通过正交匹配追踪的无训练分词器移植

Charles Goddard and Fernando Fernandes Neto

{ charles,fernando } @arcee.ai

Abstract

我们提出了一种免训练的方法，通过使用正交匹配追踪（OMP）重构未见过的词元嵌入，从而在预训练的大型语言模型（LLMs）中移植分词器。具体来说，我们将每个超出词汇范围的词元近似为共享词元的稀疏线性组合，分为两个阶段：首先，使用一个包含共享锚点词元的小字典计算每个新词元在捐赠者嵌入空间中的表示，然后将这些相同的稀疏系数转移回基础模型的嵌入空间。在两个具有挑战性的分词器跨越任务——Llama → 、Mistral NeMo (12B) 和 Qwen → Llama (1B) 上，我们展示了 OMP 在多个基准测试中实现了最佳零样本保留基础模型性能，而其他零样本方法则显著下降。与基线（零初始化、均值初始化，以及现有方法如 WECHSEL、FOCUS、ZETT）相比，OMP 始终实现最佳整体性能，有效弥合了大型分词器差异而无需梯度更新。我们的分析进一步识别出不匹配的数值分词方案是保留数学推理能力的关键挑战。这一技术使得通过新的分词器直接重用预训练模型权重成为可能，促进分词器跨越知识蒸馏、探索性解码、融合、合并和领域特定词汇适应。我们将这种方法集成到开源 `mergokit-tokensurgeon` 工具中，以便进行后期词汇对齐。

1 介绍

大型语言模型（LLMs）通常受到预训练过程中选择的分词器的限制，该分词器定义了用于处理文本的固定词汇。然而，这些词汇并不是通用的；它们可能对其他语言、方言或领域进行次优的分词 [31, 9]。虽然在训练后替换模型的分词器在许多情况下是非常可取的，但在不降低性能的情况下这样做仍然是一个艰巨的技术挑战。

核心难点在于新标记的嵌入初始化。简单的初始化可能显著降低性能 [26, 12]，而继续预训练以学习新的嵌入可能代价过高 [4, 2]。同时，现有的零样本启发式方法效果不均衡，并可能导致在诸如问答或推理等任务上出现较大的性能下降 [9, 16]，特别是当诸如数字标记化方案等基本表征结构在模型之间不同的时候。

这种词汇不匹配在几个重要场景中产生了实际障碍。在推测解码时 [22, 1]，一个小模型提出部分输出供一个较大模型验证，这需要对齐的分词器进行直接对数比较。知识蒸馏 [17] 同样需要匹配的词汇表来在教师和学生间转移令牌级监督 [3]。模型集成或合并 [13] 假设令牌映射到相同的嵌入维度，不同的分词器阻碍了直接输出组合。一种有效的无需重新训练的事后分词器转移方法将解锁这些能力，大大扩展现有模型的应用。

在本文中，我们提出了一种完全无训练的“分词器移植”方法，其中我们使用正交匹配追踪（OMP）在基础模型的嵌入空间中重构新词汇的嵌入。我们的核心见解是将每个新标记的供体嵌入表示为共享标记嵌入的稀疏组合。然后移植变得直截了当：我们在基础模型的嵌入空间中复制相同的稀疏系数。这会生成一个新的词汇嵌入矩阵，与新分词器对齐但与基础模型的嵌入几何保持一致。

我们在两个跨标记器实验中评估了基于 OMP 的移植：Llama → Mistral NeMo (12B) 和 Qwen → Llama (1B)。在这两种情况下，OMP 在分类、推理和困惑度基准上的模型性能保持

得远比简单启发式方法（零、均值）或其他零次初始化方法要好得多。我们将这一策略整合到 `mergekit-tokensurgeon` 开源工具包中，使其能够在跨标记器知识蒸馏、推测解码流水线或领域词汇扩展中立即使用，而无需重新训练。

贡献

1. 我们提出了一种在嵌入空间中使用正交匹配追踪的无训练标记器移植方法。
2. 我们在 Llama → Mistral NeMo 和 Qwen → Llama 任务中展示了优于现有方法的零样本性能。
3. 我们识别并分析了不匹配的数值标记化方案对数学推理性能的显著负面影响，强调了零次移植的一个关键局限性。
4. 我们强调了基于 OMP 的移植如何解决跨标记器使用案例中的关键瓶颈（例如，教师-学生蒸馏、推测解码或领域扩展）。
5. 我们提供了一种高效的增量 QR OMP 实现，并在 `mergekit-tokensurgeon` 中发布以确保可重复性。

2 相关工作

在模型适配中为新的或未见过的标记分配嵌入是一个研究得较为透彻的问题。简单的（随机/零）初始化通常会显著降低性能。更复杂的方法通常通过平均或组合旧词汇表中的子词片段来计算嵌入。例如，Minixhofer et al. [26] 提出了 WECHSEL，使用双语词嵌入将一个语言的子词片段映射到另一个语言，然后继续进行预训练以完善新的嵌入。类似地，Dobler and de Melo [9] (FOCUS) 和 Ostendorff and Rehm [28] (CLPTransfer) 基于重叠标记进行初始化，给予权重以近似新的标记分布。这些方法能够生成更好的初始嵌入，但通常仍然依赖于某种持续训练以恢复原始模型的性能。相比之下，我们的方法无需进一步训练即可获得高质量结果。此外，OMP 生成带符号的系数，因此探索更大的线性子空间，而不是限制在锚点的凸包上。

零样本分词器迁移。 ZeTT [27] 为任意标记器训练了一个超网络，在元训练后可以零样本应用。其他工具 [18, 13] 复制或平均现有的嵌入。我们的工作与这些方法的目标一致，即最少或不需要额外训练，但使用正交匹配追踪 (OMP) 来在现有嵌入的稀疏集中锚定新的标记嵌入。该方法产生更准确的近似，特别是在词汇量差异较大的情况下，并且不需要元训练的超网络。

稀疏近似技术。 正交匹配追踪 (OMP) [30] 是一种用于稀疏重建的标准技术，在信号处理和字典学习中被广泛使用。尽管在自然语言处理领域的使用有限 [24, 32]，但据我们所知，它还没有被用于跨标记器的嵌入转移。我们的方法是首次将 OMP 应用于跨标记器嵌入对齐的解决方案。

3 方法论

3.1 问题设置和符号

设 $\mathcal{M}_{\text{base}}$ 是一个预训练的基础模型，具有词汇表 V_{base} 和嵌入矩阵 $E^{(\text{base})} \in \mathbb{R}^{|V_{\text{base}}| \times d_{\text{base}}}$ 。我们希望将其词汇表替换为来自另一个模型 $\mathcal{M}_{\text{donor}}$ 的 V_{donor} ，该模型具有嵌入矩阵 $E^{(\text{donor})} \in \mathbb{R}^{|V_{\text{donor}}| \times d_{\text{donor}}}$ 。注意， d_{donor} 可能与 d_{base} 不同。我们用 $V_{\cap} = V_{\text{base}} \cap V_{\text{donor}}$ 表示共享（重叠）的标记集合。

Algorithm 1 用于供体空间近似的 OMP

Require: Dictionary $\Phi = [e_j^{(\text{donor})}]_{j \in V_\cap} \in \mathbb{R}^{d \times |V_\cap|}$, target $v = e_t^{(\text{donor})} \in \mathbb{R}^d$, sparsity k .

Ensure: A set of k indices Λ and sparse coeffs $x \in \mathbb{R}^{|V_\cap|}$.

- 1: Initialize $\Lambda \leftarrow \emptyset$, $r \leftarrow v$.
 - 2: **for** $i = 1 \dots k$ **do**
 - 3: $\lambda^* \leftarrow \arg \max_{j \in V_\cap} |\langle r, \phi_j \rangle|$
 - 4: $\Lambda \leftarrow \Lambda \cup \{\lambda^*\}$
 - 5: Solve $x_\Lambda = \arg \min_{x_\Lambda} \|v - \Phi_\Lambda x_\Lambda\|_2^2$
 - 6: $r \leftarrow v - \Phi_\Lambda x_\Lambda$
 - 7: **end for**
 - 8: return (Λ, x) where x is zero outside Λ
-

情况 1: 共享标记。 如果 $t \in V_\cap$, 我们只需将 $e_t^{(\text{base})}$ 复制到新的嵌入矩阵 $E^{(\text{new})}$ 。

当 $t \notin V_{\text{base}}$ 时, 我们有一个捐赠者嵌入 $e_t^{(\text{donor})}$, 但没有对应的 $e_t^{(\text{base})}$ 。我们必须通过参考两个词汇表中都存在的锚定词来近似 $e_t^{(\text{base})}$ 。换句话说:

$$e_t^{(\text{donor})} \approx \sum_{j \in A} \alpha_j e_j^{(\text{donor})}, \quad A \subseteq V_\cap, \quad |A| \leq k,$$

然后我们放置

$$e_t^{(\text{new})} = \sum_{j \in A} \alpha_j e_j^{(\text{base})}.$$

在这里, k 是选择的稀疏级别 (例如, 8、32、64), A 是在捐赠者的嵌入空间中由 OMP 求解器选择的小集合锚定词。通过在基模型的空间中应用这些系数, 我们创建新的嵌入 $e_t^{(\text{new})}$ 。由于仅传递与单位无关的系数向量 α , 因此不依赖于匹配的嵌入维度。

3.2 正交匹配追踪

正交匹配追踪 [30] 是一种贪婪算法, 它通过字典 Φ 中的列来构建目标向量 v 的稀疏表示。在我们的设置中:

$$v = e_t^{(\text{donor})} \in \mathbb{R}^{d_{\text{donor}}}, \quad \Phi = [e_j^{(\text{donor})}]_{j \in V_\cap} \in \mathbb{R}^{d_{\text{donor}} \times |V_\cap|}.$$

我们希望

$$\min_{x \in \mathbb{R}^{|V_\cap|}} \|v - \Phi x\|_2 \quad \text{subject to} \quad \|x\|_0 \leq k.$$

OMP 迭代选择与当前残差最相关的字典列, 更新部分最小二乘解, 并细化残差。经过 k 步骤后, 我们得到一小组锚点索引 Λ 加上非零系数 x_Λ 。算法 1 展示了该过程。

每次迭代求解最小二乘问题可能会很昂贵。我们采用 Φ_Λ 的增量 QR 分解来更有效地更新解决方案。

与其在每次迭代中从头解决最小二乘问题, 我们保留了已选定的 Φ 列的 QR 分解, 并在添加新列时逐步更新给定的 $\Phi_{t-1} = Q_{t-1} R_{t-1}$ 。当我们添加列 ϕ_{λ_t} 时, 我们: (1) 计算 $\hat{q} = \phi_{\lambda_t} - Q_{t-1} Q_{t-1}^T \phi_{\lambda_t}$ (与现有列正交的分量); (2) 归一化以获得 $q_t = \hat{q} / \|\hat{q}\|_2$; (3) 用 q_t 扩展 Q_{t-1} 并据此更新 R_{t-1} 。为了数值稳定性, 我们以固定的间隔再正交化, 但需注意的是 OMP 的原子选择本身很好地避免了灾难性消除——是否再正交化的结果仅在浮点精度上有所不同。

这种方法将每次迭代的复杂度从 $O(td^2)$ 降低到 $O(td)$ ，使得该算法在大量锚点的情况下是实用的。

在获得稀疏系数 x_Λ 后，我们对所有未见过的标记 $t \notin V_{\text{base}}$ 重复此操作，而共享标记 $t \in V_\cap$ 则简单地复制其现有的基础嵌入。最后，我们用 $E^{(\text{new})}$ 替换原始嵌入矩阵 $E^{(\text{base})}$ ，确保模型现在在匹配捐赠者分词器的词汇表和 ID 映射。

3.3 通过近似正交对齐进行几何验证

我们方法的基本原理依赖于嵌入对齐和稀疏恢复的两个原则：

近似正交等价 独立训练的嵌入空间（例如，单词或 LLM 标记 [20, 21]）常常通过一个近似正交变换 $U \in \mathbb{R}^{d_{\text{base}} \times d_{\text{donor}}}$ （其中 $UU^\top \approx I_{d_{\text{base}}}$ ， $U^\top U \approx I_{d_{\text{donor}}}$ ）在其共享子空间上对齐 [7]。因此，对于 $j \in V_\cap$ ， $U e_j^{(\text{donor})} \approx e_j^{(\text{base})}$ ，这意味着 U 应用于捐赠者嵌入的线性组合近似于基空间中相同的组合。

稀疏重建稳定性。 正交匹配追踪 (OMP) [30, 33] 可以可靠地从少量字典原子中重构目标向量，如果字典表现出适度的不相干性或限制等距性质。如果 $e_t^{(\text{donor})} \approx \sum_{j \in A} \alpha_j e_j^{(\text{donor})}$ ，OMP 能以证明的小误差找到 $\{\alpha_j\}$ 。 $\|e_t^{(\text{donor})} - \sum_{j \in A} \alpha_j e_j^{(\text{donor})}\|_2$ 。

将这些结合起来，如果 $e_t^{(\text{donor})} \approx \sum_{j \in A} \alpha_j e_j^{(\text{donor})}$ ，应用 U 会得到：

$$U e_t^{(\text{donor})} \approx \sum_{j \in A} \alpha_j U e_j^{(\text{donor})} \approx \sum_{j \in A} \alpha_j e_j^{(\text{base})} = e_t^{(\text{new})}.$$

因此，将 OMP 派生的系数移植到基础模型的嵌入空间中，隐式地执行了一个最小二乘投影和一个近似正交对齐。

这为我们方法的有效性提供了一个概念上的解释，尽管我们没有证明（或寻求证明）这种对齐普遍存在，但实证结果表明它在实践中成立。

4 实验与结果

我们在两种交叉分词器设置上评估我们基于 OMP 的方法：

- Llama → Mistral NeMo (12B): Llama 3 的 [14] ~ 128k 词汇 transplanted into a 12B Mistral NeMo, which has ~ 131k tokens. 重叠: ~ 71k tokens (~ 54%)。
- Qwen → Llama (1B): Qwen 2.5 的 [34] ~ 152k 词汇移植到一个拥有 ~ 128k 词汇的 1B 参数 Llama 3 模型中。重叠: ~ 110k 标记 (~ 86%)。

在每一种情况下，我们比较：

我们在 WikiText 上评估困惑度，并在 MMLU [15]、ARC [5]、GSM8K [6]、LAMBADA [29]、XNLI [8] 和 Paws-X [35] 等上使用 Eleuther AI 的 LM Evaluation Harness [11] 评估分类或 QA 准确性。对于已发表的基线（如 ZETT、FOCUS、WECHSEL 和 CLPTransfer），我们仅应用了其提议的嵌入初始化技术，省略了原始方法中建议的任何后续微调或继续预训练步骤，以确保零样本初始化质量的直接比较。我们还在第 4.4 节中比较了在相同标记预算下的训练后结果。

表 1 总结了评估设置以及通过引导法 [11] 计算的关键基准的主要指标的典型标准误差 (stderr)。这些误差量化了源于评估数据集样本有限大小的不确定性。由于标准误差相对于不同方法之间观察

到的性能差异始终较小，并且不会影响相对排名或主要结论，为了简明起见，我们在主要结果表（表 2、3、4、5）中省略了它们。

Table 1: 主要基准的评估设置和代表性标准误差 (stderr)。

Benchmark	Metric	Few-shot	StdErr
MMLU	acc (agg.)	0	0.004
ARC Challenge	acc_norm	0	0.014
GSM8K	exact_match (flex)	5	0.013
XNLI	acc (agg.)	0	0.003
AGIEval	acc (agg.)	0	0.005
Lambada (EN)	acc	0	0.006
WikiText	bits_per_byte	0	N/A*

* LM Eval Harness does not report bootstrapped stderr for perplexity metrics.

‘(flex)’ refers to flexible answer extraction; ‘(agg.)’ is aggregated score.

4.1 Llama → Mistral NeMo (12B)

在 Llama → Mistral NeMo 中，基础 NeMo 模型具有较高的 MMLU 准确率 (64.5%)。表 2 显示，OMP 与 $k = 64$ 一起将 MMLU 保持在 62.2% (-3.6%)，而简单基准下降到 $\approx 58\%$ (-9.3%)。虽然所有零次试验移植都显著降低了数学任务的表现，但 OMP 整体上仍然是最稳定的。

Table 2: Llama → Mistral NeMo 结果。相对性能变化显示在括号中。

Model	MMLU	ARC-C	XNLI	GSM8K	AGIEval	Lambada (EN)	WikiText (Bits/Byte)
Baseline (Mistral NeMo)	0.6452	0.5811	0.4367	0.5588	0.3752	0.7819	0.5296
OMP-K8	0.6129 (-5.00%)	0.5179 (-10.87%)	0.3782 (-13.39%)	0.1289 (-76.93%)	0.3137 (-16.38%)	0.6897 (-11.79%)	0.7798 (+47.24%)
OMP-K32	0.6158 (-4.56%)	0.5239 -9.84%	0.3780 (-13.44%)	0.1296 (-76.80%)	0.3146 (-16.15%)	0.6905 (-11.69%)	0.8048 (+51.95%)
OMP-K64	0.6222 (-3.57%)	0.5273 (-9.25%)	0.3780 (-13.43%)	0.1463 (-73.81%)	0.3158 (-15.83%)	0.6895 (-11.81%)	0.7417 (+40.05%)
FOCUS	0.2299 (-64.37%)	0.2585 (-55.51%)	0.3286 (-24.74%)	0.0076 (-98.64%)	0.2447 (-34.78%)	0.0000 (-100.00%)	5.8441 (+1003.47%)
ZeroEmbed	0.5852 (-9.29%)	0.4949 (-14.83%)	0.3689 (-15.53%)	0.0334 -94.03%	0.2905 (-22.57%)	0.6598 (-15.61%)	0.9881 (+86.58%)
MeanEmbed	0.5936 (-8.00%)	0.4966 (-14.54%)	0.3693 (-15.43%)	0.0523 (-90.64%)	0.2952 (-21.31%)	0.6670 (-14.69%)	0.8993 (+69.81%)

4.2 Qwen → Llama (1B) 结果

由于非常高的 token 重叠率 (约 86 %), Qwen → Llama 移植呈现了一个有趣的情形, 这几乎包括了与像 MMLU 和 ARC 这样的基准相关的所有英语 token。表格 3 显示了关键指标。其中, ZeroEmbed 和 MeanEmbed 的弹性表明避免由新 token 初始化干扰这些共享嵌入是一个关键因素。OMP 在这种高重叠场景中巧妙导航。它在 MMLU (36.40 % vs. 36.73 % 基线) 和 ARC (36.26 % vs. 36.26 % 基线) 中保持了接近基线的性能, 并在 Belebele 中表现强劲, 仅在 WikiText 上稍微增加了困惑度 (0.7159 vs. 0.6605 bits/byte)。该性能优于所有更简单的启发式方法。对于 GSM8K, 再次表明所有的零样本方法都严重退化; 而 OMP 仍然具有竞争力。

Table 3: Qwen → Llama (1B) 分词器移植性能。

Model	MMLU	ARC-C	GSM8K	Belebele	WikiText (Bits/Byte) ↓
Baseline (Llama)	0.3673	0.3626	0.0675	0.2813	0.6605
OMP-K8	0.3641 (-0.85 %)	0.3626 (+0.00 %)	0.0144 (-78.65 %)	0.2687 (-4.46 %)	0.7160 (+8.40 %)
OMP-K32	0.3640 (-0.89 %)	0.3626 (+0.00 %)	0.0144 (-78.65 %)	0.2693 (-4.25 %)	0.7160 (+8.39 %)
OMP-K64	0.3640 (-0.89 %)	0.3626 (+0.00 %)	0.0144 (-78.65 %)	0.2704 (-3.86 %)	0.7159 (+8.38 %)
CLPTransfer	0.2295 (-37.52 %)	0.2858 (-21.18 %)	0.0000 (-100.00 %)	0.2299 (-18.26 %)	6791.6044 (+1e6 %)
FOCUS	0.2695 (-26.62 %)	0.3549 (-2.12 %)	0.0174 (-74.16 %)	0.2478 (-11.91 %)	0.8870 (+34.29 %)
WECHSEL	0.2314 (-36.98 %)	0.2389 (-34.12 %)	0.0099 (-85.39 %)	0.2470 (-12.17 %)	3.1852 (+382.24 %)
ZETT	0.2634 (-28.27 %)	0.3055 (-15.76 %)	0.0000 (-100.00 %)	0.2480 (-11.82 %)	1.1426 (+72.99 %)
ZeroEmbed	0.3640 (-0.89 %)	0.3626 (+0.00 %)	0.0144 (-78.65 %)	0.2648 (-5.87 %)	0.7169 (+8.53 %)
MeanEmbed	0.3641 (-0.85 %)	0.3626 (+0.00 %)	0.0144 (-78.65 %)	0.2577 (-8.38 %)	0.7180 (+8.70 %)

我们研究了稀疏参数 k 如何影响基准测试中的性能。我们的实验测试了从 $k = 8$ 到 $k = 256$ 的值。

在 Qwen → Llama 实验中, 我们发现超出 $k = 16$ 只带来极小的收益, 大多数任务在 $k = 32$ 到 64 范围内表现趋于平稳。例如, 对于 $k = 8$ 、 $k = 32$ 和 $k = 256$, MMLU 分数分别为 36.41 %、36.40 % 和 36.41 %。

在 Llama → Mistral NeMo 的情况下, 我们观察到 k 值较高时有更显著的改进, 尤其是在像 GSM8K 这样的数学任务中 (在 $k = 8$ 时为 12.89 % 对比在 $k = 64$ 时为 14.63 %)。然而, 超过 $k = 64$ 后收益递减。

我们建议使用 $k = 64$, 因为它在性能和效率之间提供了最佳平衡。

4.3 数值性能下降。

在 Llama → Mistral NeMo (表 2) 和 Qwen → Llama (表 3) 实验中, 一个引人注目的观察是数学推理性能的显著下降, 尤其是在 GSM8K 上 (对于 OMP-K64, 分别下降了-73.8 % 和-78.7 %, 与基准相比)。我们推测这源于数字标记方案的根本差异。Mistral NeMo 和 Qwen 采用单数字标记化 (例如, “1234” → “1”, “2”, “3”, “4”), 仅产生 10 个基础数字标记。相比之下, Llama 3 使用基于三位数的分块 (例如, “1234” → “123”, “4”), 导致数量庞大的数字词汇表 (1110 个专用数字标记)。

在方案不匹配的模型之间进行转换时 (如 Qwen → Llama 或 Llama → Mistral NeMo), 一模型的大多数数值符号缺乏直接对应关系, 因此必须通过可能的非数字锚点的 OMP 逼近。这很可能在模型学习的数值表示和操作中引入系统性失真, 这些表示和操作可能依赖于与预训练分词方案相关的特定几何结构。例如, 如果模型在一个对数值符号序列敏感的“广义螺旋” [19] 结构上表示数字, 那么通过不匹配方案逼近嵌入 (如通过 OMP 使用 Qwen 的 ‘1’, ‘2’, ‘3’ 锚点重建 Llama 的三联轴 ‘123’) 可能无法保持这种几何安排, 从而削弱算术能力。值得注意的是, 在这些不匹配的场景中, 移植的模型经常对多位数问题给出个位数的答案, 这表明数学推理能力与预训练期间遇到的特定分词模式紧密相关。

为了验证这一假设, 我们进行了额外的实验, 将 Mistral NeMo 的分词器移植到 Qwen 2.5 7B 模型中。至关重要的是, 两个模型都使用相同的单数字符号化方案, 尽管它们的整体词汇表不同。如表 4 所示, 在这一 Mistral NeMo 到 Qwen 的移植过程中, OMP-K64 在 GSM8K 上的性能仅下降了-5.6 % (从 82.6 % 降至 78.0 %)。这与数值方案不匹配时观察到的 >70 % 的降幅形成了鲜明对比。

Table 4: Mistral NeMo → Qwen 的基准测试结果

Model	MMLU	ARC-C	XNLI	GSM8K	AGIEval	Lambda (EN)	WikiText (Bits/Byte) ↓
Baseline (Qwen)	0.7196	0.5137	0.4344	0.8264	0.5639	0.7173	0.5847
OMP-K64	0.7064 (-1.83 %)	0.4872 (-5.15 %)	0.3769 (-13.22 %)	0.7801 (-5.60 %)	0.4423 (-21.56 %)	0.6396 (-10.82 %)	0.6781 (+15.97 %)
MeanEmbed	0.6839 (-4.95 %)	0.4693 (-8.64 %)	0.3560 (-18.04 %)	0.7900 (-4.40 %)	0.4288 (-23.96 %)	0.6309 (-12.04 %)	0.7524 (+28.69 %)
ZeroEmbed	0.6874 (-4.46 %)	0.4804 (-6.48 %)	0.3542 (-18.47 %)	0.7779 (-5.87 %)	0.4278 (-24.13 %)	0.6299 (-12.18 %)	0.7722 (+32.07 %)
CLPTransfer	0.2297 (-68.08 %)	0.2474 (-51.83 %)	0.3326 (-23.44 %)	0.0038 (-99.54 %)	0.2456 (-56.44 %)	0.0000 (-100.00 %)	5.8987 (+908.82 %)

这一结果强烈支持了我们的假设: 当数值标记化方案对齐时, OMP 成功地在更大程度上保留了数学性能。尽管 OMP 能有效地连接一般文本标记的语义表示, 但数值表示的结构差异带来了独特的挑战, 这显著影响了算术和定量推理, 除非基础的数值标记化策略被保留或极为相似。在移植过程中可能需要对数值标记进行专门处理, 以充分保留任意标记器对间的数学能力。

4.4 持续预训练

虽然我们的方法不需要额外的训练, 但我们注意到, 部分微调或领域适应可以进一步提高性能。我们的实验证实, 即使是少量的持续训练也有助于在敏感任务 (如 GSM8K) 上恢复性能, 但零样本

OMP 本身已经优于其他零样本启发式方法。

这些模型在 DCLM [23] 数据集的一个包含二十亿标记的子集上微调一个周期。所有基准模型都使用 AdamW 优化器和 $5e-6$ 的学习率进行调整。OMP 需要一个更低的学习率为 $4e-7$ 。这可能表明 OMP 初始化的嵌入已经定位良好并对较大的更新敏感。本实验的结果显示在表格 5 中。我们注意到 OMP 在 XNLI 上的意外轻微降级在 CPT 之后。尽管确切原因尚不清楚，考虑到 XNLI 的跨语言特性和仅包含英文的 CPT 数据集，这可能表明 CPT 微妙地扰乱了针对该特定任务的初始零次调整，潜在原因可能是由于 OMP 初始化嵌入的敏感性（这可以通过所需的较低学习率得到证明）。

Table 5: Qwen $\rightarrow$ Llama 对与未经和经过在 DCLM 数据集上持续预训练的标记器移植方法的性能比较。灰色部分的百分比显示了与基线模型相比的相对性能变化。对于 WikiText (位/字节)，值越低越好；对于所有其他指标，值越高越好。

Model	MMLU	XNLI	AGIEval	LAMBADA (EN)	Belebele	WikiText (Bits/Byte) $\downarrow$
Baseline	0.3673	0.4086	0.2690	0.6204	0.2813	0.6605
Zero-Shot						
ZETT	0.2634 (-28.3 %)	0.3437 (-15.9 %)	0.2487 (-7.6 %)	0.3460 (-44.2 %)	0.2480 (-11.8 %)	1.1426 (+73.0 %)
FOCUS	0.2695 (-26.6 %)	0.3578 (-12.4 %)	0.2577 (-4.2 %)	0.5936 (-4.3 %)	0.2478 (-11.9 %)	0.8870 (+34.3 %)
OMP-K64	0.3640 (-0.9 %)	0.3430 (-16.1 %)	0.2605 (-3.1 %)	0.6200 (-0.1 %)	0.2704 (-3.9 %)	0.7159 (+8.4 %)
With Continued Pre-Training						
ZETT-DCLM-2B	0.3151 (-14.2 %)	0.3784 (-7.4 %)	0.2638 (-1.9 %)	0.6115 (-1.4 %)	0.2725 (-3.1 %)	0.6819 (+3.2 %)
FOCUS-DCLM-2B	0.3444 (-6.2 %)	0.3750 (-8.2 %)	0.2579 (-4.1 %)	0.6198 (-0.1 %)	0.2559 (-9.0 %)	0.6733 (+1.9 %)
OMP-K64-DCLM-2B	0.3725 (+1.4 %)	0.3388 (-17.1 %)	0.2604 (-3.2 %)	0.6185 (-0.3 %)	0.2724 (-3.2 %)	0.6730 (+1.9 %)

4.5 计算效率。

除了性能指标之外，分词器移植的实际效用还依赖于其计算成本。我们测量了 Qwen $\rightarrow$ Llama (1B) 任务的大致执行时间，该任务涉及大约 $\sim 41,000$ 个标记。OMP 证明效率极高：在单个 H100 GPU 上，仅需 38 秒 (有 $k = 8$) 和 74 秒 (有 $k = 32$)。像 ZeroEmbed 和 MeanEmbed 这样的简单方法几乎没有计算成本 (在 CPU 上几乎是瞬间的)。相比之下，其他一些复杂的零样本方法要求更高：CLPTransfer 在 CPU 上大约需要 9 小时，而 FOCUS 花费的时间相近 (具体时间没有记录，但观察到是相似的)。ZeTT 涉及一次性的大规模元训练成本 (在我们的设置中，在 8x H100 GPU 节点上需 39 小时)，随后是快速的每模型应用步骤 (在单个 H100 GPU 上执行此任务仅需 2 分钟)。这种比较突出了 OMP 的优势，作为一种真正事后、轻量级的解决方案，可以在无需大量预计算或运行时的情况下实现快速部署。

5 讨论

为什么 OMP 效果很好？ 词嵌入空间表现出近似的局部线性和类比结构 [25]，而大语言模型嵌入在高维中形成紧密的语义聚类 [10]。OMP 利用这两点事实：通过贪婪地选择共享锚嵌入的签名， k -稀疏编码，它在不进行任何梯度更新的情况下捕捉到未见标记的主导语义方向。

应用

1. 跨分词器知识蒸馏。教师和学生常常在词汇上存在差异；通过将教师的分词器移植到学生身上，我们可以对匹配的词元 ID 直接应用交叉熵或基于 logit 的蒸馏。
2. 推测性解码。[22] 较小的“草稿”模型必须与较大的“验证器”模型共享词汇表。基于 OMP 的移植允许在不考虑原始词汇表的情况下使用任意模型对。
3. 领域词汇扩展。OMT 可以被用于为新的领域专用词语（例如，医学、化学、代码或多语言）初始化嵌入，而不是完全替换，通过从现有锚点重建它们。这增加了最小的开销并保留了现有的性能。

局限性和未来工作。 我们的方法依赖于在 V_n 中至少有一些重叠；如果不存在重叠，则该方法无法应用——不过由于任何现代字节级或完整的 Unicode 分词器都会为每个码点序列分配一些 ID，在实践中 $|V_n| > 0$ 。结构上不同的数字分词方案（如数字聚类或从右到左与从左到右的分段）可能会显著降低数学任务的效果。在我们发现这一关键限制的基础上，未来工作的一个关键方向是开发混合移植策略。这种策略将利用 OMP 处理一般词汇，同时采用专门为弥合这些结构性表示差距而设计的数字标记处理。其他途径包括为几乎不相交的词汇开发弥合策略或应用其他稀疏编码技术。

我们工作的主要动机是提高预训练语言模型在知识蒸馏、有效的投机解码和特定领域适应等有益应用中的灵活性和可重用性。通过实现无需训练的分词器移植，我们旨在降低技术壁垒，促进这些领域的创新。

然而，与任何提高模型修改和互操作性便捷性的技术一样，也存在潜在的负面社会影响需要考虑。虽然我们的方法本身并不会在模型中创造新的恶意能力，但它可能会无意中降低将现有模型用于非预期或有害目的的技术或计算障碍。例如：

- 促进有问题模型的适应：如果一个基础模型具备可能被滥用的能力（例如，生成复杂的虚假信息、表现出强烈的偏见或产生不安全的内容），我们的技术可能会使恶意行为者更高效地将这种模型适应新词汇或将其集成到有害的流程中。无训练的方法降低了通常与此类重新分词工作相关的成本和专业性要求。
- 偏差和伤害的传播：移植过程直接转移了学习的表示。如果基础模型包含未被缓解的偏差或安全缺陷，当其词汇被修改时，这些偏差或缺陷可能会无缝传播，如果不进行仔细评估，可能会影响移植的分词器所针对的新语言或领域。

必须强调，部署 LLM 相关的伦理责任仍由开发者和用户承担。我们的工具是一个在现有模型上运行的组件，责任在于用户，以确保基础模型被负责任地使用，并且任何通过标记器移植产生的模型在部署之前都得到了彻底的安全性、公平性和其预期应用的评估。我们提倡继续研究针对所有语言模型的强大评估技术和保护措施，无论其标记化方案如何。

6 结论

我们引入了一种无训练的分词器移植方法，利用在共享标记嵌入上的正交匹配追踪。我们的实验表明，OMP 能比简单的启发式方法更好地保持困惑度和分类准确性，从而在新分词器下方便地重用预训练的 LLM 权重。此方法通过消除词汇不匹配而无须额外训练，解锁了关键应用（跨分词器蒸

馏、推测、领域扩展)。我们的实现已在 `mergekit-tokensurgeon` 中公开可用，邀请更广泛的采用和未来的增强。通过强调嵌入空间中稀疏近似的力量，我们希望激发对预训练 LLM 的进一步模块化、后期增强。

References

- [1] Zachary Ankner, Rishab Parthasarathy, Aniruddha Nrusimha, Christopher Rinard, Jonathan Ragan-Kelley, and William Brandon. Hydra: Sequentially-dependent draft heads for medusa decoding. arXiv preprint arXiv:2402.05109 , 2024.
- [2] Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. On the cross-lingual transferability of monolingual representations. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics , pages 4623–4637, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.421. URL <https://aclanthology.org/2020.acl-main.421/>.
- [3] Nicolas Boizard, Kevin El Haddad, Céline Hudelot, and Pierre Colombo. Towards cross-tokenizer distillation: the universal logit distillation loss for llms. arXiv preprint arXiv:2402.12030 , 2024.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. Advances in neural information processing systems , 33:1877–1901, 2020.
- [5] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457 , 2018.
- [6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168 , 2021.
- [7] Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data, 2018. URL <https://arxiv.org/abs/1710.04087>.
- [8] Alexis Conneau, Guillaume Lample, Ruty Rinott, Adina Williams, Samuel R Bowman, Holger Schwenk, and Veselin Stoyanov. Xnli: Evaluating cross-lingual sentence representations. arXiv preprint arXiv:1809.05053 , 2018.
- [9] Konstantin Dobler and Gerard de Melo. FOCUS: Effective embedding initialization for monolingual specialization of multilingual models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing , pages 13440–13454, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.829. URL <https://aclanthology.org/2023.emnlp-main.829/>.
- [10] Kawin Ethayarajh. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In Kentaro Inui, Jing Jiang, Vincent

- Ng, and Xiaojun Wan, editors, Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) , pages 55–65, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1006. URL <https://aclanthology.org/D19-1006/>.
- [11] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 12 2023. URL <https://zenodo.org/records/10256836>.
- [12] Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torrioni. Fast vocabulary transfer for language model compression. In Yunyao Li and Angeliki Lazaridou, editors, Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track , pages 409–416, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-industry.41. URL <https://aclanthology.org/2022.emnlp-industry.41/>.
- [13] Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. Arcee’s MergeKit: A toolkit for merging large language models. In Franck Dernoncourt, Daniel Preoțiuc-Pietro, and Anastasia Shimorina, editors, Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track , pages 477–485, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-industry.36. URL <https://aclanthology.org/2024.emnlp-industry.36/>.
- [14] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783 , 2024.
- [15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300 , 2020.
- [16] John Hewitt. Initializing new word embeddings for pretrained language models, 2021. URL <https://nlp.stanford.edu/~johnhew//vocab-expansion.html>.
- [17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 , 2015.
- [18] jukofyork. Vocab transplantation tool (github repository), 2025. URL <https://github.com/jukofyork/transplant-vocab>.
- [19] Subhash Kantamneni and Max Tegmark. Language models use trigonometry to do addition. arXiv preprint arXiv:2502.00873 , 2025.

- [20] Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching-Yun Chang. Cross-lingual alignment methods for multilingual BERT: A comparative study. In Trevor Cohn, Yulan He, and Yang Liu, editors, Findings of the Association for Computational Linguistics: EMNLP 2020 , pages 933–942, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.83. URL <https://aclanthology.org/2020.findings-emnlp.83/>.
- [21] Andrew Lee, Melanie Weber, Fernanda Viégas, and Martin Wattenberg. Shared global and local geometry of language model embeddings, 2025. URL <https://arxiv.org/abs/2503.21073>.
- [22] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning , pages 19274–19286. PMLR, 2023.
- [23] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruva Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models. arXiv preprint arXiv:2406.11794 , 2024.
- [24] Remya R. K. Menon, S Gargi, and S Samili. Clustering of words using dictionary-learned word representations. In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI) , pages 1539–1545, 2016. doi: 10.1109/ICACCI.2016.7732267.
- [25] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 , 2013.
- [26] Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies , pages 3992–4006, Seattle, United States, July 2022. Association for Computational Linguistics. URL <https://aclanthology.org/2022.naacl-main.293>.
- [27] Benjamin Minixhofer, Edoardo Maria Ponti, and Ivan Vuli. Zero-shot tokenizer transfer, 2024. URL <https://arxiv.org/abs/2405.07883>.
- [28] Malte Ostendorff and Georg Rehm. Efficient language model training through cross-lingual and progressive transfer learning, 2023.

- [29] Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. arXiv preprint arXiv:1606.06031 , 2016.
- [30] Yagyensh Chandra Pati, Ramin Rezaifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In Proceedings of 27th Asilomar conference on signals, systems and computers , pages 40–44. IEEE, 1993.
- [31] Phillip Rust, Jonas Pfeiffer, Ivan Vuli, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. pages 3118–3135, 01 2021. doi: 10.18653/v1/2021.acl-long.243.
- [32] Konstantinos Skianis, Nikolaos Tziortziotis, and Michalis Vazirgiannis. Orthogonal matching pursuit for text classification. arXiv preprint arXiv:1807.04715 , 2018.
- [33] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. IEEE Transactions on information theory , 53(12):4655–4666, 2007.
- [34] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 , 2024.
- [35] Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. Paws-x: A cross-lingual adversarial dataset for paraphrase identification. arXiv preprint arXiv:1908.11828 , 2019.

A 附录

A.1 标记分解示例

这些例子展示了在嵌入空间中，符号如何被表示为其他符号的线性组合。

Token	Sparse Linear Decomposition
UTF8gbsn 读者	$\approx$ ‘readers’ $\times$ 0.147 + ‘reader’ $\times$ 0.129 + ‘UTF8gbsn 作者’ $\times$ 0.126 + ‘UTF8mj 閱’ $\times$ 0.139 + ‘UTF8gbsn 学生’ $\times$ 0.111 + ‘retail’ $\times$ 0.112 + ‘visually’ $\times$ 0.107 + ‘UTF8mj 讀’ $\times$ 0.149
UTF8mj 閱 閱 閱	$\approx$ ‘UTF8mj 閱 閱 閱’ $\times$ 0.142 + ‘UTF8mj 閱 閱’ $\times$ 0.122 + ‘UTF8mj 閱 閱’ $\times$ 0.101 + ‘UTF8gbsn らず’ $\times$ 0.094 + ‘Despite’ $\times$ 0.090 + ‘UTF8gbsn のに’ $\times$ 0.106 + ‘ ’ $\times$ 0.090 + ‘-D’ $\times$ -0.073
UTF8gbsn 它是	$\approx$ ‘UTF8gbsn 它’ $\times$ 0.245 + ‘UTF8gbsn 是’ $\times$ 0.126 + ‘UTF8gbsn 这是’ $\times$ 0.154 + ‘{ } :’ $\times$ 0.111 + ‘Exist’ $\times$ 0.106 + ‘metro’ $\times$ 0.108 + ‘cht’ $\times$ -0.095 + ‘UTF8gbsn 地说’ $\times$ 0.091
‘várias’	$\approx$ ‘varias’ $\times$ 0.208 + ‘diversas’ $\times$ 0.151 + ‘muit’ $\times$ 0.125 + ‘varios’ $\times$ 0.161 + ‘uma’ $\times$ 0.089 + ‘various’ $\times$ 0.090 + ‘ários’ $\times$ 0.097 + ‘Brief’ $\times$ -0.077
UTF8gbsn 消費者	$\approx$ ‘consumers’ $\times$ 0.122 + ‘UTF8gbsn 市場’ $\times$ 0.099 + ‘UTF8mj 閱 閱’ $\times$ 0.113 + ‘UTF8gbsn 企業’ $\times$ 0.101 + ‘Consumers’ $\times$ 0.112 + ‘UTF8gbsn 篇’ $\times$ -0.072 + ‘visitors’ $\times$ 0.080 + ‘portions’ $\times$ -0.078
UTF8gbsn 营利	$\approx$ ‘-profit’ $\times$ 0.141 + ‘UTF8gbsn 莉’ $\times$ 0.124 + ‘ylim’ $\times$ 0.127 + ‘lucrative’ $\times$ 0.130 + ‘Windows’ $\times$ 0.097 + ‘purpos’ $\times$ 0.147 + ‘UTF8gbsn 营’ $\times$ 0.095 + ‘-selling’ $\times$ 0.106
UTF8gbsn 专家	$\approx$ ‘experts’ $\times$ 0.166 + ‘expert’ $\times$ 0.163 + ‘UTF8gbsn 教师’ $\times$ 0.119 + ‘UTF8gbsn 主任’ $\times$ 0.118 + ‘UTF8gbsn 政策’ $\times$ 0.095 + ‘Scientists’ $\times$ 0.125 + ‘UTF8gbsn 咨询’ $\times$ 0.105 + ‘UTF8gbsn 资产’ $\times$ 0.098
UTF8gbsn 必要があり ます	$\approx$ ‘UTF8gbsn できます’ $\times$ 0.197 + ‘UTF8mj 閱 閱 閱 閱’ $\times$ 0.164 + ‘ ’ $\times$ 0.114 + ‘harus’ $\times$ 0.092 + ‘UTF8gbsn べき’ $\times$ 0.126 + ‘However’ $\times$ 0.086 + ‘dispositivo’ $\times$ 0.115 + ‘muss’ $\times$ 0.092
UTF8gbsn 氟	$\approx$ ‘fluoride’ $\times$ 0.123 + ‘UTF8gbsn 弗’ $\times$ 0.150 + ‘lithium’ $\times$ 0.141 + ‘UTF8gbsn 氧’ $\times$ 0.112 + ‘aa’ $\times$ 0.139 + ‘ann’ $\times$ -0.103 + ‘fluor’ $\times$ 0.134 + ‘dynam’ $\times$ -0.098
‘geführt’	$\approx$ ‘gemacht’ $\times$ 0.161 + ‘führt’ $\times$ 0.202 + ‘UTF8mj 閱 閱’ $\times$ 0.097 + ‘geben’ $\times$ 0.088 + ‘UTF8gbsn であり’ $\times$ 0.076 + ‘chnitt’ $\times$ 0.089 + ‘iliated’ $\times$ 0.077 + ‘Sch’ $\times$ -0.061

Figure 1: Qwen 2.5 词汇中选定标记的稀疏线性分解。每个标记被分解为 $k = 8$ 个基础标记的加权和，系数根据大小着色（绿色为正，红色为负）。

A.2 使用的资产和软件

表 6 详细列出了本研究中使用的关键模型、数据集（除文中引用的标准基准测试外）和软件库，以及它们的来源和许可证，以帮助重现。标准评估基准测试如 MMLU、ARC、GSM8K 等，在正文

中被引用，并通过 LM Evaluation Harness 访问。

Table 6: 关键资产和使用的软件概述。

Asset Name	Creator / Origin	Version / Identifier	Source / Access	License
Models				
Llama 3.2 1B	Meta AI [14]	Sept. 2024 release	https://huggingface.co/meta-llama/Llama-3.2-1B	Llama 3 Community License
Qwen 2.5 7B	Alibaba [34]	Sept. 2024 release	https://huggingface.co/Qwen/Qwen2.5-7B	Apache 2.0
Mistral 12B	NeMo Mistral AI	July 2024 release	https://huggingface.co/mistralai/Mistral-Nemo-Base-2407	Apache 2.0
Software				
<code>transformers</code>	Hugging Face	4.50.0	<code>pip install transformers==4.50.0</code>	Apache 2.0
LM Eval Harness	EleutherAI [11]	0.4.8	<code>pip install lm-eval-harness==0.4.8</code>	MIT License
<code>mergekit</code>	Arcee AI [13]	Hash: 4da40d2	https://github.com/arcee-ai/mergekit	BuSL-1.0
<code>axolotl</code>	Axolotl AI	0.8.1	<code>pip install axolotl==0.8.1</code>	Apache 2.0