Understanding Subword Compositionality of Large Language Models

Qiwei Peng♡**, Yekun Chai**♠**, Anders Søgaard**♡
♡University of Copenhagen ♠ETH Zurich
{qipe,soegaard}@di.ku.dk yechai@ethz.ch

Abstract

Large language models (LLMs) take sequences of subwords as input, requiring them to effective compose subword representations into meaningful word-level representations. In this paper, we present a comprehensive set of experiments to probe how LLMs compose subword information, focusing on three key aspects: structural similarity, semantic decomposability, and form retention. Our analysis of the experiments suggests that five LLM families can be classified into three distinct groups, likely reflecting difference in their underlying composition strategies. Specifically, we observe (i) three distinct patterns in the evolution of structural similarity between subword compositions and whole-word representations across layers; (ii) great performance when probing layer by layer their sensitivity to semantic decompositionality; and (iii) three distinct patterns when probing sensitivity to formal features, e.g., character sequence length. These findings provide valuable insights into the compositional dynamics of LLMs and highlight different compositional pattens in how LLMs encode and integrate subword information.

1 Introduction

Large language models (LLMs) rely heavily on subword tokenization (Achiam et al., 2023; Dubey et al., 2024) that processes words into a sequence of subwords which potentially disrupts morpheme boundaries (Batsuren et al., 2024). Despite this, LLMs have demonstrated impressive capability in comprehending word meanings (Shani et al., 2023; Xu et al., 2024), suggesting that they effectively construct meaningful word representations from subword components. One possible approach to this is memorization, where models store entire input-output pairs. This strategy, adopted by Ned Block's *humongous table program* (Block, 1981), scales only if all input-output pairs have been seen during training. However, this is computationally infeasible due to the exponential growth in possible combinations with increasing input length and vocabulary size. Given their promising ability on word meaning understanding, LLMs must be employing systematic compositional strategies rather than relying solely on memorization to generalize beyond seen data. This motivates our investigation into how LLMs construct word representations from subword components and uncover potential consistent and systematic patterns in subword composition.

To systematically examine these compositional strategies, we analyze subword composition from three key perspectives. First, we examine how the geometry of composed word representations relates to that of their subword constituents. Specifically, we assess whether composed representations maintain *linear alignment* with their constituent representations, revealing patterns of structural similarity across layers. Prior studies have explored geometry properties of word and phrase embeddings they construct (Gong et al., 2017), and examined distances between composed subwords and full-word embeddings in vector space (Chai et al., 2024a). Our focus here is to identify linear alignment patterns that reveal structural similarity and transformation dynamics between composed representations and whole-word representations across layers.

Second, we probe whether composed representations encode fundamental aspects of word meaning, particularly the distinction between semantically decomposable and non-decomposable words. Building on previous work that assessed embeddings for their awareness of syntactic and semantic properties, such as sentence length, tense, and identification of semantic roles (Conneau et al., 2018; Ettinger et al., 2018; Klafka and Ettinger, 2020), our analysis focuses on whether LLMs preserve relevant information on semantic decompositionality during composition. Third, we investigate whether

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 22524–22535
November 4-9, 2025 ©2025 Association for Computational Linguistics

composed representations retain surface-level features, such as word length across models and layers. While some models exhibit strong retention of such features, others abstract away form-related information, which shows variations in how form and content are preserved. By analyzing LLMs across these dimensions, our study provides valuable insights into how LLMs process subwords and form word-level representations, contributing to a broader understanding of compositional dynamics in LLMs.

Contributions In this work, we present a set of new experiments designed to probe the compositional dynamics of LLMs around subwords. Our experiments with six different LLMs across five LLM families on three types of tasks demonstrate that: (i) In most models, subword composition is **isometric to simple addition**. (ii) Content information such as semantic decompositionality is well-preserved in the composed representation for all models across all layers. Formal information about word length, in contrast, is only preserved in some models. This has direct implications for the **derivability of form and content** of the input. (iii) The six LLMs fall into three groups, relying on **three distinct compositional strategies**, i.e., ways of constructing composed representations from subwords.

2 Related Work

Tokenization Current generations of LLMs (Achiam et al., 2023; Touvron et al., 2023; Team et al., 2023; Lozhkov et al., 2024), heavily rely on subword tokenization where an input text is split into a sequence of subwords derived from a predefined vocabulary. Such approaches include frequency-based methods such as Byte-Pair Encoding (Sennrich et al., 2016) and Byte-level BPE (Wang et al., 2020), probability-based methods such as WordPiece (Schuster and Nakajima, 2012) and Unigram (Kudo, 2018). Tokenization approaches need to balance the trade-off between vocabulary size and diverse language coverage in multilingual scenarios. Tokenization-free or pixel-based approaches have been proposed to side-step this trade-off (Rust et al., 2023; Tai et al., 2024; Chai et al., 2024b), and various tasks have been proposed to better examine the impact and robustness of subword tokenization (Gee et al., 2022; Cao et al., 2023; Chai et al., 2024a; Wang et al., 2024a; Batsuren et al., 2024). Our work aims to understand subword compositionality in LLMs.

Compositionality The compositional ability allows models to generalize beyond simple memorization. Previous works have thoroughly examined compositionality in phrase (Yu and Ettinger, 2020; Bertolini et al., 2021) and sentence embeddings (Dasgupta et al., 2018; Xu et al., 2023). Recent studies have also explored general compositional behaviors of LLMs in reasoning tasks (Dziri et al., 2024; Li et al., 2024b) and rule following (Wang et al., 2024b). Our work sets out to investigate whether subwords, as a result of tokenization, exhibit any compositional dynamics through geometry and probing analysis. Procrustes analysis, which is a form of statistical shape analysis (Schönemann, 1966), is widely used to analyze structural similarity between two language spaces (Peng and Søgaard, 2024) and modality spaces (Li et al., 2024a). Additionally, probing analysis is a standard approach for dissecting syntactic and semantic features in neural models, such as syntactic depth, tense, and semantic roles (Ettinger et al., 2018; Conneau et al., 2018; Hewitt and Manning, 2019; Klafka and Ettinger, 2020).

3 Geometry Analysis

We first conduct geometry analysis on the internal vector space of different LLM, focusing on the structural similarity between composed representations and the original whole word representation.

3.1 Dataset

Batsuren et al. (2022) proposed a benchmark on morpheme segmentation which collected more than 577,374 unique English words with its morphological categories. We take advantage of this resource and pick out words that have both its whole word form and potential subwords in the model's vocabulary. In this work, we specifically focus on two-subword combination (e.g., limit ⇒ (li, mit)[1]). After going through six different language models, we end up with a parallel[2] dataset across these language models. In total, we have 3,432 words covering 2,316 root words (words that are free morphemes, such as *dog* and *progress*) and 1,116 non-root words (words that fall into other morphological categories such as inflection only, e.g., *prepared*, derivation only, e.g., *intensive*, and compound, e.g.,

[1] *limit*, *li*, and *mit* are all in model's vocabulary.

[2] It is parallel in the whole word form, while the tokenized results might be different.

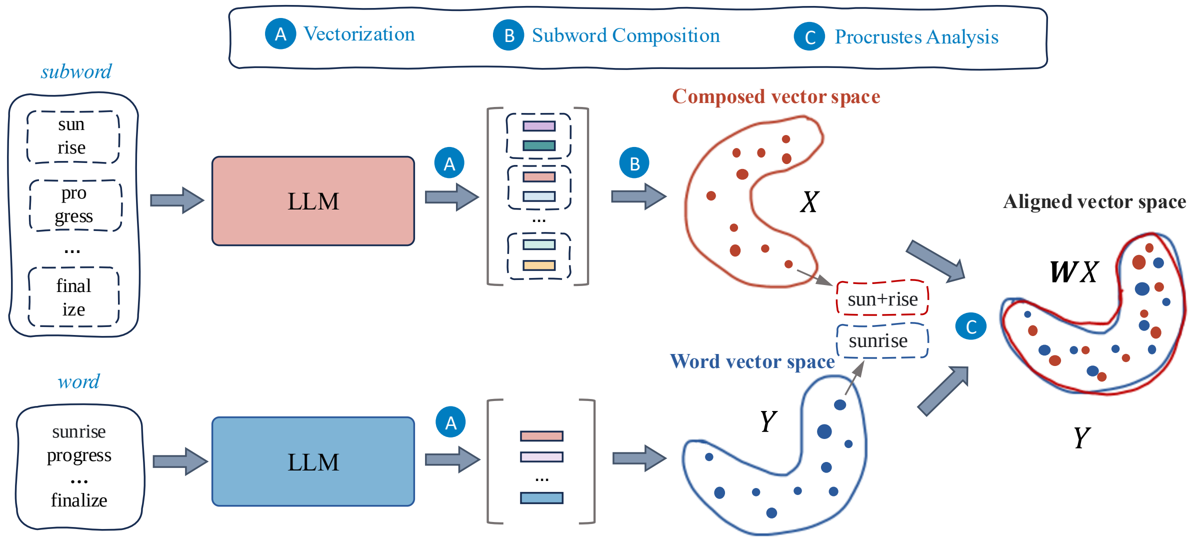


Figure 1: Illustration of the pipeline of our geometry analysis. All words and subwords exist in models' vocabulary. Vector representations are first obtained by feeding them into LLMs. Composed vector space is then constructed by applying composition operations among subword representations. Procrustes analysis is performed between the original word vector space and the composed vector space to find the linear alignment.

hotpot). As one word could have multiple subword combinations discovered (e.g., numeric ⇒ (n, umeric), (num, eric), (numer, ic)), we have all combinations included. In the following experiments, we conduct 3 runs where each run with randomly picked combination to reflect the variation. The ribbons in the experiment figures demonstrate standard deviations brought in by such variation. We randomly split these words into train, test splits.

	Root	Non-Root	Total
Train	1852	893	2745
Test	464	223	687
Total	2316	1116	3432

Table 1: The statistics of the dataset.

LLMs and Vector Representation The six instruction-tuned LLMs we experiment include Llama3-8B-Instruct, Llama3.1-8B-Instruct (Dubey et al., 2024), Aya-expanse-8B (Dang et al., 2024), Gemma2-9B-it (Team, 2024a), Qwen2.5-7B-Instruct (Team, 2024b), and Falcon-7B-Instruct (Almazrouei et al., 2023a). All above models adopt subword tokenization strategy and are instruction-tuned. The whole word vector representation is derived through feeding the exact word to the model. Subword representations are obtained separately through the same pipeline. As all words and subwords exist in models' vocabulary, we can directly obtain their vector representations without additional operations. Different composition operations are then performed on subword representations to obtain the composed representation, which will later be compared against the original whole-word representation to examine structural similarity.

3.2 Methods

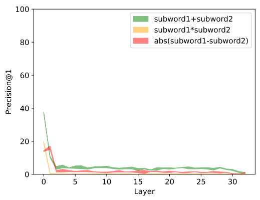
We utilize Procrustes Analysis (Schönemann, 1966), i.e., the induction of a linear projection between two subspaces, to quantify the *isometry* or structural similarity between whole word representations and composed representations of subwords. Assume X and Y are two matrices of size $n \times d$ (n is the number of examples, and d refers to the embedding dimension). Such that the i-th row of X is the composed embedding of two subwords, and ith row of Y is the original embedding of the whole word. The linear transformation is derived through singular value decomposition (SVD) of YX^T:

$$W^* = \underset{W \in O_{d(\mathbb{R})}}{\arg\min} ||WX - Y||_F = UV^T \quad (1)$$

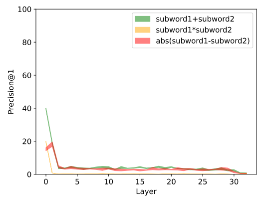
where $U\Sigma V^T = \text{SVD}(YX^T)$. With the obtained W^*, we transform composed embeddings X into the original vector space. We then perform cosine similarity to retrieve the most similar original word vector. Following previous works on measuring representation alignment (Li et al., 2024a; Wu et al., 2024), we use Precision@1 (P@1) as our performance metric. The overall pipeline of the method is illustrated in Figure 1. The train split is used to find the optimal linear transformation W^* which will then be applied to the test split for evaluation.

3.3 Results

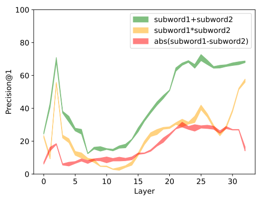
Main Geometry Results Our first experiment simply evaluates the structural similarity of LLM whole word representations and addition of, multiplication of, and absolute difference between constituent representations, by measuring their performance (P@1) across layers.



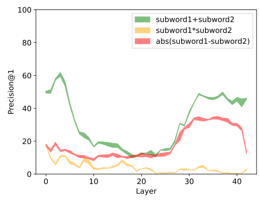
(a) Llama3



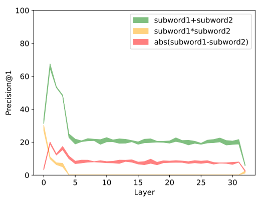
(b) Llama3.1



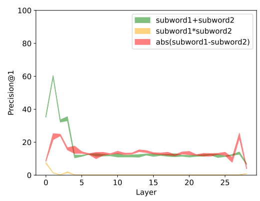
(c) Aya-expanse



(d) Gemma2



(e) Falcon

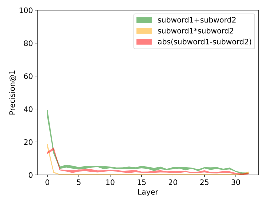


(f) Qwen2.5

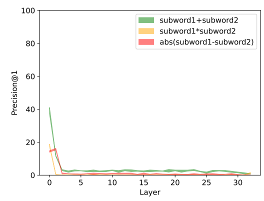
Figure 2: Structural similarity between LLM composition and simple composition (P@1). Green band is simple addition. Orange refers to multiplication. Red is the performance of absolute difference. The colored bands indicate standard deviation. LLM composition is significantly more similar to simple addition.

A key takeaway from Figure 2 is that simple addition consistently outperforms other operations across all models and layers. This suggests that summing two subword representations produces a composed representation with strong structural similarity to the original whole-word representation. However, the degree of similarity varies across models, revealing three distinct patterns. Aya-expanse and Gemma2 exhibit the most impressive P@1 score, indicating high-level structural similarity between composed vectors and the original vectors. Unlike other models, the demonstrated structural similarity is able to maintain across later layers. The high precision in linear alignment exhibited in early layers of Falcon and Qwen2.5 drops in later layers. Llama models, on the other hand, only demonstrate moderate level of structural similarity between composed vectors and word vectors at the embedding layer. The structural similarity drops almost immediately.

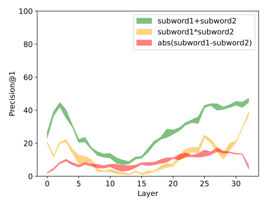
It is easy to see how the six LLMs can be placed in three groups with very distinct plots: Llama 3 and Llama 3.1 show very little structural similarity, and only at the embedding layer, suggesting non-linear composition or memorization. Aya-expanse and Gemma show high structural similarity, in particular at the innermost and outermost layers. Finally, Falcon and Qwen2.5 show moderate levels of structural similarity that drop last-minute. We discuss these differences in detail in Section 5.



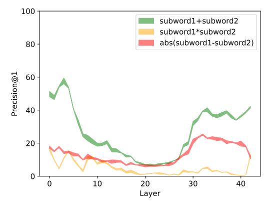
(a) Llama3



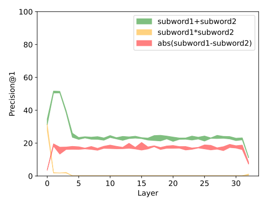
(b) Llama3.1



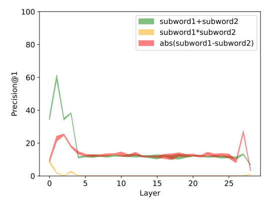
(c) Aya-expanse



(d) Gemma2



(e) Falcon



(f) Qwen2.5

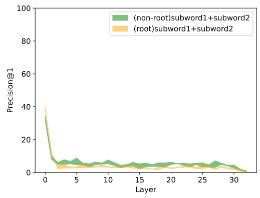
Figure 3: Structural similarity between LLM composition (base version, not instruction-tuned) and simple composition (P@1). Green band is simple addition. Orange refers to multiplication. Red is the performance of absolute difference. The colored bands indicate standard deviation. Instruction tuning seems to have little to no impact on our results; compare with Figure 2.

Impact of Instruction Tuning All models so far were instruction-tuned. Could differences in instruction tuning explain the differences between the compositional strategies of the six LLMs? We investigate this by repeating our experiments on the base versions of the above models. This allows us to evaluate the impact on instruction-tuning on structural similarity of LLM composition and simple composition.

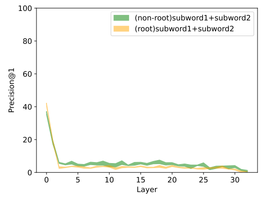
The patterns in Figure 3 are very similar to those observed for instruction-tuned models (Figure 2). Simple addition consistently produces composed vector spaces that most closely resemble the original word vector spaces, with same three distinct groups emerging. The only small difference lies in relative performance. Structural similarity is slightly higher in instruction-tuned models compared to their base versions, while the overall patterns remain unchanged. This suggests that although instruction-tuning enhances general similarity scores, it is not the key factor driving the isometry between LLM composition and simple arithmetic operations. Instead, the structural similarity is induced during pre-training. Pre-training on large-scale corpora captures distributional and compositional regularities, inducing representations designed to facilitate composition (one way or another). What is perhaps surprising is the degree to which LLMs differ in how representations are composed. Instruction tuning improves overall similarity, but seems to merely act as a refinement process, rather than having impact on compositional strategies.

Root and Non-Root Words The words in our dataset can be categorized into root and non-root words; see §3.1 for details. Since simple addition gave the best performance in the above, we rely on this form of composition in the following experiments. We now analyze how structural similarity varies across root and non-root words. Our hypothesis is that non-root words, which can be broken down into smaller meaningful units, will exhibit higher structural similarity, whereas root words, which cannot and lack obvious internal structure, will exhibit weaker alignment.

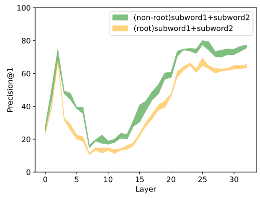
Figure 4 illustrates that across different models and layers, non-root words consistently exhibit higher structural similarity than root words. This suggests that simple addition more effectively produces a composed vector representation that aligns linearly with the original word representation for non-root words. This was expected and lends support to our original hypothesis. In contrast, root words exhibit weaker linear alignment, likely because they function as semantic atoms that are not easily decomposed into smaller parts in a meaningful way. Since their meanings are not derived from the interaction of multiple components, their representations may be shaped more by contextual factors and usage patterns than by explicit compositional relationships. This could introduce greater variability in their spatial organization, leading to generally lower structural similarity.



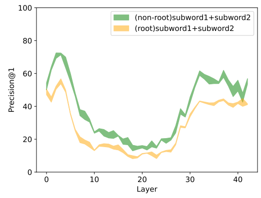
(a) Llama3



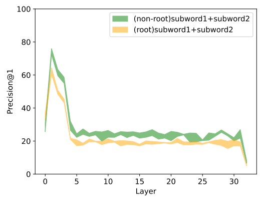
(b) Llama3.1



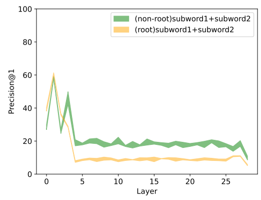
(c) Aya-expanse



(d) Gemma2



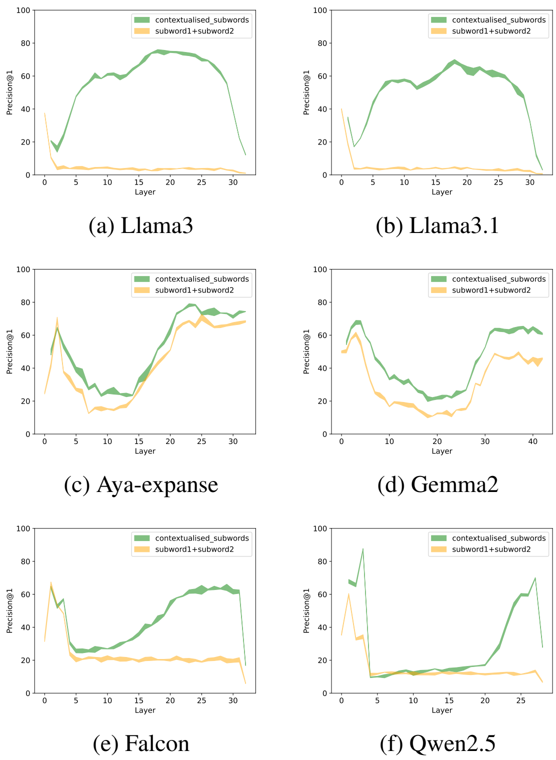
(e) Falcon



(f) Qwen2.5

Figure 4: Structural similarity between LLM composition and simple composition (P@1) across all layers and different word types. Green refers to the performance on non-root words. Orange refers to root words.

Impact of Contextualization Previous experiments have investigated the structural similarity between composed vectors—formed by combining two separate static subword representations—and original word representations. Many recent approaches using LLMs produce word, phrase, or sentence embeddings by applying mean pooling over their contextualized token representations. In this experiment, we take a similar approach by feeding both subwords into the LLM simultaneously, allowing their representations to interact and refine with each other. We then examine whether a simple addition of these contextualized subword representations can effectively reconstruct a composed representation that maintains structural similarity



(a) Llama3

(b) Llama3.1

(c) Aya-expanse

(d) Gemma2

(e) Falcon

(f) Qwen2.5

Figure 5: Structural similarity between LLM composition and simple composition (P@1) across all layers w/wo contextualization. Green refers to the performance with contextualization. Orange refers to without contextualization.

to the original word representation.

Figure 5 compares results with and without contextualization. When contextualization is applied, all models exhibit stronger linear alignment across layers. Notably, Llama models, along with Falcon and Qwen2.5, display distinct patterns. Instead of showing minimal structural similarity, Llama models demonstrate high levels of isometry in their middle layers. Falcon and Qwen2.5 also achieve higher P@1 scores in later layers. Meanwhile, Aya and Gemma models maintain a pattern similar to the non-contextualized scenario, but with generally higher structural similarity. These findings suggest that for some LLMs, e.g., Llama and Llama3.1, composed representations are only similar to simply arithmetic compositions when the LLM has observed both subwords in the same context. This highlights two distinct composition mechanisms. The first, seen in Aya and Gemma2, allows a linearly alignable composed representation to be directly formed by adding the separate subword representations. The second, observed in Llama, requires the model to process the subwords in the same context before producing a linearly alignable composed representation, possibly indicating higher degrees of memorization.

4 Probing Analysis

Previous geometry experiments have demonstrated that there exists a high degree of structural similarity between composed representations and the whole-word representations. However, this structural similarity varies across layers and models. In the following experiments, we investigate whether some basic aspects of the word understanding, specially content and form, have been preserved in the composed representation.

4.1 Root and Non-Root Words

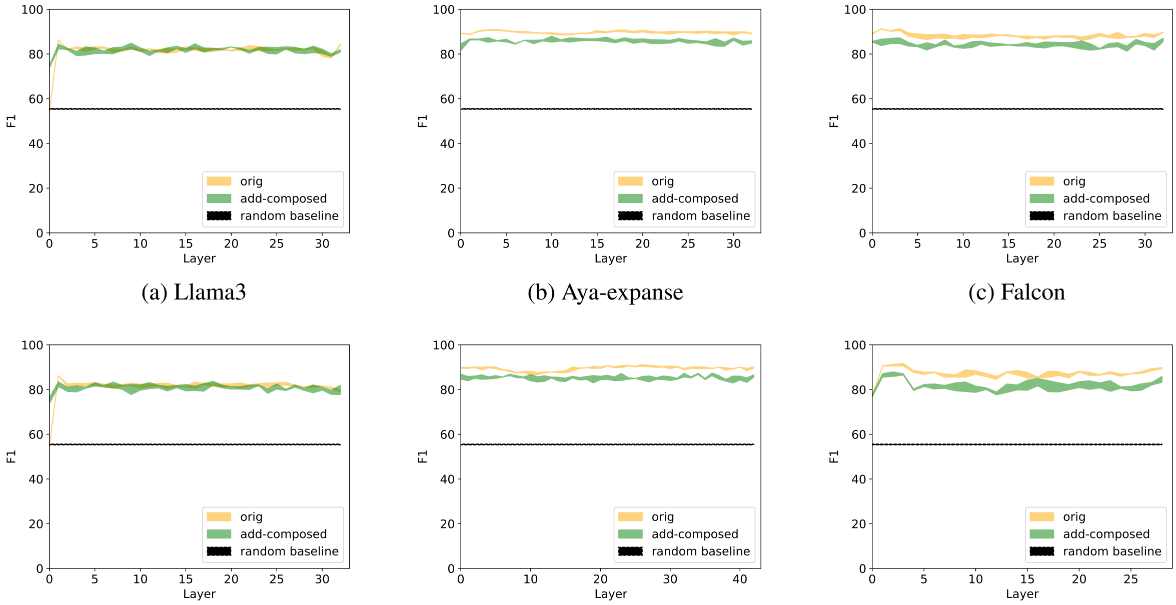
As shown in Table 3.1, words in the dataset can be classified into root and non-root words. Identifying whether a given vector representation corresponds to a root or non-root word requires capturing content information. Root words are the smallest meaningful units that cannot be broken down further, whereas non-root words are decomposable in meaning.

Method This word type prediction task is framed as a binary classification problem. We train a simple logistic regression model using either the original word representations or the composed subword representations as input. The classifier is trained for three epochs with a batch size of 8, utilizing the Adam optimizer with a learning rate of 1e-3.

Results The experiment results, measured by the weighted F1 score, are summarized in Figure 6. The orange line represents the weighted F1 score across all layers using the original word representations as input, while the green line shows the performance when using composed representations obtained by summing two subword representations.

Preliminary experiments indicate that a random baseline (black line) would achieve approximately 56% weighted F1 score. In contrast, features extracted from composed representations enable the model to achieve over 80% weighted F1 score, demonstrating that the distinction between root and non-root words is inherently embedded in the composed representations. The small gap between the green and original lines further suggests a high degree of content information preservation.

Despite the variations in structural similarity observed in previous geometry analysis, composed



(a) Llama3 (b) Aya-expanse (c) Falcon

(d) Llama3.1 (e) Gemma2 (f) Qwen2.5

Figure 6: Performance (weighted F1) of different LLMs on word type classification across all layers. Orange indicates the performance of using the original whole words. Green refers to addition-composed performance, and black is the random baseline. The colored bands indicate standard deviation.

representations maintain consistently high performance across different layers in the word type prediction task. This implies that even if a composed representation does not perfectly align with the original word representation in vector space, it could still preserve essential semantic information about the word.

4.2 Word Length Prediction

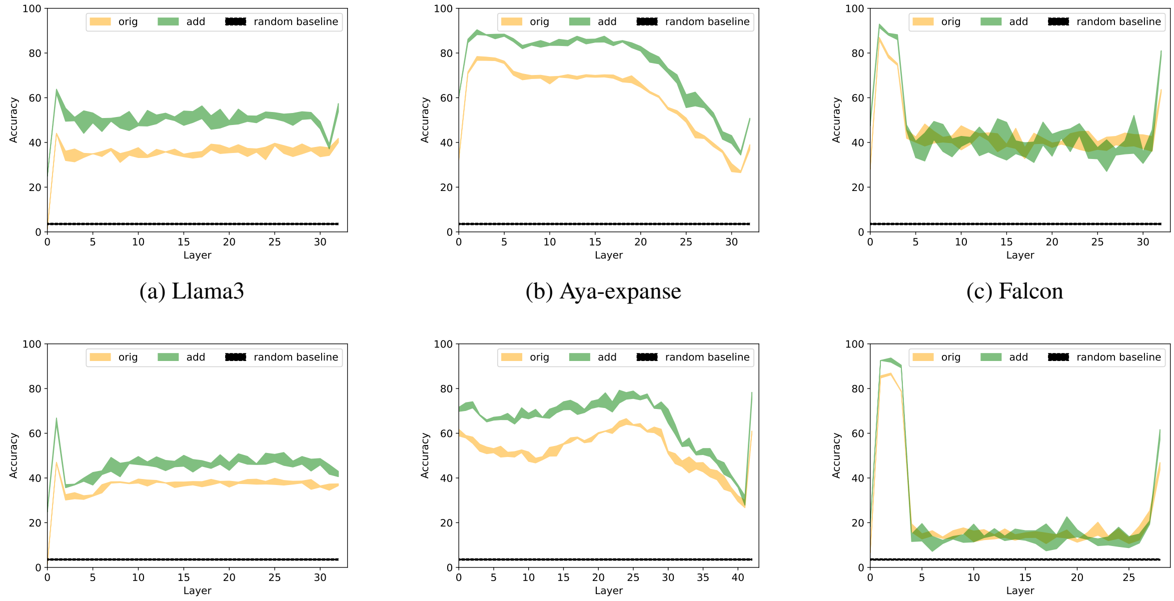
Having considered semantic classes, we now investigate whether LLMs retain form-related properties of subword constituents, specifically whether information about *word length* is passed up the network. Similar to our earlier experiment, we assess whether this information is encoded by predicting word length from both original and composed representations.

Method We formulate word length prediction as a regression task. Using linear regression, we predict the word length from a given vector representation. The regressor is trained for three epochs with a batch size of 8, using Adam optimizer with a learning rate of 1e-3. Since word length is a discrete value, the predicted outputs are rounded to the nearest integer before computing accuracy.

Results Figure 7 presents the overall accuracy across different models and layers. A random baseline (black line) achieves approximately 3.5% accuracy, reflecting the difficulty of the task without meaningful features. In contrast, both original word representations and composed subword representations result in significantly higher accuracy, demonstrating that word length information is inherently encoded in these embeddings.

Across all six LLMs, a consistent pattern emerges: the highest accuracy is observed in the early layers, suggesting that form-related properties, such as word length, are well-preserved at lower levels of the representation. However, as layers deepen, accuracy gradually decreases, likely due to the increasing abstraction of form information. Interestingly, at the final layers, accuracy improves again, indicating that some form-related information re-emerges at later processing stages. This suggests that while middle layers prioritize semantic abstraction, early and late layers retain more explicit surface-level features.

Consistent with the geometry analysis, these six models can be grouped into the same three categories based on their layer-wise accuracy patterns: (1) Llama3 and Llama3.1, (2) Aya and Gemma2, and (3) Falcon and Qwen2.5. The similar trends observed across models reinforce the idea that there are some systematic differences in their internal composition strategies that lead to systematic differ-



(a) Llama3 (b) Aya-expanse (c) Falcon

(d) Llama3.1 (e) Gemma2 (f) Qwen2.5

Figure 7: Performance (Accuracy) of different LLMs on word length prediction across all layers. Orange indicates the performance of using the original whole words. Green refers to addition-composed performance, and black is the random baseline. The colored bands indicate standard deviation.

ences in how they encode and retain form-related properties across layers.

5 Different Composition Strategies

The experimental results strongly indicate that the six LLMs can be categorized into three distinct groups. This pattern emerges consistently across our geometry analysis and probing tasks, suggesting that these differences stem from systematic variations in composition strategies rather than random noise.

The first group, which includes Aya and Gemma2, demonstrates a strong structural alignment between composed representations and original word representations across all layers. These models maintain high precision in geometry experiments, and their word type and length prediction performance remains stable, suggesting that both relevant information are generally well preserved. This implies that these models use a relatively direct and stable composition strategy, where subword embeddings are combined in a way that closely resembles the whole-word embedding throughout all layers. The fact that geometries are isometric to a very large degree, and both form-related and content-related attributes are restored, means the derivation history is implicitly kept, making the input more easily derivable from the output.

The second group, represented by Falcon and Qwen2.5, follows a different trend. In early layers, their composed representations exhibit good structural similarity with whole-word representations, but this alignment weakens in later layers. The word type semantic information remains relatively stable, but form-related information such as word length disappears in mid-layers and re-emerges towards the end. This suggests that these models initially retain subword structures but shift towards more abstract representations in deeper layers. Rather than maintaining a fixed composition throughout, they seem to undergo a transformation process where subword-based structure gives way to more semantic abstraction.

The third group, consisting of Llama3 and Llama3.1, exhibits a rapid loss of structural similarity beyond the embedding layer. While the word type prediction results indicate that semantic content is still preserved, form-related features degrade much more quickly than in the other groups. This suggests a more aggressive abstraction process where subword compositions are quickly absorbed into high-level representations, losing their original structural alignment. Unlike the first group, which retains subword traces throughout, these models prioritize semantic fusion over maintaining direct compositional structure.

As discussed in Section 3.3, such distinct patterns are already established during pre-training phase. Given the similarities in model architecture and training paradigms across these LLMs, we hypothesize that the main factor leading to this distinction is pre-training data and its data mixture. However, since such information is not fully disclosed[3] for the models we experimented, drawing a definitive conclusion remains challenging. We hope our work provides insights for future work on exploring different composition strategies.

6 Conclusion

In this work, we examine subword compositionality from the perspective of vector spaces, focusing on three key dimensions: structural similarity, content, and form understanding. Experimental results demonstrate that certain composition operations produce representations that are structurally similar to the original word representations. Additionally, we conducted two probing tasks to analyze content and form information. The results show that content information is consistently preserved across different models and layers, while the preservation of form information exhibits a more variable pattern. The performance of six different LLMs reveals three distinct groups based on their composition strategies.

Limitations

Our work provides valuable insights into subword composition in LLMs, but several limitations should be noted. First, the size of our dataset (3,432 words) reflects a trade-off between the number of models analyzed and the number of words included. Since different models have varying vocabularies, selecting words (and subwords) that exist across all models required balancing dataset size and model coverage. While carefully selected, the dataset may not fully capture the full range of word structures. Expanding it could offer an even more comprehensive understanding. Additionally, our work focuses on two-subword composition. It would be valuable to extend to compositions with more subwords. Second, our analysis is focused on English, and it remains an open question whether the same composition strategies hold across languages with different morphological properties. Extending this study to other languages would provide a broader perspective on subword composition in LLMs. Third, we have identified three distinct composition strategies, but the underlying reasons for these differences remain to be explored. Factors such as pre-training data and data mixture may play a role, and further investigation could shed light on why LLMs adopt these different composition behaviors.

Ethical Consideration

We do not anticipate any risks in the work. In this study, our use of existing artifacts is consistent with their intended purposes. The dataset is under the Creative Commons Attribution-ShareAlike 3.0 Unported License. Falcon[4] and Qwen2.5[5] models are under Apache-2.0. Aya-expanse models[6] are under cc-by-nc-4.0. Llama3[7] and Llama3.1[8] are under the Llama3 and Llama3.1 Community License Agreements respectively. Gemma2 models[9] are under the Gemma Terms of Use.

Acknowledgement

We would like to thank all anonymous reviewers for their insightful comments and feedback. This work was supported by DisAI - Improving scientific excellence and creativity in combating disinformation with artificial intelligence and language technologies, a project funded by European Union under the Horizon Europe, GA No. 101079164.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Merouane Debbah, Etienne Goffinet, Daniel Heslow, Julien Launay, Quentin Malartic, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023a. Falcon-40B: an open large language model with state-of-the-art performance.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru,

[3] We include all available information on data mixture in the appendix.

[4] huggingface.co/tiiuae/falcon-7b-instruct

[5] huggingface.co/Qwen/Qwen2.5-7B-Instruct

[6] huggingface.co/CohereForAI/aya-expanse-8b

[7] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[8] https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct

[9] https://huggingface.co/google/gemma-2-9b-it

Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, et al. 2023b. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Khuyagbaatar Batsuren, Gábor Bella, Aryaman Arora, Viktor Martinovic, Kyle Gorman, Zdeněk Žabokrtský, Amarsanaa Ganbold, Šárka Dohnalová, Magda Ševčíková, Kateřina Pelegrinová, Fausto Giunchiglia, Ryan Cotterell, and Ekaterina Vylomova. 2022. The SIGMORPHON 2022 shared task on morpheme segmentation. In *Proceedings of the 19th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 103–116, Seattle, Washington. Association for Computational Linguistics.

Khuyagbaatar Batsuren, Ekaterina Vylomova, Verna Dankers, Tsetsuukhei Delgerbaatar, Omri Uzan, Yuval Pinter, and Gábor Bella. 2024. Evaluating subword tokenization: Alien subword composition and oov generalization challenge. *arXiv preprint arXiv:2404.13292*.

Lorenzo Bertolini, Julie Weeds, David Weir, and Qiwei Peng. 2021. Representing syntax and composition with geometric transformations. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3343–3353, Online. Association for Computational Linguistics.

Ned Block. 1981. Psychologism and behaviorism. *Philosophical Review*, 90(1):5–43.

Qi Cao, Takeshi Kojima, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Unnatural error correction: GPT-4 can almost perfectly handle unnatural scrambled text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8898–8913, Singapore. Association for Computational Linguistics.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. 2024a. Tokenization falling short: On subword robustness in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1582–1599, Miami, Florida, USA. Association for Computational Linguistics.

Yekun Chai, Qingyi Liu, Jingwu Xiao, Shuohuan Wang, Yu Sun, and Hua Wu. 2024b. Autoregressive pre-training on pixels and texts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3106–3125, Miami, Florida, USA. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Kocmi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, Acyr Locatelli, Hangyu Lin, Dwarak Talupuru, Bharat Venkitesh, David Cairuz, Bowen Yang, Tim Chung, Wei-Yin Ko, Sylvie Shang Shi, Amir Shukayev, Sammie Bae, Aleksandra Piktus, Roman Castagné, Felipe Cruz-Salinas, Eddie Kim, Lucas Crawhall-Stein, Adrien Morisot, Sudip Roy, Phil Blunsom, Ivan Zhang, Aidan Gomez, Nick Frosst, Marzieh Fadaee, Beyza Ermis, Ahmet Üstün, and Sara Hooker. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *Preprint*, arXiv:2412.04261.

Ishita Dasgupta, Demi Guo, Andreas Stuhlmüller, Samuel J Gershman, and Noah D Goodman. 2018. Evaluating compositionality in sentence embeddings. *arXiv preprint arXiv:1802.04302*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.

Allyson Ettinger, Ahmed Elgohary, Colin Phillips, and Philip Resnik. 2018. Assessing composition in sentence vector representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1790–1801, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Leonidas Gee, Andrea Zugarini, Leonardo Rigutini, and Paolo Torroni. 2022. Fast vocabulary transfer for language model compression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 409–416, Abu Dhabi, UAE. Association for Computational Linguistics.

Hongyu Gong, Suma Bhat, and Pramod Viswanath. 2017. Geometry of compositionality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Josef Klafka and Allyson Ettinger. 2020. Spying on your neighbors: Fine-grained probing of contextual embeddings for information about surrounding words. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4801–4811, Online. Association for Computational Linguistics.

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

Jiaang Li, Yova Kementchedjhieva, Constanza Fierro, and Anders Søgaard. 2024a. Do vision and language models share concepts? a vector space alignment study. *Transactions of the Association for Computational Linguistics*, 12:1232–1249.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024b. Understanding and patching compositional reasoning in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9668–9688, Bangkok, Thailand. Association for Computational Linguistics.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, et al. 2024. Starcoder 2 and the stack v2: The next generation. *arXiv preprint arXiv:2402.19173*.

Qiwei Peng and Anders Søgaard. 2024. Concept space alignment in multilingual LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5511–5526, Miami, Florida, USA. Association for Computational Linguistics.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2023. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations*.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5149–5152. IEEE.

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.

Chen Shani, Jilles Vreeken, and Dafna Shahaf. 2023. Towards concept-aware large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13158–13170, Singapore. Association for Computational Linguistics.

Yintao Tai, Xiyang Liao, Alessandro Suglia, and Antonio Vergari. 2024. PIXAR: Auto-regressive language modeling in pixel space. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14673–14695, Bangkok, Thailand. Association for Computational Linguistics.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team. 2024a. Gemma.

Qwen Team. 2024b. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

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

Changhan Wang, Kyunghyun Cho, and Jiatao Gu. 2020. Neural machine translation with byte-level subwords. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 9154–9160.

Dixuan Wang, Yanda Li, Junyuan Jiang, Zepeng Ding, Guochao Jiang, Jiaqing Liang, and Deqing Yang. 2024a. Tokenization matters! degrading large language models through challenging their tokenization. *arXiv preprint arXiv:2405.17067*.

Siyuan Wang, Zhongyu Wei, Yejin Choi, and Xiang Ren. 2024b. Can LLMs reason with rules? logic scaffolding for stress-testing and improving LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7523–7543, Bangkok, Thailand. Association for Computational Linguistics.

Di Wu, Yibin Lei, Andrew Yates, and Christof Monz. 2024. Representational isomorphism and alignment of multilingual large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14074–14085, Miami, Florida, USA. Association for Computational Linguistics.

Ningyu Xu, Qi Zhang, Menghan Zhang, Peng Qian, and Xuanjing Huang. 2024. On the tip of the tongue: Analyzing conceptual representation in large language models with reverse-dictionary probe. *arXiv preprint arXiv:2402.14404*.

Zhaozhen Xu, Zhijin Guo, and Nello Cristianini. 2023. On compositionality in data embedding. In *International Symposium on Intelligent Data Analysis*, pages 484–496. Springer.

Lang Yu and Allyson Ettinger. 2020. Assessing phrasal representation and composition in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4896–4907, Online. Association for Computational Linguistics.

A Data Mixture for Different LLMs

Llama3 and Llama3.1 As revealed in Dubey et al. (2024), the final data mix for Llama3 pretraining contains roughly 50% of tokens corresponding to general knowledge, 25% of mathematical and reasoning tokens, 17% code tokens, and 8% multilingual tokens.

Falcon The pre-training data mixture for Falcon is summarized in Figure 8.

Corpora Name	Source	Stock	**Pretraining** Fraction	Used
RefinedWeb-English	Filtered and deduplicated CommonCrawl, see Penedo et al. (2023)	~5,000B	76%	2,700B
RefinedWeb-Euro	Filtered and deduplicated multilingual (Europe-focused) CommonCrawl, see Penedo et al. (2023)	~2,000B	8%	400B
Books	Project Gutenberg	215B	6%	214B
Conversations	Reddit, StackOverflow, HackerNews, IRC, YouTube Subtitles	170B	5%	168B
Code	GitHub	~1,000B	3%	115B
Technical	arXiv, PubMed, USPTO, Wikipedia	60B	2%	57B

Figure 8: The figure taken from Almazrouei et al. (2023b) that illustrates pre-training data mixture in Falcon models.

Gemma2 Gemma 2 models (9B) are pre-trained on 8 trillion tokens. These tokens come from a variety of data sources, including web documents, code, and science articles. However, exact proportions of these data types are not disclosed. Instead, it is noted that the final data mixture was determined through ablations similar to the approach in Gemini 1.0 (Team et al., 2023).

Aya-expanse The details of pre-training data mixture is not mentioned or discussed in Dang et al. (2024).

Qwen2.5 The fraction of data mixture for Qwen2.5 models are not revealed. Team (2024b) mentions that they employ Qwen2-Instruct models to optimize the pre-training data distribution across different domains and results in a pre-training data of 18 trillion tokens.

Tokenizers Tokenizers of these different LLMs all adopt the BPE algorithm and give quite similar results for tokenization. This is also why we choose these six models as they give the highest overlap in words, ensuring enough data to experiment with.