on Transformer language models explored injecting linguistic features post-hoc (Hofmann et al., 2021; Gan et al., 2022) or during pretraining (Park et al., 2021; Cui et al., 2022; Matthews et al., 2018; Blevins and Zettlemoyer, 2019; Hofmann et al., 2020; Seker et al., 2022; Peng et al., 2019), but such approaches are absent in modern LLMs. Recent work examined word segmentation effects on performance (Marco and Fraser, 2024; Lerner and Yvon, 2025), as well as morphology-aware tokenization to better reflect word structure (Bauwens and Delobelle, 2024; Asgari et al., 2025). Rather than injecting linguistic structure, we leverage compositional representations already present in LLMs.

Vector arithmetic of word representations Linear structure in word representations was first observed in Word2Vec (Mikolov et al., 2013a,b; Levy and Goldberg, 2014; Vylomova et al., 2015). Recent work found similar structures in LLMs across the unembedding layer (Park et al., 2024, 2025), residual stream (Merullo et al., 2023; Hendel et al., 2023; Todd et al., 2024), and in behavior-steering directions (Subramani et al., 2022; Hernandez et al., 2024). We further show that such structure is usable for end-to-end language modeling. Beyond morphology, *transformation* vectors could capture semantic relations (e.g., country–nationality; Gladkova et al., 2016) or tie word embeddings across languages (Schut et al., 2025).

Post-hoc vocabulary modification Recent work has proposed methods to expand or modify token vocabulary by training new embeddings and fine-tuning internal model layers (Kim et al., 2024; Takase et al., 2024; Han et al., 2025; Minixhofer et al., 2024; Ben-Artzy and Schwartz, 2025; Dobler and de Melo, 2023). We avoid continual pretraining of model weights, and represent new forms by using the model's existing linguistic knowledge.

10 Conclusion

We have shown that word representations in LLMs are inherently compositional, and leveraged this property to introduce compositional vocabularies. Such vocabularies are more compact in size and more expressive in lexical coverage—freeing token slots that can be reallocated to words, languages, and domains that are currently tokenized inefficiently. Our results demonstrate that by integrating compositional vocabularies into future models, LLMs could cover more words, languages, and domains, without sacrificing performance.

[18] Importantly, this does not imply that large-vocabulary models lack morphological knowledge, only that they rely less on linearly encoded morphology in their embedding space.

Limitations

Our framework employs external morphological resources to define transformation pairs. While this allows for clean experimental control, it limits immediate applicability to languages or domains lacking annotated morphological data. However, UniMorph serves as an experimental scaffold in this paper, not as a requirement of the framework itself: in principle, the method only needs a decomposition map, which could come from unsupervised segmentation, statistical morphology learning, or bootstrapped analyses. In future work, we will explore whether transformation vectors can be induced directly from data in an unsupervised fashion.

Post-hoc adaptation is bounded by what pretrained models already encode reliably enough for linear composition. This is most visible in derivational morphology, where composed input representations are often interpreted as the base form. Importantly, we do not observe such issues when pretraining models with compositional vocabularies from scratch.

Our vocabulary reshaping approach also assumes a relatively simple decomposition of each surface form into a base word and a set of transformation vectors. While effective for many cases, this simplification does not account for certain words which admit multiple plausible morphological analyses. Still, these problems are also encountered with standard tokenization approaches, with models learning to disambiguate such words into their intended meanings.

Acknowledgments

We thank Guy Peskin and Amit Ben-Artzy for valuable conversations about this work. We are also grateful to the reviewers for their constructive and thoughtful feedback. This work was supported in part by the Israel Science Foundation (grant no. 2045/21) and by NSF-BSF grant 2020793.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

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.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Richard Rutmann, Max Lübbering, Johannes Leveling, Katrin Klug, Jan Ebert, Niclas Doll, Jasper Buschhoff, Charvi Jain, Alexander Weber, Lena Jurkschat, Hammam Abdelwahab, Chelsea John, Pedro Ortiz Suarez, Malte Ostendorff, Samuel Weinbach, Rafet Sifa, and 2 others. 2024. Tokenizer choice for LLM training: Negligible or crucial? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3907–3924, Mexico City, Mexico. Association for Computational Linguistics.

Mohamed Taher Alrefaie, Nour Eldin Morsy, and Nada Samir. 2024. Exploring tokenization strategies and vocabulary sizes for enhanced arabic language models. *arXiv preprint arXiv:2403.11130*.

Zaid Alyafeai, Maged S. Al-Shaibani, Mustafa Ghaleb, and Irfan Ahmad. 2021. Evaluating various tokenizers for arabic text classification. *Neural Processing Letters*, 55:2911–2933.

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.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, and 1 others. 2024. Aya 23: Open weight releases to further multilingual progress. *arXiv preprint arXiv:2405.15032*.

Ehsaneddin Asgari, Yassine El Kheir, and Mohammad Ali Sadraei Javaheri. 2025. MorphBPE: A morpho-aware tokenizer bridging linguistic complexity for efficient llm training across morphologies. *Preprint*, arXiv:2502.00894.

M Saiful Bari, Yazeed Alnumay, Norah A. Alzahrani, Nouf M. Alotaibi, Hisham Abdullah Alyahya, Sultan AlRashed, Faisal Abdulrahman Mirza, Shaykhah Z. Alsubaie, Hassan A. Alahmed, Ghadah Alabduljabbar, Raghad Alkhathran, Yousef Almushayqih, Raneem Alnajim, Salman Alsubaihi, Maryam Al Mansour, Saad Amin Hassan, Dr. Majed Alrubaian, Ali Alammari, Zaki Alawami, and 7 others. 2025. ALLam: Large language models for arabic and english.

In *The Thirteenth International Conference on Learning Representations*.

Khuyagbaatar Batsuren, Omer Goldman, Salam Khalifa, Nizar Habash, Witold Kieraś, Gábor Bella, Brian Leonard, Garrett Nicolai, Kyle Gorman, Yustinus Ghanggo Ate, Maria Ryskina, Sabrina Mielke, Elena Budianskaya, Charbel El-Khaissi, Tiago Pimentel, Michael Gasser, William Abbott Lane, Mohit Raj, Matt Coler, and 76 others. 2022. UniMorph 4.0: Universal Morphology. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 840–855, Marseille, France. European Language Resources Association.

Thomas Bauwens and Pieter Delobelle. 2024. BPE-knockout: Pruning pre-existing BPE tokenisers with backwards-compatible morphological semi-supervision. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5810–5832, Mexico City, Mexico. Association for Computational Linguistics.

Amit Ben-Artzy and Roy Schwartz. 2025. Spellm: Character-level multi-head decoding. *Preprint*, arXiv:2507.16323.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Terra Blevins and Luke Zettlemoyer. 2019. Better character language modeling through morphology. *arXiv preprint arXiv:1906.01037*.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/Daily Mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. 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 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457.

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.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):1–34.

Yiming Cui, Wanxiang Che, Shijin Wang, and Ting Liu. 2022. Lert: A linguistically-motivated pre-trained language model. *arXiv preprint arXiv:2211.05344*.

Gautier Dagan, Gabriele Synnaeve, and Baptiste Rozière. 2024. Getting the most out of your tokenizer for pre-training and domain adaptation. *arXiv preprint arXiv:2402.01035*.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, and 1 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

Konstantin Dobler and Gerard de Melo. 2023. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13440–13454, Singapore. Association for Computational Linguistics.

Sheridan Feucht, David Atkinson, Byron C Wallace, and David Bau. 2024. Token erasure as a footprint of implicit vocabulary items in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9727–9739, Miami, Florida, USA. Association for Computational Linguistics.

Negar Foroutan, Clara Meister, Debjit Paul, Joel Niklaus, Sina Ahmadi, Antoine Bosselut, and Rico Sennrich. 2025. Parity-aware byte-pair encoding: Improving cross-lingual fairness in tokenization.

Guobing Gan, Peng Zhang, Sunzhu Li, Xiuqing Lu, and Benyou Wang. 2022. Morphte: Injecting morphology in tensorized embeddings. *Advances in Neural Information Processing Systems*, 35:33186–33200.

Bar Gazit, Shaltiel Shmidman, Avi Shmidman, and Yuval Pinter. 2025. Splintering nonconcatenative languages for better tokenization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22405–22417, Vienna, Austria. Association for Computational Linguistics.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*.

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't. In *Proceedings of the NAACL Student Research Workshop*, pages 8–15.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Wes Gurnee, Neel Nanda, Matthew Pauly, Katherine Harvey, Dmitrii Troitskii, and Dimitris Bertsimas. 2023. Finding neurons in a haystack: Case studies with sparse probing. *Transactions on Machine Learning Research*.

HyoJung Han, Akiko Eriguchi, Haoran Xu, Hieu Hoang, Marine Carpuat, and Huda Khayrallah. 2025. Adapters for altering LLM vocabularies: What languages benefit the most? In *The Thirteenth International Conference on Learning Representations*.

Roee Hendel, Mor Geva, and Amir Globerson. 2023. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9318–9333, Singapore. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2024. Inspecting and editing knowledge representations in language models. In *First Conference on Language Modeling*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2020. DagoBERT: Generating derivational morphology with a pretrained language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3848–3861, Online. Association for Computational Linguistics.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Superbizarre is not superb: Derivational morphology improves BERT's interpretation of complex words. 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 3594–3608, Online. Association for Computational Linguistics.

Valentin Hofmann, Hinrich Schuetze, and Janet Pierrehumbert. 2022. An embarrassingly simple method to mitigate undesirable properties of pretrained language model tokenizers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–393, Dublin, Ireland. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Xun Zhou. 2025. Overtokenized transformer: Vocabulary is generally worth scaling. *arXiv preprint arXiv:2501.16975*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. 2024. modded-nanogpt: Speedrunning the nanogpt baseline.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Guy Kaplan, Matanel Oren, Yuval Reif, and Roy Schwartz. 2025. From tokens to words: On the inner lexicon of LLMs. In *The Thirteenth International Conference on Learning Representations*.

Pride Kavumba, Naoya Inoue, Benjamin Heinzerling, Keshav Singh, Paul Reisert, and Kentaro Inui. 2019. When choosing plausible alternatives, clever hans can be clever. In *Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing*, pages 33–42, Hong Kong, China. Association for Computational Linguistics.

Seungduk Kim, Seungtaek Choi, and Myeongho Jeong. 2024. Efficient and effective vocabulary expansion towards multilingual large language models. *arXiv preprint arXiv:2402.14714*.

Stav Klein and Reut Tsarfaty. 2020. Getting the ##life out of living: How adequate are word-pieces for modelling complex morphology? In *Proceedings of the*

17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pages 204–209, Online. Association for Computational Linguistics.

Paul Lerner and François Yvon. 2025. Unlike "likely", "unlike" is unlikely: BPE-based segmentation hurts morphological derivations in LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5181–5190, Abu Dhabi, UAE. Association for Computational Linguistics.

Omer Levy and Yoav Goldberg. 2014. Linguistic regularities in sparse and explicit word representations. In *Conference on Computational Natural Language Learning*.

Tomasz Limisiewicz, Jivr'i Balhar, and David Marevcek. 2023. Tokenization impacts multilingual language modeling: Assessing vocabulary allocation and overlap across languages. In *Annual Meeting of the Association for Computational Linguistics*.

Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A. Smith, and Yejin Choi. 2025. SuperBPE: Space travel for language models. *ArXiv*, abs/2503.13423.

Chengyuan Liu, Shihang Wang, Lizhi Qing, Kun Kuang, Yangyang Kang, Changlong Sun, and Fei Wu. 2024a. Gold panning in vocabulary: An adaptive method for vocabulary expansion of domain-specific LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7442–7459, Miami, Florida, USA. Association for Computational Linguistics.

Yihong Liu, Peiqin Lin, Mingyang Wang, and Hinrich Schuetze. 2024b. OFA: A framework of initializing unseen subword embeddings for efficient large-scale multilingual continued pretraining. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1067–1097, Mexico City, Mexico. Association for Computational Linguistics.

Manuel Mager, Arturo Oncevay, Elisabeth Mager, Katharina Kann, and Thang Vu. 2022. BPE vs. morphological segmentation: A case study on machine translation of four polysynthetic languages. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 961–971, Dublin, Ireland. Association for Computational Linguistics.

Marion Di Marco and Alexander Fraser. 2024. Subword segmentation in LLMs: Looking at inflection and consistency. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12050–12060, Miami, Florida, USA. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M Guerreiro, Ricardo Rei, Duarte M Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, and 1 others. 2025. Eurollm: Multilingual language models for europe. *Procedia Computer Science*, 255:53–62.

Austin Matthews, Graham Neubig, and Chris Dyer. 2018. Using morphological knowledge in open-vocabulary neural language models. 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 1435–1445.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. Language models implement simple word2vec-style vector arithmetic. *arXiv preprint arXiv:2305.16130*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Distributed representations of words and phrases and their compositionality. In *Neural Information Processing Systems*.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013b. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Benjamin Minixhofer, Edoardo Ponti, and Ivan Vulić. 2024. Zero-shot tokenizer transfer. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Itay Nakash, Nitay Calderon, Eyal Ben-David, Elad Hoffer, and Roi Reichart. 2025. Adaptivocab: Enhancing LLM efficiency in focused domains through lightweight vocabulary adaptation. In *Second Conference on Language Modeling*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. 2 olmo 2 furious. *ArXiv*, abs/2501.00656.

OpenAI. 2024. tiktoken: A fast BPE tokeniser for use with openai's models.

Hyunji Hayley Park, Katherine J. Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: A multilingual language modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Kiho Park, Yo Joong Choe, Yibo Jiang, and Victor Veitch. 2025. The geometry of categorical and hierarchical concepts in large language models. In *The Thirteenth International Conference on Learning Representations*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 39643–39666. PMLR.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, and 1 others. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849.

Guilherme Penedo, Hynek Kydlíček, Vinko Sabolčec, Bettina Messmer, Negar Foroutan, Amir Hossein Kargaran, Colin Raffel, Martin Jaggi, Leandro Von Werra, and Thomas Wolf. 2025. Fineweb2: One pipeline to scale them all — adapting pre-training data processing to every language. In *Second Conference on Language Modeling*.

Hao Peng, Roy Schwartz, and Noah A. Smith. 2019. PaLM: A hybrid parser and language model. 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 3644–3651, Hong Kong, China. Association for Computational Linguistics.

Aleksandar Petrov, Emanuele La Malfa, Philip Torr, and Adel Bibi. 2023. Language model tokenizers introduce unfairness between languages. *Advances in neural information processing systems*, 36:36963–36990.

Marko Pranjić, Marko Robnik-Šikonja, and Senja Pollak. 2024. LLMSegm: Surface-level morphological segmentation using large language model. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10665–10674, Torino, Italia. ELRA and ICCL.

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

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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. 2025. Do multilingual llms think in english? *arXiv preprint arXiv:2502.15603*.

Amit Seker, Elron Bandel, Dan Bareket, Idan Brusilovsky, Refael Greenfeld, and Reut Tsarfaty. 2022. AlephBERT: Language model pre-training and evaluation from sub-word to sentence level. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 46–56, Dublin, Ireland. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, and 4 others. 2025. Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, and Mikko Kurimo. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 21–24, Gothenburg, Sweden. Association for Computational Linguistics.

Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.

Sho Takase, Ryokan Ri, Shun Kiyono, and Takuya Kato. 2024. Large vocabulary size improves large language models. *arXiv preprint arXiv:2406.16508*.

Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. 2024. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Falcon-LLM Team. 2024. The falcon 3 family of open models.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiroaju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

Cagri Toraman, Eyup Halit Yilmaz, Furkan Şahinuç, and Oguzhan Ozcelik. 2022. Impact of tokenization on language models: An analysis for turkish. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22:1 – 21.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Reut Tsarfaty, Shoval Sadde, Stav Klein, and Amit Seker. 2019. What's wrong with Hebrew NLP? and how to make it right. 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): System Demonstrations*, pages 259–264, Hong Kong, China. Association for Computational Linguistics.

Ekaterina Vylomova, Laura Rimell, Trevor Cohn, and Timothy Baldwin. 2015. Take and took, gaggle and goose, book and read: Evaluating the utility of vector differences for lexical relation learning. *ArXiv*, abs/1509.01692.

Erik Wijmans, Brody Huval, Alexander Hertzberg, Vladlen Koltun, and Philipp Kraehenbuehl. 2025. Cut your losses in large-vocabulary language models. In *The Thirteenth International Conference on Learning Representations*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

A Supplementary Results

A.1 English Patchscopes Results

For the results of the Patchscopes experiments on other models, see Table 10 and Table 11.

A.2 Multilingual Patchscopes Results

For the results on multilingual Patchscopes interpretations of compositional input embeddings after detokenization, see Table 9.

A.3 Post-hoc Adaptation Results

For post-hoc adaptation results on other models, see Table 12 and Table 13.

B Additional Analysis

B.1 Analyses of Patchscopes Results

Patchscopes error breakdown. We classify each Patchscopes generation of `Llama-3.1-8B` for English single-token *base*+*transformation* targets into four outcomes: exact match of target, exact match of base form, exact match different inflection of the same base, and other (Table 7). The dominant error is collapse to the base form—showing the model either interprets the compositional representation correctly, or as the corresponding base word. We also verify that when the target is already a base form, Patchscopes almost always returns that same base form rather than an inflected variant.

Geometry of offset vectors. Our Patchscopes and end-to-end language modeling results indicate that language models often represent words compositionally. We further compare each pairwise offset (e.g., *walked-walk*) to its own *transformation* category vector versus other *transformation* vectors using cosine similarity for `Llama-3.1-8B`. The resulting separation is clear in both input and output spaces (Table 8): offsets are consistently closer to their own *transformation* type than to others, and top-1 *transformation* accuracy is high, indicating separability between *transformations*. Still, while models *functionally* operate as if each *transformation* vector as a single crisp direction in embedding space (as shown in the Patchscopes and end-to-end language modeling experiments), these results indicate this is an over-simplification.

B.2 Exemplar Count Predicts Out-of-Vocabulary Generalization for Transformation Vectors

We test whether the number of in-vocabulary exemplar pairs used to estimate each *transformation* vector (Eq. 3) predicts whether the resulting composed embedding is interpreted as the intended surface form. For each *transformation* t, we use the number of single-token in-vocabulary (IV) base/surface pairs as a proxy for exemplar set size, and compute Spearman correlations with additive success on IV targets and on out-of-vocabulary (OOV; multi-token) targets separately.

For `Llama-3.1-8B`, across individual transformations ($n = 24$), IV additive success is nearly independent of exemplar count (Spearman's $\rho = 0.04$). By contrast, IV exemplar count strongly predicts OOV performance ($\rho = 0.77$, $p = 1.5 \times 10^{-5}$). This dissociation suggests that exemplar richness is not the main bottleneck for IV targets, which largely saturate once a direction is available, but is important for generalization to multi-token OOV surface forms.

Restricting the analysis to transformations within each coarse class yields the same qualitative pattern, though with limited power due to small n. Among inflectional transformations ($n = 8$), IV success correlates moderately with the number of IV exemplars ($\rho = 0.50$). Among derivational transformations ($n = 14$), IV success instead trends negative ($\rho = -0.44$), consistent with the observation that derivations remain difficult even when many IV exemplars are available. In both classes, OOV success remains positively correlated with IV exemplar count (inflection: $\rho = 0.38$; derivation: $\rho = 0.38$), suggesting that OOV generalization depends on having enough IV pairs, even if this alone does not close the gap.

B.3 Filtering the Vocabulary Decomposition for Failed Surface Form Compositions

In §5 we have seen that, even though the compositional embeddings work well for many in- and out-of-vocabulary words, there are also failure cases where we cannot be certain that the model interprets the compositional representation correctly. Intuitively, this means that using these representations in end-to-end language modeling might hurt model performance; indeed, when we remove the surface forms corresponding to these failures from the decomposition map (and after fine-tuning the

Targets	Exact	Base	Diff. infl.	Other	N
Inflected targets	81.47	15.77	1.96	0.80	9.4k
Base targets	99.98	–	–	0.02	14k

Table 7: Patchscopes outcome rates for English targets. Errors are dominated by collapse to the base form, whereas confusion with a *different* inflection of the same base is rare.

Space	Self sim.	Other sim.	Margin	Top-1 acc.
Input	0.186	0.040	0.146	98.21
Output	0.289	0.108	0.181	95.63

Table 8: Analysis of the separability of initialized *transformation* vectors across the eight English *transformation* types, using individual inflection-base offsets (e.g., *walked-walk*) and their corresponding labels (e.g., *past tense*). Self-transformation cosine similarity exceeds cross-transformation similarity in both input and output spaces, with high top-1 classification accuracy.

transformation vectors as usual), we observe an average 1.6 points improvement across downstream benchmarks, compared to no filtering. We note that for input-only restructuring, we observe no effect, likely because the model has more error-correction opportunities across its layers. We therefore apply this filtering in experiments in §6.

B.4 Decoding Speed

Our compositional language modeling approach introduces some additional complexity into next-token prediction: to compute token scores over the full, extended vocabulary, we map and sum up logit contributions from the *base form* and *transformation* vocabularies (Eq. 2). To validate that this does not introduce meaningful overhead, we let both the baseline and compositional `Llama-3-8B` models generate text in response to prompts from the CNN-DailyMail dataset (Chen et al., 2016), and measure the average number of tokens generated per second.[19] Our approach introduces only a 0.8% drop in decoding speed (39.6 vs. 39.9 tokens/sec).

Still, since our compositional next-token prediction approach occurs in two stages—first deciding on likely candidates for *base forms* and *transformations*—it naturally allows for optimizations like pruning base-form candidates before computing the logits over the full vocabulary (Holtzman et al., 2020), which could further decrease runtime.

[19]We use 50 random prompts and let models generate up to 256 tokens, on an L40S GPU.

B.5 Compositional representation of morphology

For regular embedding models (top panel in Figure 3), we use `Llama2-7B` (Touvron et al., 2023), `Mistral-7B` (Jiang et al., 2023), `OLMo2-7B` (OLMo et al., 2024), `Phi4-14B` (Abdin et al., 2024), `Llama3-8B` (Grattafiori et al., 2024), `Qwen2.5-7B` (Yang et al., 2024), and `Falcon3-7B` (Team, 2024). For tied input-output embedding models (bottom panel), where input and output embeddings share parameters, we analyze `Llama3-3B` (Grattafiori et al., 2024), `Qwen2.5-3B` (Yang et al., 2024), `Phi4-Mini-Instruct-4B` (Abdin et al., 2024), `Aya-Expanse-8B` (Dang et al., 2024), `Gemma2-2B` and `Gemma2-9B` (Team et al., 2024b).

C Experimental Details

C.1 Post-hoc Fine-tuning Details

For fine-tuning, we use a learning rate of $5e-5$, a warmup ratio of 0.03, a weight decay of 0.0, and a sequence length of $m = 256$. We train on $20k$ examples for 1 epoch. We run the post-hoc adaptation experiments on a single L40S GPU, with fine-tuning taking roughly 30 minutes, and inference taking up to 1 hour.

C.2 Pretraining Details

This appendix provides implementation and evaluation details for the pretraining experiment in §7.

Implementation. We follow the default hyperparameters in the `modded-nanoGPT` codebase (Jordan et al., 2024). All pretraining runs use 4 L40S GPUs, except for training on 1B tokens.

Compositional tokenizer and coverage. We start from a 50k-token GPT-2 tokenizer and remove any token that can be expressed as a base form plus *transformations*, including a whitespace-prefix *transformation* to capture pairs like " walking" vs. "walking". Unlike our post-hoc setup, we do not filter out compositions based on Patchscopes interpretation failures (Appendix B.3).

Factorized next-token prediction. In the pretraining setting, each surface word is represented as a base form together with one choice from each transformation group, including a null label when no transformation from that group is active. If

$w_i = (b_i, t_i^{(1)}, \ldots, t_i^{(G)})$, then the model factorizes

$$p(w_i \mid w_{<i}) = p(b_i \mid w_{<i}) \prod_{g=1}^{G} p\Big(t_i^{(g)} \mid b_i, w_{<i}\Big) \tag{5}$$

The model first computes base logits from the final hidden state $\mathbf{h}_i$,

$$\begin{aligned} \boldsymbol{\ell}_i^{\text{base}} &= W_{\text{base}} \mathbf{h}_i + \mathbf{b}_{\text{base}}, \\ p(b_i \mid w_{<i}) &= \text{softmax}(\boldsymbol{\ell}_i^{\text{base}}) \end{aligned} \tag{6}$$

It then predicts each transformation group conditioned on the selected base by passing $\mathbf{h}_i$ together with the chosen base's unembedding vector $\mathbf{u}_{b_i}$ to a transformation head, yielding group-wise logits

$$\begin{aligned} \boldsymbol{\ell}_i^{(g)} &= f_g(\mathbf{h}_i, \mathbf{u}_{b_i}), \\ p\Big(t_i^{(g)} \mid b_i, w_{<i}\Big) &= \text{softmax}(\boldsymbol{\ell}_i^{(g)}) \end{aligned} \tag{7}$$

During training, under teacher forcing, the transformation heads are conditioned on the gold base token. The negative log-likelihood therefore decomposes into a base-prediction term and a sum of transformation-group terms:

$$\begin{aligned} \mathcal{L} = -\sum_i \Bigg[& \log p(b_i \mid w_{<i}) \\ & + \sum_{g=1}^{G} \log p\Big(t_i^{(g)} \mid b_i, w_{<i}\Big) \Bigg] \end{aligned} \tag{8}$$

At inference time, we first sample a base token, and then sample one transformation value from each group (for the experiments in this paper, both sampling operations use argmax), and finally compose them back into the surface realization. In other words, next-token prediction is hierarchical rather than a single softmax over surface forms.

Bits-Per-Byte (BPB). For both baseline and compositional models, we report BPB, computed as the average negative log-likelihood divided by the number of UTF-8 bytes in the evaluation text (lower is better). For the compositional model, we use the teacher-forced joint likelihood under the factorized distribution.

D Downstream Evaluation

We include 5 in-context examples for every task. For each dataset, we use 5,000 examples (or the maximum available as some datasets have fewer available samples).

ARC features 4-option multiple-choice science questions from grades 3 through 9. It has two subsets: ARC-Easy, focused on basic science knowledge, and ARC-Challenge, which involves more complex, procedural reasoning (Clark et al., 2018).

BoolQ comprises naturally occurring yes/no questions accompanied by passages that support the answer (Clark et al., 2019).

COPA offers binary multiple-choice questions centered around causal and consequential reasoning (Kavumba et al., 2019).

HellaSwag includes 4-option multiple-choice questions where the task is to select the most plausible continuation of a given context (Zellers et al., 2019).

MMLU presents 4-option multiple-choice questions across 57 subject areas, testing both factual knowledge and reasoning skills (Hendrycks et al., 2021).

PIQA provides multiple-choice questions designed to evaluate physical commonsense understanding (Bisk et al., 2020).

SQuAD pairs reading passages with related questions, where the correct answer is always a text span from the passage itself (Rajpurkar et al., 2016).

TriviaQA features open-domain questions aimed at assessing general world knowledge (Joshi et al., 2017).

Winogrande contains questions modeled after the Winograd schema but scaled up in size and difficulty (Sakaguchi et al., 2021).

XNLI provides natural language inference examples in 15 languages, where the task is to determine whether a hypothesis is entailed by, contradicts, or is neutral with respect to a given premise (Conneau et al., 2018).

XQuAD is a cross-lingual question answering dataset that pairs reading passages with related questions in 11 languages, where the correct answer is always a text span from the passage itself (Artetxe et al., 2020).

Global MMLU extends the original MMLU benchmark to assess multilingual capabilities, featuring 4-option multiple-choice questions across 57 subject areas in 42 languages including low-resource languages, testing both factual knowledge

	Language	Capitalization		Noun Inflection		Adjective Inflection		Verb Inflection		Derivation	
		In-Vocab.	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*	*In-Vocab.*	*Out-Vocab.*
ALLaM	Arabic	—	—	78% (1.8k)	16% (3.6k)	69% (0.5k)	25% (1.0k)	43% (1.0k)	15% (2.7k)	—	—
EuroLLM	German	100% (0.2k)	89% (0.4k)	—	—	27% (0.3k)	11% (1.3k)	88% (0.3k)	44% (1.2k)	—	—
	Russian	98% (66)	96% (0.7k)	72% (0.6k)	28% (4.2k)	100% (50)	93% (94)	100% (6)	50% (10)	—	—
	Spanish	100% (1.0k)	97% (2.8k)	83% (0.7k)	59% (1.9k)	82% (0.5k)	67% (1.1k)	72% (0.8k)	42% (6.9k)	46% (65)	20% (0.4k)
Llama-3	English	89% (6.0k)	85% (8.4k)	93% (2.4k)	63% (5.6k)	89% (61)	32% (0.9k)	85% (0.9k)	34% (6.4k)	34% (41)	4% (12.8k)

Table 9: Accuracy of Patchscopes *detokenization* interpretations for compositional input embeddings across languages.

Transformation	In-vocab.			Out-of-vocab.		
	embed	*detok*	*N*	*embed*	*detok*	*N*
Inflection						
Plural (N)	92%	92%	0.8k	24%	31%	3.4k
Plural (N) & Present Singular (V)	86%	87%	1.6k	35%	44%	2.1k
Present Singular (V)	91%	91%	0.1k	54%	64%	0.3k
Past (V)	65%	68%	0.6k	10%	15%	2.9k
Past Participle (V)	79%	79%	14	24%	29%	21
Gerund (V)	83%	84%	0.2k	17%	22%	3.2k
Superlative (ADJ)	87%	87%	31	3%	10%	0.4k
Comparative (ADJ)	47%	67%	30	4%	12%	0.4k
Capitalization	72%	73%	6.0k	74%	76%	8.3k
Derivation						
-y	17%	22%	18	2%	6%	1.5k
-er	25%	25%	12	1%	3%	2.6k
-al	62%	62%	8	1%	2%	0.7k
un-	0%	33%	3	0%	1%	3.3k
re-	67%	67%	3	0%	1%	1.8k
-ic	100%	100%	2	5%	7%	0.4k
All derivatives	40%	44%	52	0%	1%	31.4k

Table 10: Accuracy of Patchscopes interpretations for `Qwen-2.5-7B`.

Transformation	In-vocab.			Out-of-vocab.		
	embed	*detok*	*N*	*embed*	*detok*	*N*
Inflection						
Plural (N)	93%	94%	0.8k	34%	42%	3.4k
Plural (N) & Present Singular (V)	86%	90%	1.6k	41%	58%	2.1k
Present Singular (V)	90%	91%	0.1k	60%	71%	0.3k
Past (V)	74%	85%	0.6k	12%	24%	2.9k
Past Participle (V)	100%	100%	14	24%	43%	21
Gerund (V)	93%	97%	0.2k	26%	38%	3.2k
Superlative (ADJ)	97%	97%	31	20%	38%	0.4k
Comparative (ADJ)	87%	90%	30	7%	18%	0.4k
Capitalization	80%	96%	6.0k	50%	85%	8.3k
Derivation						
-y	65%	65%	17	13%	19%	1.5k
-er	25%	33%	12	6%	19%	2.6k
-al	75%	88%	8	4%	11%	0.7k
un-	33%	33%	3	1%	6%	3.3k
re-	100%	100%	3	1%	17%	1.8k
-ic	100%	100%	2	10%	15%	0.4k
All derivatives	63%	67%	51	2%	6%	31.4k

Table 11: Accuracy of Patchscopes interpretations for `OLMo-2-7B`.

and reasoning skills in diverse linguistic contexts (Singh et al., 2025).

E Additional Related Work

Tokenization for morphologically-rich languages Standard BPE tokenization often struggles to capture morphologically complex languages (Klein and Tsarfaty, 2020; Park et al., 2021; Mager et al., 2022; Hofmann et al., 2022). Arabic inflection, for instance, uses non-concatenative morphology that breaks standard subword reusability (Alyafeai et al., 2021; Alrefaie et al., 2024; Tsarfaty et al., 2019; Gazit et al., 2025). Compositional vocabularies can bypass such limitations by representing surface forms as transformations over lexical roots, enabling reuse of base forms even when their surface realizations use diverging token sequences.

Category	**Task**	**Baseline**	**End-to-end**	Δ
Knowledge	MMLU (Acc.)	74.2	74.0	-0.2
	ARC (Acc.)	59.2	57.5	-1.7
Reading Comprehension	BoolQ (Acc.)	87.5	88.0	+0.5
	TriviaQA (EM)	58.3	56.1	-2.2
	SQuAD (EM)	37.3	36.2	-1.1
Commonsense	Hellaswag (Acc.)	59.6	58.4	-1.2
	Winogrande (Acc.)	75.5	75.0	-0.5
	PIQA (Acc.)	79.5	78.8	-0.7
	COPA (Acc.)	91.0	91.0	+0.0
Average		**69.1**	**68.3**	-0.8

Table 12: Downstream performance of English compositional-vocabulary models (*End-to-end*) and their original, unmodified version (*Baseline*) for `Qwen2.5-7B`.

Category	**Task**	**Baseline**	**End-to-end**	Δ
Knowledge	MMLU (Acc.)	62.7	62.2	-0.5
	ARC (Acc.)	60.5	58.4	-2.1
Reading Comprehension	BoolQ (Acc.)	84.4	84.4	+0.0
	TriviaQA (EM)	65.4	61.1	-4.3
	SQuAD (EM)	39.9	36.9	-3.0
Commonsense	Hellaswag (Acc.)	61.1	58.6	-2.5
	Winogrande (Acc.)	77.3	77.2	-0.1
	PIQA (Acc.)	80.2	79.2	-1.0
	COPA (Acc.)	90.0	91.0	+1.0
Average		**69.0**	**67.7**	-2.3

Table 13: Downstream performance for `OLMo-2-7B`.