Why bother with geometry?
On the relevance of linear decompositions of Transformer embeddings

Timothee Mickus
University of Helsinki
timothee.mickus@helsinki.fi

Raúl Vázquez
University of Helsinki
raul.vazquez@helsinki.fi

Abstract

A recent body of work has demonstrated that Transformer embeddings can be linearly decomposed into well-defined sums of factors, that can in turn be related to specific network inputs or components. There is however still a dearth of work studying whether these mathematical reformulations are empirically meaningful. In the present work, we study representations from machine-translation decoders using two of such embedding decomposition methods. Our results indicate that, while decomposition-derived indicators effectively correlate with model performance, variation across different runs suggests a more nuanced take on this question. The high variability of our measurements indicate that geometry reflects model-specific characteristics more than it does sentence-specific computations, and that similar training conditions do not guarantee similar vector spaces.

1 Introduction

It stands to reason that important research efforts are being devoted to explaining the behavior and understanding the mechanics of Transformer-based NLP models: Most models that are currently discussed within the NLP community are based on this architecture, and they have achieved resounding successes. One trend of work in particular attempts to characterize Transformer models by means of their geometry (Rogers et al., 2020; Ethayarajh, 2019; Timkey and van Schijndel, 2021, e.g.,).

However, most studies focus on a single handful of 'foundation' models or fine-tuned variants thereof, and explicitly or implicitly assume that the reported results generalize on to other models—yet the effects of random initialization, training data or variation in the definition of objective functions are left unstudied. Moreover, and perhaps more crucially, there is no guarantee that Transformer-embeddings geometry is indicative of model quality: That embeddings are arranged in a certain fashion in hyperspace says little of what downstream performance we should expect.

Taken together, these two assumptions—that results applicable to one model will apply to many, and that geometry can provide explanations—call into question the validity of geometry-based approaches. We focus on two linear decomposition approaches for Transformer embeddings (Mickus et al., 2022; Oh and Schuler, 2023): works attempting to summarize model computation through their effects on the resulting output embedding spaces. By construction, these decompositions reflect topological features of the Transformer architecture.

Our goal is to verify whether these two assumptions are in fact supported. One would hope, for instance, that the geometry a model settles on differs along training data but not random initialization. Another natural expectation to have is that different uses of the model, such as forcing the production of a given sentence or searching for a plausible generation would yield distinct computations and therefore distinct geometric arrangements. Lastly, if we intend to explain model performance via embedding geometry, then we should observe consistent differences in geometry whenever we see differences in quality metrics.

To answer whether all three of these expectations are met, we experiment with machine-translation decoder embeddings and study how their geometry evolves over training and multilingualism. In a nutshell, our experiments suggest a nuanced outlook on the usefulness of linear decompositions. Decomposition-derived indicators tend to correlate well with corpus-level model performance, but are less appropriate when it comes to sentence-level performance. We also observe that variation in geometry across different runs for a same translation task can exceed what we observe for models trained for different translation tasks.

Our findings question the relevance of geometry-based approaches for Transformer model explain-

Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP, pages 127–141
December 7, 2023. ©2023 Association for Computational Linguistics

ability. As our measurements display high variability across different model training runs, this work suggests that geometry reflects model-specific characteristics more than it does sentence-specific computations: Models trained in similar conditions need not yield similar vector spaces.

2 Related works

There is a rich literature that connects the objective of static embedding models such as word2vec to characteristics of the resulting vector space. In particular, Allen and Hospedales (2019) worked out how the linguistic regularities found by Mikolov et al. (2013) result from the exact loss landscape.

As for contextual embeddings, research has been more commonly limited to empirical observations (Ethayarajh, 2019; Timkey and van Schijndel, 2021, e.g.). Recently, Ferrando et al. (2022b,a); Modarressi et al. (2022); Mickus et al. (2022); Oh and Schuler (2023); Yang et al. (2023) and others have developed methods to provide mechanistic interpretations of Transformer outputs: These works rely on linear algebra to derive mathematically exact attributions, where a contextual embedding is decomposed as a sum of interpretable vector terms.

These approaches build upon two peculiarities of the Transformer architecture. Perhaps the most famous one—at least, one that has found significant traction more generally across explainable NLP—is that of the scaled dot *attention mechanism*. Transformers were presented by Vaswani et al. (2017) as attention-only models. Attention mechanisms can be seen as weighted sums over value vectors (Kobayashi et al., 2020), where the attention weights are derived non-linearly. This observation was first brought forth within the sustained and ongoing discussion about the relevance of attention weights, and whether they are efficient means of explaining Transformer behaviors—a subject hotly debated (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019; Pruthi et al., 2020). In particular, Serrano and Smith (2019); Kobayashi et al. (2020) highlight the importance of considering the full embedding space geometry.

The second characteristic of importance to Transformer decomposition approaches is the systematic use of residual connections throughout a Transformer models, only interrupted by layer normalization operations. This fact, often described as a *residual stream* of information, has been leveraged to interpret the behavior of feed-forward sub-layers (Geva et al., 2021; Ferrando et al., 2023; Dar et al., 2023) or layer commutativity (Zhao et al., 2021). Given that on the one hand a layer norm is a linear map, and on the other hand a residual connection simply consists on adding a sub-network's input to its output, this entails that most of the computations done in a Transformer are distributive.

3 Methodology

Our focus here is on sequence-to-sequence encoder-decoder architectures (Vaswani et al., 2017). Transformer embeddings can be decomposed into a linear combination of nonlinear transformations using properties of the residual connections and attention mechanisms. Here, we focus on whether these decompositions do provide meaningful explanations, or whether they merely reflect topological characteristics of the Transformer architecture.[1]

3.1 Models & Data

Connecting with previous literature (Voita et al., 2021; Ferrando et al., 2022a; Vázquez et al., 2022, e.g.), our focus in this work is on decoder embeddings from Transformers trained on machine translation (MT) objectives, with varying degrees of multilinguality. NMT systems provide a useful framework to study the validity of explainability methodologies. First, significant efforts have been devoted to the creation of MT evaluation metrics that correlate well with human intuitions. Empirical investigations of what drives phenomena such as hallucinations also abound. Lastly, translation as a task has the advantage that is straightforward for humans to relate input to output.

Our models are trained on different subsets of the Tatoeba Challenge corpus (Tiedemann, 2020), each of them sampling up to 5M sentences per language pair. We train models with sources of different degrees of multilinguality: multilingual-to-English, with 76M sentences; Indo-European-to-English, with 58M sentences; Slavic-to-English, with 33M sentences; and Russian-to-English with 5M sentences. For the bilingual dataset (Ru–En), we train three different model seeds. All models are trained using the marian-MT library (Junczys-Dowmunt et al., 2018) for 72 hours on 4 V100 GPUs. We saved checkpoints every 1000 training steps to compare decompositions at different training stages. Hyperparameters and training details are listed in appendix A. We systematically run all

[1] Code at `github.com/TimotheeMickus/seq2seq-splat`.

of our experiments on the same held out test set of 19,425 Russian and English paired sentences.

3.2 Decomposition approaches

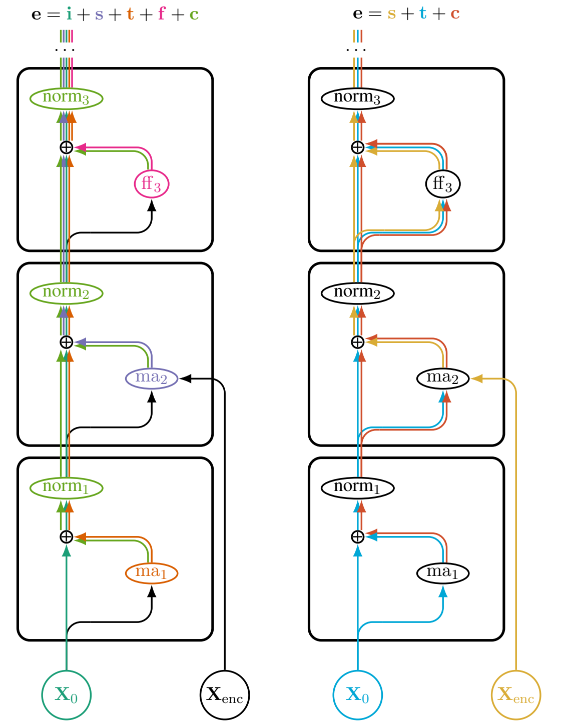
$\mathbf{Z}$	matrix
$(\mathbf{Z})_i$	i^{th} row of $\mathbf{Z}$
$\mathbf{z}$	(row) vector
k, κ, K	scalars
$\mathbf{y} \oplus \mathbf{z}$	concatenation of vectors $\mathbf{y}$ and $\mathbf{z}$
$\bigoplus_n \mathbf{z}_n$	$\mathbf{z}_1 \oplus \mathbf{z}_2 \oplus \cdots \oplus \mathbf{z}_n$
$\mathbf{y} \odot \mathbf{z}$	element-wise multiplication of $\mathbf{y}$ and $\mathbf{z}$
$\bigodot_n \mathbf{z}_n$	$\mathbf{z}_1 \odot \mathbf{z}_2 \odot \cdots \odot \mathbf{z}_n$
$\vec{1}$	vector with all components set to 1
$\vec{0}$	vector with all components set to 0
$\mathbf{0}_{m,n}$	null matrix of shape $m \times n$
$\mathbf{I}_n$	identity matrix of shape $n \times n$

(a) General notations

Λ	total number of sub-layers
λ	sub-layer index
L	total number of layers, i.e., $\Lambda/3$
l	layer index
d	dimension of representations
H	number of heads
$\mathbf{W}^{(\text{m})}_{\lambda}$	sub-module m in sub-layer λ weight matrix
$\mathbf{b}^{(\text{m})}_{\lambda}$	bias for sub-module m in sub-layer λ
$\mathbf{g}^{(\text{ln})}_{\lambda}$	layer-norm gain parameter in sub-layer λ
$\mathbf{E}_{\lambda}$	output of sub-layer λ (all embeddings)
$\mathbf{e}_{\lambda,t}$	output of sub-layer λ at position t
$\dot{\mathbf{e}}_{\lambda,t}$	output of sub-layer λ at position t before layer-norm and residual connection
$\ddot{\mathbf{e}}_{\lambda,t}$	output of sub-layer λ at position t before layer-norm
$\mathbf{X}_{\lambda}$	target-side input to sub-layer λ, $\mathbf{E}_{\lambda-1}$
$\mathbf{x}_{\lambda,t}$	t^{th} target-side input of sub-layer λ, $\mathbf{e}_{\lambda-1}$
$\mathbf{X}_{\text{enc}}$	memory bank, i.e., output of the encoder
$\mathbf{A}_{\lambda,h}$	attention weight matrix for h^{th} head of the multi-head attention at sublayer λ
$a_{\lambda h t t'}$	Attention weight for head h, sub-layer λ query t, value t'
ϕ	non-linear activation function
$m_{\lambda t}$	mean from the layer-norm of sub-layer λ
$s_{\lambda t}$	standard deviation from λ^{th} layer-norm

(b) Transformer-specific notations

Table 1: Notations



$\mathbf{e} = \mathbf{i} + \mathbf{s} + \mathbf{t} + \mathbf{f} + \mathbf{c}$
norm$_3$
ff$_3$
norm$_2$
ma$_2$
norm$_1$
ma$_1$
$\mathbf{X}_0$
$\mathbf{X}_{\text{enc}}$
$\mathbf{e} = \mathbf{s} + \mathbf{t} + \mathbf{c}$
norm$_3$
ff$_3$
norm$_2$
ma$_2$
norm$_1$
ma$_1$
$\mathbf{X}_0$
$\mathbf{X}_{\text{enc}}$

(a) Sub-layer-wise decomposition Dcp_{sl}

(b) Token-wise decomposition Dcp_{tok}

Figure 1: Overview of decomposition methods, focusing on the first three sublayers of the decoder. Colors indicate what a decomposition term is imputed to.

We consider two approaches: a sub-layer-wise decomposition, and a token-wise decomposition. They are inspired by Mickus et al. (2022) and Oh and Schuler (2023) and illustrated in fig. 1. In table 1, we list the notations used throughout this work. See appendix B for a primer on the Transformer architecture.

Sub-layer-wise decomposition. The first approach is a sub-layer-level decomposition in five terms. That is, we decompose embedding $\mathbf{e}$ into a linear combination of functions that refer to the target-side input $(\mathbf{i})$, the source attention $(\mathbf{s})$, the target attention $(\mathbf{t})$, the feed-forwards $(\mathbf{f})$, or the models' biases $(\mathbf{c})$. We note it as Dcp_{sl}. As can be seen in fig. 1a, it essentially entails that we break down embeddings depending on where in the network a specific term comes from. Hence, for token position t:

$$\mathbf{e}_t = \mathbf{i}_t + \mathbf{s}_t + \mathbf{t}_t + \mathbf{f}_t + \mathbf{c}_t \qquad (1)$$

where

$$\mathbf{i}_t = f^{(\text{ln})}_{1}\left(\mathbf{x}_{0,t}\right) \qquad (2)$$

$$\mathbf{t}_t = \sum_{l=0}^{\Lambda/3-1} f^{(\text{ln})}_{3l+1}\left(\left(f^{(\text{ma})}_{3l+1}\left(\mathbf{X}_{3l+1}\right)\right)_t\right) \qquad (3)$$

$$\mathbf{s}_t = \sum_{l=0}^{\Lambda/3-1} f^{(\text{ln})}_{3l+2}\left(\left(f^{(\text{ma})}_{3l+2}\left(\mathbf{X}_{\text{enc}}\right)\right)_t\right) \qquad (4)$$

$$\mathbf{f}_t = \sum_{l=0}^{\Lambda/3-1} f^{(\text{ln})}_{3l+3}\left(f^{(\text{ff})}_{3l+3}\left(\mathbf{x}_{3l+3,t}\right)\right) \qquad (5)$$

$$
\begin{aligned}
\mathbf{c}_t = \mathbf{b}_{\Lambda}^{(\mathrm{ln})} + f_1^{(\mathrm{ln})}(-m_1\vec{1}) + \sum_{\lambda=2}^{\Lambda} f_{\lambda}^{(\mathrm{ln})}\left(\mathbf{b}_{\lambda-1}^{(\mathrm{ln})} - m_\lambda\vec{1}\right) \\
+ \sum_{l=0}^{\Lambda/3-1} f_{\lambda+1}^{(\mathrm{ln})}\left(\mathbf{b}_{\lambda+1}^{(\mathrm{ma,O})} + \sum_{h=1}^{H}\mathbf{H}_{\lambda+1,h}\mathbf{b}_{\lambda+1}^{(\mathrm{ma,V})}\right) \\
+ \sum_{l=0}^{\Lambda/3-1} f_{\lambda+2}^{(\mathrm{ln})}\left(\mathbf{b}_{\lambda+2}^{(\mathrm{ma,O})} + \sum_{h=1}^{H}\mathbf{H}_{\lambda+2,h}\mathbf{b}_{\lambda+2}^{(\mathrm{ma,V})}\right) \\
+ \sum_{l=0}^{\Lambda/3-1} f_{\lambda+3}^{(\mathrm{ln})}\left(\mathbf{b}_{\lambda+3}^{(\mathrm{ff,out})}\right)
\end{aligned}
\tag{6}
$$

The cumulative effects of the layer-norms after sub-layer λ, $f_{\lambda}^{(\mathrm{ln})}(\mathbf{x})$, the unbiased outputs of a feed-forward layer, $f_{\lambda}^{(\mathrm{ff})}(\mathbf{x})$, and of a multi-head attention layer, $f_{\lambda}^{(\mathrm{ma})}(\mathbf{X})$, are defined as follows:

$$
\begin{aligned}
f_{\lambda}^{(\mathrm{ln})}(\mathbf{x}) &= \frac{1}{\prod_{\lambda'=\lambda}^{\Lambda} s_{\lambda'}} \bigodot_{\lambda'=\lambda}^{\Lambda} \mathbf{g}_{\lambda'} \odot \mathbf{x} \\
f_{\lambda}^{(\mathrm{ff})}(\mathbf{x}) &= \mathbf{W}_{\lambda}^{(\mathrm{ff,out})}\phi\left(\mathbf{W}_{\lambda}^{(\mathrm{ff,in})}\mathbf{x} + \mathbf{b}_{\lambda}^{(\mathrm{ff,in})}\right) \\
f_{\lambda}^{(\mathrm{ma})}(\mathbf{X}) &= \mathbf{W}_{\lambda}^{(\mathrm{ma,O})}\left(\bigoplus_{h=1}^{H}\mathbf{A}_{\lambda,h}\mathbf{W}_{\lambda,h}^{(\mathrm{ma,V})}\mathbf{X}\right)
\end{aligned}
$$

For convenience we also define the linear map associated with going from a given head h to the output of sub-layer λ:

$$
\begin{aligned}
\mathbf{H}_{\lambda,h} &= \mathbf{W}_{\lambda}^{(\mathrm{ma,O})}\mathbf{S}_h \\
\mathbf{S}_h &= \begin{bmatrix}\mathbf{0}_{\frac{d}{H},\frac{d(h-1)}{H}} & \mathbf{I}_{\frac{d}{H}} & \mathbf{0}_{\frac{d}{H},\frac{d(H-h)}{H}}\end{bmatrix}
\end{aligned}
$$

Token-wise decomposition. A major issue that stands in the way of linear decomposition approaches is the use of a non-linear activation function in feed-forward sub-layers. This has prompted different approaches: side-stepping the problem altogether and leaving this component unanalyzed (Mickus et al., 2022; Ferrando et al., 2022b; Modarressi et al., 2022); relying on local linear approximations of the activation function (Oh and Schuler, 2023); or limiting the scope of inquiry to activation functions with the desired mathematical properties (Yang et al., 2023).

The second decomposition we study, which we note $\mathrm{Dcp}_{\mathrm{tok}}$, uses the locally linear approximation of Oh and Schuler (2023) to distribute the feed-forward sub-layer outputs to the input decomposition. We then group all inputs into three terms $\mathbf{s}, \mathbf{t}, \mathbf{c}$, depending on whether a vector term ultimately comes from the encoder, from the target input ($\mathbf{t}$) or model biases ($\mathbf{c}$), as shown in fig. 1b. Unlike $\mathrm{Dcp}_{\mathrm{sl}}$, this entails grouping terms based on what they originally were. More formally, we define it as:

$$
\mathbf{e}_{\lambda,t} = \mathbf{s}_{\lambda,t} + \mathbf{t}_{\lambda,t} + \mathbf{c}_{\lambda,t} \tag{7}
$$

and compute these operands by recurrence.

If we start by setting aside layer normalization and residual connection for simplicity, we can get a first approximation of what should be attributed to the source-side input at a given sub-layer, given prior computations:

$$
\dot{\mathbf{s}}_{\lambda,t} = \begin{cases}
\sum_{t'=1}^{t} a_{\lambda h t t'}\mathbf{H}_{\lambda,h}\mathbf{W}_{\lambda,h}^{(\mathrm{ma,V})}\mathbf{s}_{\lambda-1,t'} \\
\qquad\qquad \text{if } \lambda \equiv 1 \mod 3 \\
\sum_{n}\left(f_{\lambda}^{(\mathrm{ma})}\left(\mathbf{X}_{\mathrm{enc}}\right)\right)_n \\
\qquad\qquad \text{if } \lambda \equiv 2 \mod 3 \\
\mathbf{F}_{\lambda,t}\mathbf{s}_{\lambda-1,t} \quad \text{if } \lambda \equiv 0 \mod 3
\end{cases} \tag{8}
$$

given the local linear approximation of the feed-forward, $\mathbf{F}_{\lambda,t} = \mathbf{W}_{\lambda}^{(\mathrm{ff,out})}\mathbf{L}_{\lambda,\mathbf{e}_{\lambda,t}}\mathbf{W}_{\lambda}^{(\mathrm{ff,in})}$. The local linear approximation itself $\mathbf{L}_{\lambda,\mathbf{x}}$ of the activation function ϕ for sub-layer λ is defined as:

$$
\mathbf{L}_{\lambda,\mathbf{x}} = \mathbf{I}_d \odot \phi'\left(\mathbf{W}_{\lambda}^{(\mathrm{ff,in})}\mathbf{x} + \mathbf{b}_{\lambda}^{(\mathrm{ff,in})}\right)
$$

We can also remark that in the initial stages, the source-side input is not used, meaning that:

$$
\mathbf{s}_{0,t} = \vec{0}
$$

With an analogous line of thought, we can characterize what in a given sub-layer hidden representation is owed to the target-side input as:

$$
\dot{\mathbf{t}}_{\lambda,t} = \begin{cases}
\sum_{t'=1}^{t} a_{\lambda h t t'}\mathbf{H}_{\lambda,h}\mathbf{W}_{\lambda,h}^{(\mathrm{ma,V})}\mathbf{t}_{\lambda-1,t'} \\
\qquad\qquad \text{if } \lambda \equiv 1 \mod 3 \\
\vec{0} \qquad\qquad \text{if } \lambda \equiv 2 \mod 3 \\
\mathbf{F}_{\lambda,t}\mathbf{t}_{\lambda-1,t} \quad \text{if } \lambda \equiv 0 \mod 3
\end{cases} \tag{9}
$$

$$
\mathbf{t}_{0,t} = \mathbf{x}_{0,t}
$$

And similarly, we can keep track of all biases and offsets thus far ignored:

$$\dot{\mathbf{c}}_{\lambda,t} = \begin{cases} \mathbf{b}_{\lambda}^{(\text{ma,O})} + \sum\limits_{h=1}^{H} \mathbf{H}_{\lambda,h}\mathbf{b}_{\lambda,h}^{(\text{ma,V})} \\ \quad + \sum\limits_{t'=1}^{t} a_{\lambda h t t'} \mathbf{H}_{\lambda,h}\mathbf{W}_{\lambda,h}^{(\text{ma,V})}\mathbf{c}_{\lambda-1,t'} \\ \qquad\qquad \text{if } \lambda \equiv 1 \mod 3 \\ \mathbf{b}_{\lambda}^{(\text{ma,O})} + \sum\limits_{h=1}^{H} \mathbf{H}_{\lambda,h}\mathbf{b}_{\lambda,h}^{(\text{ma,V})} \\ \qquad\qquad \text{if } \lambda \equiv 2 \mod 3 \\ \mathbf{b}_{\lambda}^{(\text{ff,out})} + \mathbf{W}_{\lambda}^{(\text{ff,out})}\mathbf{l}_{\lambda,\mathbf{e}_{\lambda,t}} \\ \quad + \mathbf{W}_{\lambda}^{(\text{ff,out})}\mathbf{L}_{\lambda,\mathbf{e}_{\lambda,t}}\mathbf{b}_{\lambda}^{(\text{ff,in})} \\ \quad + \mathbf{F}_{\lambda,t}\mathbf{c}_{\lambda-1,t} \\ \qquad\qquad \text{if } \lambda \equiv 0 \mod 3 \end{cases} \tag{10}$$

$$\mathbf{c}_{0,t} = \vec{0}$$

where the intercept of the local linear approximation of the feed-forward activation is defined as:

$$\mathbf{l}_{\lambda,\mathbf{e}_{\lambda,t}} = \phi\left(\hat{\mathbf{e}}_{\lambda,t}\right) - \mathbf{L}_{\lambda,\mathbf{e}_{\lambda,t}}\hat{\mathbf{e}}_{\lambda,t}$$
$$\hat{\mathbf{e}}_{\lambda,t} = \mathbf{W}_{\lambda}^{(\text{ff,in})}\mathbf{e}_{\lambda,t} + \mathbf{b}_{\lambda}^{(\text{ff,in})}$$

Finally, we need to account for residual connections and layer normalisation so as to obtain the exact decomposition for the next layer: [2]

$$\mathbf{s}_{\lambda,t} = \frac{1}{s_{\lambda,t}}\mathbf{g}_{\lambda} \odot \left(\dot{\mathbf{s}}_{\lambda,t} + \mathbf{s}_{\lambda-1,t}\right) \tag{11}$$

$$\mathbf{t}_{\lambda,t} = \frac{1}{s_{\lambda,t}}\mathbf{g}_{\lambda} \odot \left(\dot{\mathbf{t}}_{\lambda,t} + \mathbf{t}_{\lambda-1,t}\right) \tag{12}$$

$$\mathbf{c}_{\lambda,t} = \frac{1}{s_{\lambda,t}}\mathbf{g}_{\lambda} \odot \left(\dot{\mathbf{c}}_{\lambda,t} + \mathbf{c}_{\lambda-1,t} - m_{\lambda,t}\vec{1}\right) + \mathbf{b}_{\lambda}^{(\text{ln})} \tag{13}$$

3.3 Scalar indicators

Linear decomposition approaches, by design, yield sums of high-dimensional vectors. To reduce these vectors to comprehendable scalars, we consider two scalar-valued indicator metrics: one that evaluates the relative magnitude magnitude of a term in a linear decomposition with respect to the total embedding; and a cosine-based one as an indicator of co-directionality. We choose these indicators due to their simplicity and interpretability.

We define the *norm ratio* as the ratio of l_2 norms so as to capture a sense of scale, and the cosine similarity as:

$$\text{nr}\left(\mathbf{z}, \mathbf{e}\right) = \frac{\|\mathbf{z}\|_2}{\|\mathbf{e}\|_2} \tag{14}$$

$$\cos\left(\mathbf{z}, \mathbf{e}\right) = \frac{\mathbf{z} \cdot \mathbf{e}}{\|\mathbf{z}\|_2\|\mathbf{e}\|_2} \tag{15}$$

Intuitively, if a term $\mathbf{z}$ in some decomposition Dcp of a contextual embedding $\mathbf{e}$ has a small norm, then we should expect this term $\mathbf{z}$ to be unimportant as it effectively contributes little to the total embedding $\mathbf{e}$, resulting in a small norm ratio. On the other hand, when a term $\mathbf{z}$ has a large norm, this measure assigns importance to it, regardless of its orientation with respect to the total embedding $\mathbf{e}$. This is instead captured through cosine similarity: Co-directionality indicates whether a term $\mathbf{z}$ is pointing in the same direction as the total embedding $\mathbf{e}$ (when $\cos\left(\mathbf{z}, \mathbf{e}\right) = 1$) or in the opposite direction (when $\cos\left(\mathbf{z}, \mathbf{e}\right) = -1$). Cosine similarity has long been used in IR and embedding research (Singhal, 2001). Taken together, the two indicators provide a more complete picture, allowing interpretations while retaining simplicity.[3]

4 What is geometry indicative of?

Given our experimental protocol described in section 3, we now explore what is encoded in linear decomposition terms.

Do the decoding algorithms affect the geometry of embeddings? The first element we consider is whether *forced inference*, where we feed the gold target to the model, and a *beam-search* decoding produce different embeddings, as far as a linear decomposition would capture it. We consider these two decoding algorithms, as they are commonly used in MT studies; moreover we strongly expect that they should entail different behaviors and information flows through the network: Forced decoding uses a gold reference translation in addition to the source sentence, while beam search doesn't receive this input but instead is a mode

[2] As our interest lies in disentangling source and target-side contributions, the decomposition above does not properly attribute weights to individual tokens, i.e., all inputs are not disentangled. Also remark that the local linear approximation $\mathbf{L}_{\lambda,\mathbf{x}}$ is defined with respect to the hidden state $\mathbf{e}_{\lambda,t}$: As such, the computations it describes are specific to a particular contextualized embedding, which obfuscates token-level attribution.

[3] In preliminary experiments, we also experimented with Euclidean distance as well as the scalar product importance metric μ of Mickus et al. (2022), eq. 6. We do not include them in the present article for simplicity. Also remark that for all decomposition term $\mathbf{z}$ of a given embedding $\mathbf{e}$, we have $\cos(\mathbf{z}, \mathbf{e})\text{nr}(\mathbf{z}, \mathbf{e}) = \mu(\mathbf{z}, \mathbf{e})$

searching heuristic. It is sensible to expect that the decomposition of embeddings from both decoding algorithms differ.

In particular, it makes sense to consider how forced inference and beam-search decoding evolve across training. Models are trained to optimize the likelihood on iid. data: As such, differences between these two decoding algorithms—if any are to be found—should become less important as training progresses. Hence, for each of our six models detailed in section 3.1, we consider the embeddings obtained at intervals of 1000 updates: i.e., we compute output decoder embeddings after $1000, 2000, \ldots, 1000N$ updates. We can then measure whether scalar indicators defined in section 3.3 differ across updates when using forced inference or beam-search.

In other words, we define series of paired scalar observations for each model, depending on which decomposition ($\mathrm{Dcp} \in \{\mathrm{Dcp}_{\mathrm{tok}}, \mathrm{Dcp}_{\mathrm{sl}}\}$), term (viz., $\mathbf{z} \in \{\mathbf{i}, \mathbf{s}, \mathbf{t}, \mathbf{f}, \mathbf{c}\}$ for $\mathrm{Dcp}_{\mathrm{sl}}$ or $\mathbf{z} \in \{\mathbf{c}, \mathbf{s}, \mathbf{t}\}$ for $\mathrm{Dcp}_{\mathrm{tok}}$) and indicator used ($f \in \{\mathrm{nr}, \cos\}$). We pair, checkpoint by checkpoint, the average of the scalar indicator f across our held-out test set when using either beam-search or forced inference, before computing correlation measures.

Remarkably, we find both Spearman's ρ and Pearson's r to be very highly correlated ($\rho > 0.986$ and $r > 0.901$) in all cases that we test.[4] This extreme correlation indicates that, across training, embeddings derived through beam-search and embeddings derived through forced inference always exhibit the same geometric structures: For instance, if for a given checkpoint, decomposition and term, we observe a low average cosine average between the said terms and the full embeddings as obtained through beam-search, then we are almost certain to obtain a similarly low cosine with forced inference as well. In other words, corpus-level scalar indicators derived from linear decompositions do not appear to be sensitive to which decoding algorithm is used to compute embeddings.

Is geometry indicative of model performance at the corpus level? We have just established that linear decompositions appear stable across different means of decoding. This is broadly compatible with two interpretations: either linear decompositions only capture idiosyncrasies of Transformer geometries; or there are other factors that could influence our scalar indicators. One likely candidate would be model performances: We expect the embeddings of a highly performing model to differ significantly from that of a randomly initialized one or an under-trained one. By extension, differences in quality, as measured through automated metrics, should entail differences in geometry and in scalar indicators derived thereof.

For simplicity, let $\bar{f}(M)$ denote the average of applying function f across our held-out dataset $\mathcal{D}$ using model M, i.e.

$$\bar{f}(M) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} f(M(x))$$

To assess whether differences in geometry and quality are commensurate, we:

i) sample pairs of models M_i, M_{i+1}

ii) compute differences in scalar indicators $\bar{f}_{\mathbf{z}}(M_i) - \bar{f}_{\mathbf{z}}(M_{i+1})$, for $f_{\mathbf{z}} \in \{\mathrm{nr}, \cos\}$ from eqs. (14) and (15);

iii) compute $\bar{f}_s(M_i) - \bar{f}_s(M_{i+1})$ for some scoring function f_s such as BLEU;

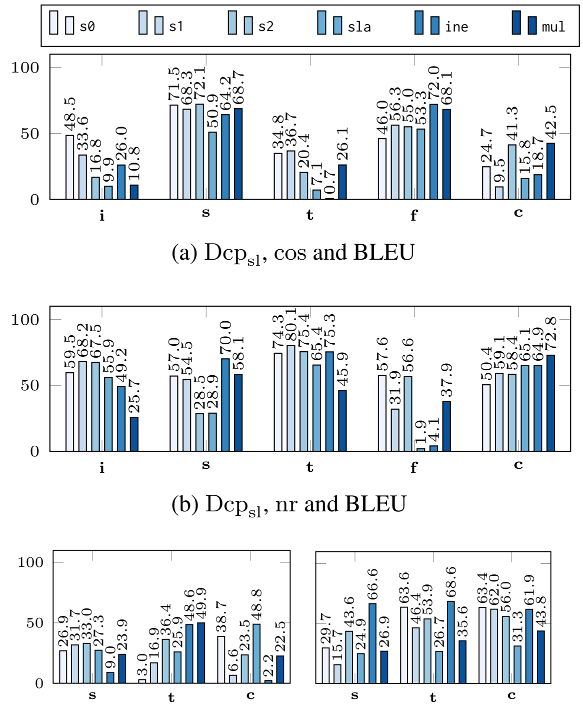
iv) compute the absolute value of Spearman correlation between these two series $|\rho\left(S_{f_{\mathbf{z}}}, S_{f_s}\right)|$[5]

We experiment with BLEU, COMET and chrF++ (Papineni et al., 2002; Rei et al., 2020; Popović, 2017) as scoring functions. Corresponding results for BLEU are presented in fig. 2. We defer results with COMET and chrF++ to appendix C.1, figs. 5 and 6, as they are in line with BLEU. The notations `s0`, `s1` and `s2` refer to our three different runs for Russian-to-English; `sla`, `ine` and `mul` correspond to the Slavic-to-English, Indo-European-to-English and multilingual-to-English model.

There are several trends that we can observe. First, correlation magnitudes tend to be high: This

[4] Only 7 setups yield Pearson correlation coefficient below 0.99, all but one involving the $\mathbf{i}$ term of the $\mathrm{Dcp}_{\mathrm{sl}}$ decomposition: both cosine ($r > 0.944$) and norm-ratio ($r > 0.977$) for the Indo-European-to-English model; the norm-ratio of the multilingual-to-English model ($r > 0.954$); and the cosine for the three Russian-to-English models (with $r > 0.901$, $r > 0.974$ and $r > 0.979$); the lowest of these Russian models also yield $r > 0.968$ for the $\mathbf{t}$ term in $\mathrm{Dcp}_{\mathrm{tok}}$.

[5] This is similar to performing a representational similarity analysis (Kriegeskorte et al., 2008) with the exception that we are looking at signed differences and computing the magnitude of the (anti-) correlations. Our aim is to capture whether scalar indicators and scoring functions are consistent with one another rather than determine what the optimal geometry is. As such, the directionality of a given effect is irrelevant (i.e., we do not care whether the cosine for a specific term has to be low or high for the model to perform well).



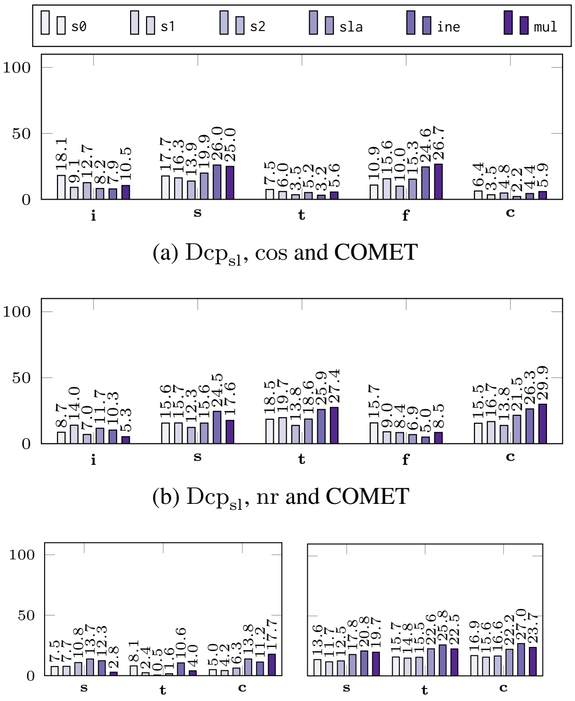
(a) $\mathrm{Dcp}_{\mathrm{sl}}$, cos and BLEU

(b) $\mathrm{Dcp}_{\mathrm{sl}}$, nr and BLEU

(c) $\mathrm{Dcp}_{\mathrm{tok}}$, cos and BLEU (d) $\mathrm{Dcp}_{\mathrm{tok}}$, nr and BLEU

Figure 2: Corpus-level correlation magnitudes (Spearman's $|\rho|$, in %) between scalar indicators (cos, nr) and BLEU. Remark the high variability across models, hinting at a lack of systematicity.

indicates that, on the whole, scalar indicators derived from linear decompositions tend to reflect model quality well (as captured by automatic metrics such as BLEU). Second, and perhaps most importantly, we remark that results across the three seeds for Russian-to-English can display a high degree of variation, both in $\mathrm{Dcp}_{\mathrm{tok}}$ and $\mathrm{Dcp}_{\mathrm{sl}}$—for instance, in $\mathrm{Dcp}_{\mathrm{sl}}$, correlations between changes in cosines and changes in BLEU range from $|\rho| = 9.5$ (in s1) to $|\rho| = 41.3$ (s2) for the **c** term. Third and last, the two decomposition approaches $\mathrm{Dcp}_{\mathrm{tok}}$ and $\mathrm{Dcp}_{\mathrm{sl}}$ suggest different interpretations as to how the decoders behave. For instance, compare target-side input tokens (**t** in $\mathrm{Dcp}_{\mathrm{tok}}$) and target-side self-attention sub-layer outputs (**t** in $\mathrm{Dcp}_{\mathrm{sl}}$): While both terms aim to explain how the target-side input relates to the output embedding, the correlations we derive from the scalar indicators differ between decompositions. We find a surprisingly small correlation magnitude between cos and BLEU for the **t** term under $\mathrm{Dcp}_{\mathrm{sl}}$ in the ine model whereas the s0 model presents the second highest magnitude—but turning to the same measurements for the **t** term under $\mathrm{Dcp}_{\mathrm{tok}}$, we find the exact opposite situation, with s0 being noteworthily lower than all other models, and ine being the second highest.



(a) $\mathrm{Dcp}_{\mathrm{sl}}$, cos and COMET

(b) $\mathrm{Dcp}_{\mathrm{sl}}$, nr and COMET

(c) $\mathrm{Dcp}_{\mathrm{tok}}$, cos and COMET(d) $\mathrm{Dcp}_{\mathrm{tok}}$, nr and COMET

Figure 3: Sentence-level correlation magnitudes (Spearman's $|\rho|$, in %) between scalar indicators (cos, nr) and COMET. Magnitudes are often much lower than their counterparts in fig. 2, suggesting a poorer fit.

In sum, while embedding geometry seems to be shaped in part by how effective a model is—as attested by the often high correlation scores we can observe—the relation between the two is neither straightforward nor systematic across models.

Is geometry indicative of model performance at the sentence level? We have thus far focused on corpus-level measurements. To test whether embedding geometry can provide useful explanations for specific inputs, it is important that we verify whether our observations also hold at the sentence level.

To broach this question, we consider the following methodology: We first select a subset of $k = 3000$ sentences; then for each sentence in said subset, we randomly select two checkpoints per seed. We then compute the correlation magnitude between the signed differences in COMET scores and the signed differences in scalar indicators. [6]

Corresponding results are provided in fig. 3. We

[6] We only focus on COMET as it has been suggested to be more appropriate for sentence-level quality estimation. We also conduct supplementary experiments in appendix C.2 with a slight modification of this methodology.

can make two important remarks. First, we can see that correlation scores are often much lower than what we observed at the corpus level.[7] Nonetheless, some setups still perform reliably well—in particular, norm ratio is found to yield higher correlation magnitudes than cosine n $\mathrm{Dcp}_{\mathrm{tok}}$. This would entail that model quality factors in the results we obtain at the sentence level—if to a lesser extent. Second, we still observe important variation across all three seeds for Russian—often comparable to variation attested across training conditions.

This result suggests that geometry-based explanations are more in line with corpus-level statistics than with sentence-level observations. This naturally questions their usefulness as far as model explainability is concerned, and echoes our previous findings about decoding algorithms: We established that forced inference and beam search did not entail different geometries, we now observe that sentence-level quality is often less appropriate than corpus-level quality when attempting to account for the geometry a model settles on.

Is geometry indicative of training conditions? Throughout our previous experiments, we have seen that variation across our three Russian models was often comparable to variation across different training datasets. We now turn as to whether this fact can be established more firmly: Is there evidence that models that are trained in similar circumstances develop similar geometry? One important aspect of this question consists in assessing the *evolution across training*, rather than focusing on individual checkpoints as we have thus far.

Thus, we now consider the time-series described by our scalar indicators in eqs. (14) and (15) for each term $\mathbf{z}$ of a given decomposition Dcp, through the entire training. For each term and indicator, we compare the time series of all different models using the dynamic time warping algorithm (DTW, Bellman and Kalaba, 1959; Sakoe and Chiba, 1978). Our interest in doing this comparison resides in being able to understand how distant the time series of the different models are between them. The DTW algorithm is especially suitable to our use case, as it measures similarity in a manner that is invariant to shifts and length differences between two time series. In other words, it allows us to measure how similar the series $\mathrm{nr}_{M_1}(\mathbf{z}, \mathbf{e})|_{1,\ldots,N_1}$ and $\mathrm{nr}_{M_2}(\mathbf{z}, \mathbf{e})|_{1,\ldots,N_2}$ are, disregarding the different speed of convergence of both models M_1 and M_2 at training time.

Corresponding results are provided in fig. 4. Each of the heatmap corresponds to the time-series relating to a given term. The upper triangle of each heatmap relates to cosine, and the lower triangle to norm-ratio time series. Individual cells indicate the distance between the time series derived for the models listed in row and column. For instance, the cell in row 2, column 4 of the third plot in fig. 4a corresponds to the distance measured between cosine measurements of the $\mathbf{c}$ term under $\mathrm{Dcp}_{\mathrm{tok}}$ in the `s1` and `s1a` models. Results are z-normalized, as our interest lies in verifying whether Russian models are distinct from other models rather than establish the absolute difference.

The three Russian seeds correspond to the top three rows and columns in each heatmap. A natural expectation would be that comparisons between Russian seeds should lead to more similar time series, and thus lower (z-normalized) distances. Instead, what we observe is consistent with previous experiments: Comparisons between two Russian seeds may or may not yield lower distances. In particular, `s1` and `s2` often yield very distinct time-series, i.e., the models develop very different geometries despite their similar training conditions.

	term				
	i	**s**	**t**	**f**	**c**
cos	0.002	0.367	0.352	0.108	0.022
nr	0.020	0.002	0.002	0.002	0.297

(a) $\mathrm{Dcp}_{\mathrm{sl}}$

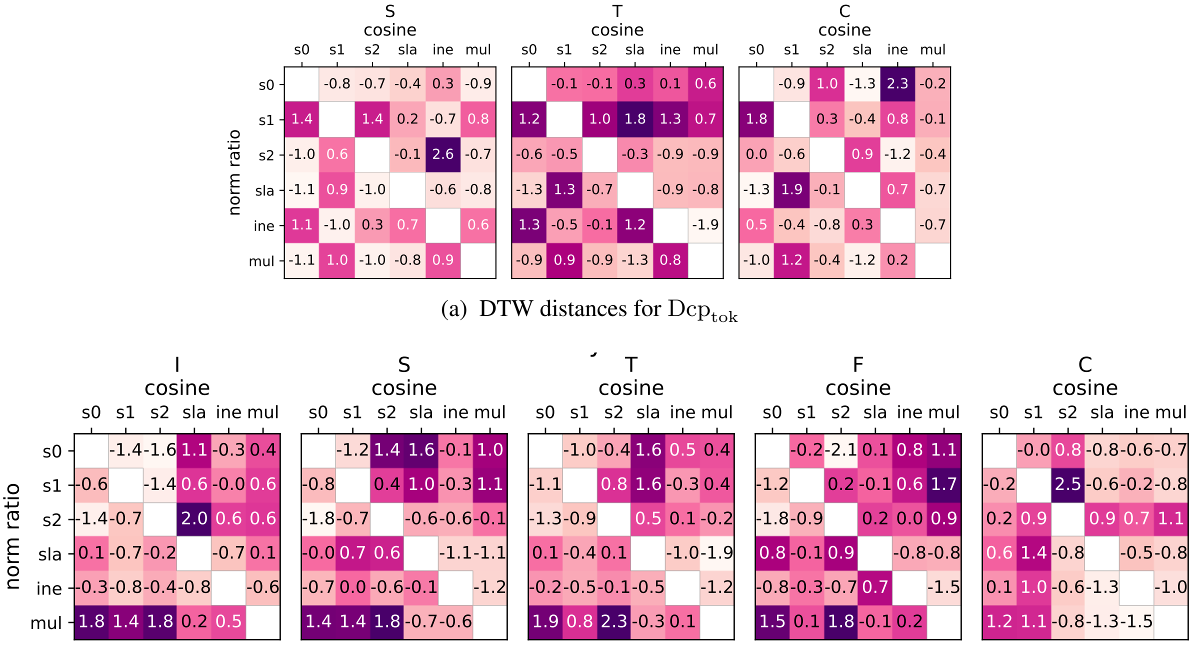
	term		
	s	**t**	**c**
cos	0.503	0.316	0.380
nr	0.222	0.422	0.231

(b) $\mathrm{Dcp}_{\mathrm{tok}}$

Table 2: p-values derived from Pitman permutation tests

To provide a more thorough outlook on this question, we conduct Pitman permutation tests (Dror et al., 2018) to establish whether comparisons between two Russian models are statistically lower than others. Corresponding results are provided in table 2. As we can see, while select setups using $\mathrm{Dcp}_{\mathrm{sl}}$ yield p-values beyond the commonly used 0.05 threshold, only half of the setup we experiment with yield the expected result. In particular,

[7] In fact the p-value provided by `scipy` for these correlation scores suggests that many of these correlations are spurious.



(a) DTW distances for $\mathrm{Dcp}_{\mathrm{tok}}$

(b) DTW distances for $\mathrm{Dcp}_{\mathrm{sl}}$

Figure 4: Dynamic time warping distance measurements, z-normalized. Remark that distances between seed replications (s0, s1, s0) do not differ from distances between models with different inputs.

all setups based on $\mathrm{Dcp}_{\mathrm{tok}}$ are insignificant.

We therefore conclude that different decomposition approaches lead to different interpretations of what Transformer geometry encodes. Had we only focused on $\mathrm{Dcp}_{\mathrm{tok}}$, we would have been lead to a much firmer rejection of the notion that decompositions are stable across random initializations. The inclusion of $\mathrm{Dcp}_{\mathrm{sl}}$ in our experiments forces us to adopt a more nuanced approach: viz., that the evidence in favor of geometry-based explainability approaches is thin; and that results derived from such approaches appear very brittle—the exact methodology used brings about variations in p-value of up to two orders of magnitude.

5 Conclusions

We have presented a series of statistical studies questioning the usefulness of linear decomposition approaches. In particular, we have highlighted that straightforward vector space characteristics, such as angle and norm of the derived vector terms, imply the following three points: (i) decompositions are invariant to the decoding algorithm employed; (ii) they are more in line with corpus-level performance than sentence-level performance, and (iii) variance across random seeds for the same training conditions is often comparable to variance across models trained on different corpora. Taken together, our experiments suggest that Transformer geometry is often highly model-specific. Observations about a specific model need not generalize.

As such, some of the assumptions underlying geometry-based explanations of Transformer behaviors are not borne out. While it is true that the geometry of successful models differs from that of unsuccessful ones, our work puts forth evidence that this difference is mostly trivial—geometry being model-specific necessarily entails that any partition of models, be it based on performance or else, will naturally highlight differences.

While our focus has been limited to linear decompositions and straightforward vector characteristics, our experiments more broadly call into question the validity of many related approaches, which we hope to investigate in future work. That straightforward vector characteristics do not yield a coherent picture *a minima* entails that linear decomposition approaches have to rely on non-straightforward, high-dimensional relationships. That similar training conditions cannot guarantee similar vector spaces naturally leads us to doubt the generalization power of methodologies that probe a handful of foundational models: If we are unable to ensure that our approaches would generalize to other similar models, can we truthfully say that the explanations we provide are indeed reasonable?

Acknowledgments

We thank Hande Celikkanat, Denis Paperno and the three anonymous reviewers for their insightful comments, as well as Jörg Tiedemann for invaluable help with data selection.

This work is part of the FoTran project, funded by the European Research Council (ERC) under the EU's Horizon 2020 research and innovation program (agreement № 771113). We also thank the CSC-IT Center for Science Ltd., for computational resources.

References

Carl Allen and Timothy Hospedales. 2019. Analogies explained: Towards understanding word embeddings. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 223–231. PMLR.

R. Bellman and R. Kalaba. 1959. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2023. Analyzing transformers in embedding space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16124–16170, Toronto, Canada. Association for Computational Linguistics.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

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

Javier Ferrando, Gerard I. Gállego, Belen Alastruey, Carlos Escolano, and Marta R. Costa-jussà. 2022a. Towards opening the black box of neural machine translation: Source and target interpretations of the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8756–8769, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, and Marta R. Costa-jussà. 2022b. Measuring the mixing of contextual information in the transformer. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8698–8714, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Javier Ferrando, Gerard I. Gállego, Ioannis Tsiamas, and Marta R. Costa-jussà. 2023. Explaining how transformers use context to build predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5486–5513, Toronto, Canada. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2020. Attention is not only a weight: Analyzing transformers with vector norms. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7057–7075, Online. Association for Computational Linguistics.

Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. 2008. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2.

Timothee Mickus, Denis Paperno, and Mathieu Constant. 2022. How to dissect a Muppet: The structure of transformer embedding spaces. *Transactions of the Association for Computational Linguistics*, 10:981–996.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. 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.

Ali Modarressi, Mohsen Fayyaz, Yadollah Yaghoobzadeh, and Mohammad Taher Pilehvar. 2022. GlobEnc: Quantifying global token attribution by incorporating the whole encoder layer in transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 258–271, Seattle, United States. Association for Computational Linguistics.

Byung-Doh Oh and William Schuler. 2023. Token-wise decomposition of autoregressive language model hidden states for analyzing model predictions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10105–10117, Toronto, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2020. Learning to deceive with attention-based explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4782–4793, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy. Association for Computational Linguistics.

Amit Singhal. 2001. Modern information retrieval: A brief overview. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 24(4):35–43.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Raúl Vázquez, Hande Celikkanat, Vinit Ravishankar, Mathias Creutz, and Jörg Tiedemann. 2022. A closer look at parameter contributions when training neural language and translation models. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4788–4800, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2021. Analyzing the source and target contributions to predictions in neural machine translation. 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 1126–1140, Online. Association for Computational Linguistics.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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 11–20, Hong Kong, China. Association for Computational Linguistics.

Sen Yang, Shujian Huang, Wei Zou, Jianbing Zhang, Xinyu Dai, and Jiajun Chen. 2023. Local interpretation of transformer based on linear decomposition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10270–10287, Toronto, Canada. Association for Computational Linguistics.

Sumu Zhao, Damián Pascual, Gino Brunner, and Roger Wattenhofer. 2021. Of non-linearity and commutativity in bert. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Slavic	bel bos bul ces cnr csb dsb hbs hrv hsb mkd pol slk slv srp szl ukr
Indo-European	afr anp arg asm ast bar ben bis bre bzj cat ckb cor cos crs cym dan deu div djk dty ell fao fas fra frp fry fur gla gle glg glv guj hat hin hne hye hyw ind isl ita jak kas kea kmr kri kur lad lav lij lim lit lmo ltg ltz mai mar mfe min mol msa mwl nds nep nld nno nob oci ori oss pan pap pdt pes pis pob por prs pus rmn rmy roh rom ron san scn sco sin spa sqi srd srm srn swe tgk tpi urd vec wae wes wln yid zlm
multilingual	abk ace ach ada aka alt alz amh ami ara arq ary arz ava aze azz bak bas bbc bci bcl bem bhw bin bod brx bts btx bug bum cab cak ceb cha chk chr chv cjk cmn cnh cop crh ctu dhv dik din dje dua dyu dzo efi epo est eus ewe fas fij fil fin fon ful fuv gil grn guc gug guw gym heb her hil hmn hne hun iba ibg ibo ido ilo ish iso ixl jav jbo jpn kab kac kal kam kan kat kau kaz kbp kek khm kik kin kmb kon koo kqn kss ksw kua kwn lam lao lfn lin loz lua lub lue lug lun luo lus lzh mah mal mam mau meh men mgr mhr mlg mlt mon mos mri mrj mxv mya nan naq nav nba nbl nch ncj ncx ndc nde ndo ngl ngu nia nij niu nso nya nyk nyn nyu nzi oke orm pag plt pon quc rar rnd run sag sat seh ses sid sme smo sna som sop sot ssw sun swa sxn syr tah tam tat tcf tdt tel tgl tha tir tiv tll tmh tog toh toi toj ton trs tsc tsn tso ttj tuk tum tur tvl twi tyv tzh tzo udm uig umb urh uzb ven vie vmw wal war wls wol wuu xho xmf yao yap yor yua yue zai zam zne zpa zul

Table 3: List of language sources for multilingual models. More multilingual sources also contain languages from less multilingual models. All models also contain Russian.

A Model training details

As noted in the main text, we use the Tatoeba Challenge corpus (Tiedemann, 2020) and the marian-MT library (Junczys-Dowmunt et al., 2018). Models were trained using four V100 nVidia GPUs.

Models use sources of different degrees of multilinguality: multilingual-to-English (`mul`); Indo-European-to-English (`ine`); Slavic-to-English (`sla`) and three different seeds for Russian-to-English (`s0`, `s1` and `s2`). All languages included in a more specific model are also present in all more multilingual models. For instance, there are datapoints in the `mul` model's training data for each of the Slavic languages used to train the `sla` model. A complete list of the languages used for multilingual models in this study can be found in table 3; all models also contain Russian.

Detailed hyperparameters is provided in table 4. We refer the reader to Junczys-Dowmunt et al. (2018) and the associated documentation[8] for further explanations. Models `s0`, `sla`, `ine` and `mul` used the first of the three listed seeds, whereas `s1` used the second and `s2` the third. In practice, none of the six models fulfilled the early stopping criterion in the allocated runtime (72h).

B Supplementary details on decompositions

Notation details. table 1 lists the notations used throughout this work. Remark that, aside from row-selection (marked $(\mathbf{Z})_i$), symbol typesetting indicates the type of mathematical object denoted: i.e., $a_{\lambda htt'}$ is a scalar and not a tensor of rank 4.

Presentation of the Transformer decoder architecture. The remainder of this appendix consists in a general introduction to a Transformer decoder architecture. We refer the reader to Vaswani et al. (2017) for a more thorough overview.

A Transformer decoder is a stack of L layers, each containing 3 sub-layers. Sub-layers are defined by means of specific sub-layer components: either *multi-head attention mechanisms* (ma) or *feed-forwards* (ff).

The latter are multi-layer perceptrons of the form:

$$\dot{\mathbf{e}}_{\lambda,t} = \mathbf{W}_{\lambda}^{(\text{ff,out})}\phi\left(\mathbf{W}_{\lambda}^{(\text{ff,in})}\mathbf{x}_{\lambda,t} + \mathbf{b}_{\lambda}^{(\text{ff,in})}\right) + \mathbf{b}_{\lambda}^{(\text{ff,}out)}$$

where ϕ is a non-linear activation function (e.g., ReLU, SiLU, GELU...).

Multi-head attention mechanisms consist in attention-based weighted average computations:

$$\dot{\mathbf{E}}_{\lambda} = \mathbf{W}_{\lambda}^{(\text{ma,O})} \bigoplus_{h=1}^{H} \mathbf{A}_{\lambda,h}\left(\mathbf{W}_{\lambda,h}^{(\text{ma,V})}\mathbf{X} + \mathbf{b}_{\lambda,h}^{(\text{ma,V})}\right) + \mathbf{b}_{\lambda}^{(\text{ma,O})}$$

where the input $\mathbf{X}$ is either the previous sub-layer output, up to token t included (i.e., $\begin{bmatrix}\mathbf{x}_{\lambda,1}\\ \vdots \\ \mathbf{x}_{\lambda,t}\end{bmatrix}$) or the output of the Transformer encoder ($\mathbf{X}_{\text{enc}}$). The attention weights $\mathbf{A}_{\lambda,h}$ are computed as:

$$\begin{aligned}
\mathbf{A}_{\lambda,h} &= \operatorname{softmax}\left(\frac{\mathbf{Q}_{\lambda,h}\mathbf{K}_{\lambda,h}^{\top}}{\sqrt{d/H}}\right) \\
\mathbf{Q}_{\lambda,h} &= \mathbf{W}_{\lambda,h}^{(\text{ma,Q})}\mathbf{X}_{\lambda} + \mathbf{b}_{\lambda,h}^{(\text{ma,Q})} \\
\mathbf{K}_{\lambda,h} &= \mathbf{W}_{\lambda,h}^{(\text{ma,K})}\mathbf{X} + \mathbf{b}_{\lambda,h}^{(\text{ma,K})}
\end{aligned}$$

[8] https://marian-nmt.github.io/docs/

H-param.	Value
type	transformer
quiet-translation	true
max-length	500
mini-batch-fit	true
workspace	24000
maxi-batch	500
valid-mini-batch	16
valid-freq	5000
save-freq	1000
disp-freq	5000
valid-metrics	perplexity cross-entropy bleu chrf
beam-size	12
normalize	1
allow-unk	true
enc-depth	6
dec-depth	6
transformer-heads	8
transformer-postprocess-emb	d
transformer-postprocess	dan
transformer-ffn-activation	swish
transformer-dropout	0.1
label-smoothing	0.1
learn-rate	0.0003
lr-warmup	16000
lr-decay-inv-sqrt	16000
lr-report	true
optimizer-params	0.9 0.98 1e-09
clip-norm	5
fp16	true
tied-embeddings-all	true
early-stopping	150
cost-type	ce-mean
exponential-smoothing	true
devices	0 1 2 3
sync-sgd	true
seed	1111 1989 20232

Table 4: Hyperparameters for models

Remark that the matrix $\mathbf{A}_{\lambda,h}$ has size $q \times k$, with q the number of rows in $\mathbf{X}_\lambda$ and k the number of rows in the input $\mathbf{X}$. Specific cell values $a_{\lambda hij}$ of this attention matrix $\mathbf{A}_{\lambda,h}$, also known as *attention weights*, can be seen as computing a similarity score for the i^{th} (linearly transformed) input contextual embedding and the j^{th} (linearly transformed) attended vector.

Lastly, around each sub-layer, a *residual connection* and a *layer-norm* are applied:

$$\mathbf{e}_{\lambda,t} = \mathbf{g}_\lambda^{(\ln)} \odot \frac{\ddot{\mathbf{e}}_{\lambda,t} - m_{\lambda,t}\vec{1}}{s_{\lambda,t}} + \mathbf{b}_\lambda^{(\ln)}$$

$$\ddot{\mathbf{e}}_{\lambda,t} = \dot{\mathbf{e}}_{\lambda,t} + \mathbf{x}_{\lambda,t}$$

where $m_{\lambda,t}$ is the mean of the components of $\ddot{\mathbf{e}}_{\lambda,t}$, and $s_{\lambda,t}$ the corresponding standard deviation.

C Supplementary results

Numerical stability. Acros all experiments, all decomposed embeddings were tested for numerical stability: We ensure an absolute tolerance of $\mathrm{tol}_a = 10^{-8}$ and a relative tolerance of $\mathrm{tol}_r = 10^{-5}$, or more formally that the following is true:

$$\forall \mathrm{Dcp}\ \forall \mathbf{e} \quad \left| \mathbf{e} - \sum_{\mathbf{z} \in \mathrm{Dcp}(\mathbf{e})} \mathbf{z} \right| \leq \mathrm{tol}_a + \mathrm{tol}_r \left| \sum_{\mathbf{z} \in \mathrm{Dcp}(\mathbf{e})} \mathbf{z} \right|$$

In practice, doing so requires 64-bit float precision, despite the models having been trained with fp16.

C.1 Performance at the corpus level

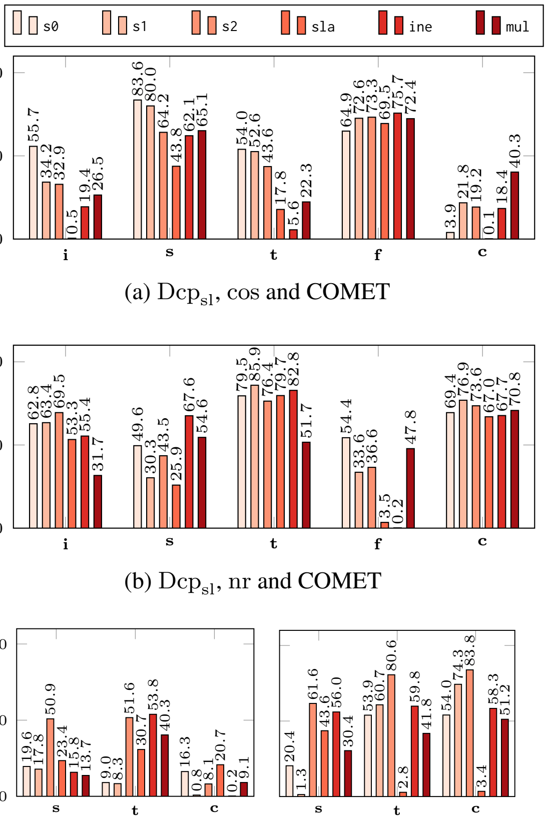
In addition to the BLEU results presented in the main text, we also compute correlation magnitudes using COMET and chrF++ as scoring functions. Corresponding results are presented in fig. 5 and fig. 6.

Overall, results are similar to what we observed with BLEU in fig. 2: Setups that yield low or insignificant correlation magnitudes do so across scoring functions. We nonetheless also attest variation across the different scoring functions, as some specific setups can switch by $\approx 10\%$ depending on the scoring function.

C.2 Performance at the sentence level

In the main body of this article, we measure correlations of sentence-level performance and scalair indicators. One debatable methodological choice is that we decide to compute signed differences for observations corresponding to the same sentence.

On the one hand, this allows us to factor out some intrinsic variation in scalar indicators that

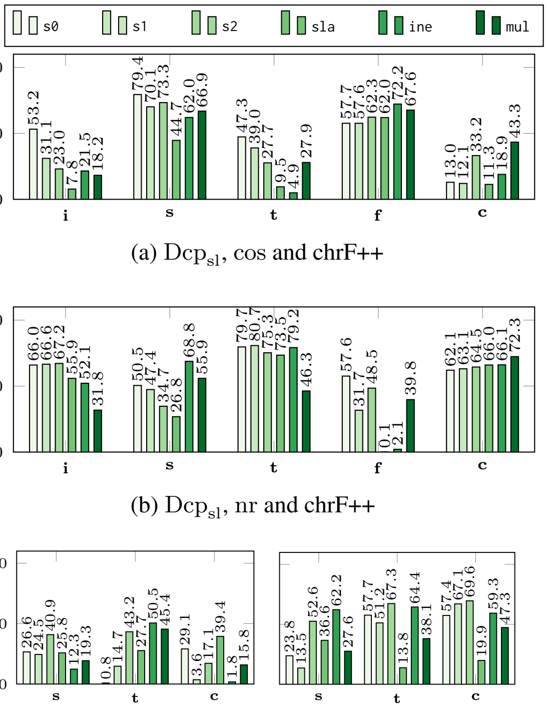


(a) $\mathrm{Dcp}_{\mathrm{sl}}$, cos and COMET

(b) $\mathrm{Dcp}_{\mathrm{sl}}$, nr and COMET

(c) $\mathrm{Dcp}_{\mathrm{tok}}$, cos and COMET(d) $\mathrm{Dcp}_{\mathrm{tok}}$, nr and COMET

Figure 5: Corpus-level correlation magnitudes (Spearman's $|\rho|$, in %) between scalar indicators (cos, nr) and COMET.

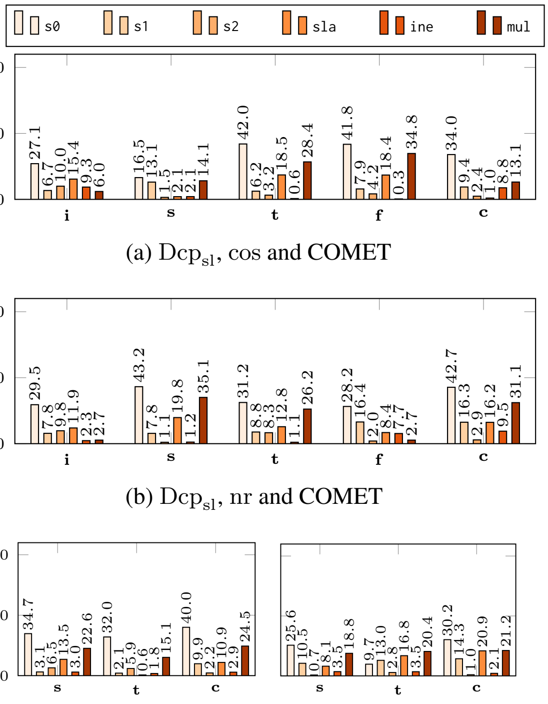


(a) $\mathrm{Dcp}_{\mathrm{sl}}$, cos and chrF++

(b) $\mathrm{Dcp}_{\mathrm{sl}}$, nr and chrF++

(c) $\mathrm{Dcp}_{\mathrm{tok}}$, cos and chrF++ (d) $\mathrm{Dcp}_{\mathrm{tok}}$, nr and chrF++

Figure 6: Corpus-level correlation magnitudes (Spearman's $|\rho|$, in %) between scalar indicators (cos, nr) and chrF++.



(a) $\mathrm{Dcp}_{\mathrm{sl}}$, cos and COMET

(b) $\mathrm{Dcp}_{\mathrm{sl}}$, nr and COMET

(c) $\mathrm{Dcp}_{\mathrm{tok}}$, cos and COMET(d) $\mathrm{Dcp}_{\mathrm{tok}}$, nr and COMET

Figure 7: Sentence-level correlation magnitudes (Spearman's $|\rho|$, in %) between scalar indicators (cos, nr) and COMET, without sentence-level pairing.

we expect to arise from sheer difference of inputs: Differences owed to sentence length, idiomaticity, and so on might influence observations—which is why the main results we present do control for input.

On the other hand, one can argue that some inputs will be inherently poorly handled by a model, regardless of its geometry, simply due to training conditions. Consider for instance a model that would have been solely trained on a bi-text derived from subtitles: Its performances on data derived from parliamentary debates will likely remain low regardless of whether it converges on an efficient set of parameters for its training data. More succinctly put, one can argue that distributional shifts may impact a sentence-paired approach such as the one we proposed earlier.

We therefore present in fig. 7 correlation magnitudes derived on unpaired inputs—i.e., we sample two sentences and two checkpoints at random, and compute the corresponding absolute value of the correlation between signed differences. One can broadly observe two facts: First, correlation magnitudes are indeed often higher than what we previously reported in fig. 3—however, do recall that one can argue that more variance is expected as we do not control for input variations. Second, and perhaps more interestingly, we see that whether high correlation magnitudes emerge or not appears highly specific to a given model: in particular, `s0` and `mul` almost systematically yields very high correlation magnitudes, whereas other models tend to produce often insignificant scores.

Overall, this supplementary experiment offers an interesting angle: We find evidence that some models' geometry can reflect sentence-level performance, but this does not generalize across different random initializations under the same training conditions.