



Article

Fractal Self-Similarity in Semantic Convergence: Gradient of Embedding Similarity across Transformer Layers

Minhyeok Lee

School of Electrical and Electronics Engineering, Chung-Ang University, Seoul 06974, Republic of Korea; mlee@cau.ac.kr

Abstract: This paper presents a mathematical analysis of semantic convergence in transformer-based language models, drawing inspiration from the concept of fractal self-similarity. We introduce and prove a novel theorem characterizing the gradient of embedding similarity across layers. Specifically, we establish that there exists a monotonically increasing function that provides a lower bound on the rate at which the average cosine similarity between token embeddings at consecutive layers and the final layer increases. This establishes a fundamental property: semantic alignment of token representations consistently increases through the network, exhibiting a pattern of progressive refinement, analogous to fractal self-similarity. The key challenge addressed is the quantification and generalization of semantic convergence across diverse model architectures and input contexts. To validate our findings, we conduct experiments on BERT and DistilBERT models, analyzing embedding similarities for diverse input types. While our experiments are limited to these models, we empirically demonstrate consistent semantic convergence within these architectures. Quantitatively, we find that the average rates of semantic convergence are approximately 0.0826 for BERT and 0.1855 for DistilBERT. We observe that the rate of convergence varies based on token frequency and model depth, with rare words showing slightly higher similarities (differences of approximately 0.0167 for BERT and 0.0120 for DistilBERT). This work advances our understanding of transformer models' internal mechanisms and provides a mathematical framework for comparing and optimizing model architectures.



Citation: Lee, M. Fractal Self-Similarity in Semantic Convergence: Gradient of Embedding Similarity across Transformer Layers. *Fractal Fract.* **2024**, *8*, 552. <https://doi.org/10.3390/fractalfract8100552>

Academic Editors: Marcin Sosnowski, Jaroslaw Krzywanski, Karolina Grabowska, Dorian Skrobek and Ghulam Moeen Uddin

Received: 25 August 2024

Revised: 16 September 2024

Accepted: 18 September 2024

Published: 24 September 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: transformer models; semantic convergence; embedding similarity; gradient analysis; BERT; DistilBERT; natural language processing; fractal-inspired analysis; layer-wise dynamics; word frequency analysis

1. Introduction

Transformer-based language models have revolutionized natural language processing, demonstrating unprecedented performance across various tasks [1,2]. These models, characterized by their multi-layered architecture and self-attention mechanisms, have become the cornerstone of modern NLP applications. However, despite their widespread adoption and empirical success, the internal mechanisms that drive their performance still need to be more adequately understood [3,4]. This gap in our understanding presents a significant obstacle to further advancements in the field and developing more efficient architectures.

Previous research has made significant strides in analyzing various aspects of transformer models [5]. Voita et al. [6] examined the evolution of representations across transformer layers, revealing that different attention heads specialize in distinct linguistic tasks. Tenney et al. [7] employed probing tasks to analyze BERT's layer-wise linguistic capabilities, proposing that BERT's layers mirror the traditional NLP pipeline. However, these studies, while invaluable, have primarily focused on specific components or linguistic features, lacking a comprehensive mathematical framework to describe the overall semantic convergence process across layers [8,9].

The primary objective of this paper is to establish a mathematical framework for analyzing the semantic convergence of token embeddings across transformer layers. Let M be a transformer model with L layers, and let $e_t^i \in \mathbb{R}^d$ denote the embedding of token t at layer i , where d is the embedding dimension. We define the semantic alignment $S(i)$ of layer i with respect to the final layer L as: $S(i) = \frac{1}{|V|} \sum_{t \in V} \cos(e_t^i, e_t^L)$, where V is the vocabulary and $\cos(\cdot, \cdot)$ denotes the cosine similarity. We aim to elucidate the fundamental processes underlying these models' ability to capture and manipulate semantic information by quantifying how these representations evolve through the network. This understanding is theoretically significant and has profound implications for model design and optimization.

The principal challenge in this endeavor lies in formulating a generalizable theory that holds across diverse model architectures and input contexts [10]. The inherent complexity and high dimensionality of transformer models render it difficult to derive universal principles that accurately describe their behavior. Let $T_i : \mathbb{R}^{n \times d} \rightarrow \mathbb{R}^{n \times d}$ denote the transformation applied by the i -th layer of a transformer model, where n is the sequence length. The challenge is to characterize the properties of the composition $T_L \circ T_{L-1} \circ \dots \circ T_1$ in terms of semantic convergence.

To address these challenges, we introduce and prove a novel theorem characterizing the gradient of embedding similarity across layers in transformer models. The proposed theorem provides a quantitative measure of how token representations converge towards their final form as they progress through the network, offering a lower bound on the rate of semantic alignment increase between consecutive layers. The implications of this work extend beyond theoretical interest. By providing a mathematical framework for analyzing the layer-wise behavior of transformer models, we offer a new tool for comparing and optimizing model architectures.

In the following sections, we present our theoretical framework, prove our main theorem, and provide empirical validation through comprehensive experiments on BERT and DistilBERT models. Our analysis covers various aspects of semantic convergence, including the impact of sentence complexity and word frequency, offering insights into how different linguistic features are processed through the layers of transformer models. We examine the properties of the function $f(i)$ and its relationship to model architecture, laying the groundwork for a deeper understanding of the mathematical principles underlying transformer-based language models.

Beyond theoretical interest, our findings have significant practical implications for NLP practitioners. By establishing a mathematical framework that describes how semantic information is progressively refined across transformer layers, we provide valuable insights that can inform the design and optimization of transformer-based architectures. Understanding the semantic convergence process can guide practitioners in tasks such as model pruning, layer selection for fine-tuning, and developing more efficient architectures that retain performance while reducing computational costs. This framework also offers a diagnostic tool for analyzing model behavior and identifying layers that contribute most to semantic understanding, thereby aiding in model interpretability and transparency.

2. Background

2.1. Transformer Architecture

Transformer models have significantly advanced the field of natural language processing by efficiently handling sequential data. Introduced by Vaswani et al. [11], transformers utilize a mechanism called self-attention, which allows the model to focus on different parts of the input sequence when generating a representation for each word. This means that the model can capture relationships between words regardless of their position in the sequence, enabling it to understand long-range dependencies in language.

The self-attention mechanism works by assigning weights to each word in the input sequence, indicating its importance relative to other words. This approach has been extensively studied, and has proven effective in capturing the overall meaning of sentences [12,13]. Researchers have explored various aspects of self-attention, such as how

it can learn to focus on important features in the data [13], and how to simplify it for improved computational efficiency [14].

Transformers are composed of multiple layers, and each layer includes two main components: the self-attention mechanism and a feed-forward neural network. The self-attention component allows the model to weigh the significance of different words when processing the input, while the feed-forward network further transforms these representations to capture higher-level features. By stacking these layers, transformers can build increasingly complex representations of the input data.

Some studies have compared transformers to recurrent neural networks (RNNs), noting that transformers can be viewed as a type of RNN that processes all positions in the sequence simultaneously, rather than sequentially [15]. This parallel processing capability makes transformers more efficient and better suited for handling long sequences than traditional RNNs.

Definition 1 (Transformer layer). A transformer layer T_i consists of two sublayers applied sequentially to the input sequence $X_i \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the embedding dimension:

Self-Attention Sublayer with Residual Connection:

$$Y_i = X_i + \text{MultiHead}(\text{LayerNorm}(X_i)) \quad (1)$$

Feed-Forward Sublayer with Residual Connection:

$$X_{i+1} = Y_i + \text{FFN}(\text{LayerNorm}(Y_i)) \quad (2)$$

In this formulation, $X_{i+1} \in \mathbb{R}^{n \times d}$ is the output sequence of the transformer layer, $\text{MultiHead}(\cdot)$ denotes the multi-head self-attention mechanism, $\text{LayerNorm}(\cdot)$ is layer normalization, and $\text{FFN}(\cdot)$ is a position-wise feed-forward network. The residual connections are added after each sublayer's output, and layer normalization is applied before each sublayer's computation. This structure aligns with the Pre-LayerNorm transformer architecture commonly used in practice.

2.2. Token Embeddings and Semantic Representation

Token embeddings are dense vector representations of discrete tokens (e.g., words or subwords) in a continuous vector space. In transformer models, these embeddings evolve through the layers, gradually capturing more complex and abstract semantic information.

Theorem 1 (Embedding transformation). Let $e_t^i \in \mathbb{R}^d$ be the embedding of token t at layer i . The transformation of this embedding through a transformer layer can be expressed as

$$e_t^{i+1} = T_i(e_t^i) + \sum_{j \neq t} \alpha_{tj} T_i(e_j^i) \quad (3)$$

where α_{tj} are attention weights and the sum represents the contextual information from other tokens.

2.3. Cosine Similarity as a Measure of Semantic Alignment

Cosine similarity serves as a fundamental metric for quantifying semantic alignment between vector representations in natural language processing. For two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, the cosine similarity is defined as follows.

Definition 2 (Cosine similarity). The cosine similarity between two non-zero vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ is defined as

$$\cos(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\sum_{i=1}^d a_i b_i}{\sqrt{\sum_{i=1}^d a_i^2} \sqrt{\sum_{i=1}^d b_i^2}} \quad (4)$$

This measure captures the angular similarity between vectors, providing a normalized metric invariant to the magnitude of the vectors. The cosine similarity has several important properties that make it particularly suitable for analyzing semantic alignment in transformer models.

In our analysis, we use the cosine similarity as defined in Equation (3) without taking the absolute value. This decision is based on the significance of vector directionality in semantic embedding spaces. A cosine similarity of 1 indicates that the vectors are perfectly aligned and convey similar semantic content, while a cosine similarity of -1 implies that the vectors are diametrically opposed, representing contrasting or unrelated meanings. By retaining the sign of the cosine similarity, we preserve critical information about the relationship between token embeddings. Taking the absolute value would obscure this distinction, treating opposing meanings as similar, which is undesirable in the context of semantic analysis. Therefore, using the cosine similarity without the absolute value allows us to accurately capture the nuances of semantic alignment and dissimilarity between token embeddings.

Theorem 2 (Properties of cosine similarity). *For any two non-zero vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, the cosine similarity satisfies the following properties:*

1. *Boundedness:* $-1 \leq \cos(\mathbf{a}, \mathbf{b}) \leq 1$.
2. *Symmetry:* $\cos(\mathbf{a}, \mathbf{b}) = \cos(\mathbf{b}, \mathbf{a})$.
3. *Invariance to scalar multiplication:* $\cos(c\mathbf{a}, \mathbf{b}) = \cos(\mathbf{a}, c\mathbf{b}) = \cos(\mathbf{a}, \mathbf{b})$ for any non-zero scalar c .
4. *Cosine distance:* $d_{\cos}(\mathbf{a}, \mathbf{b}) = 1 - \cos(\mathbf{a}, \mathbf{b})$ is a proper distance metric on the unit sphere.

Proof. The proof of these properties follows from the definition of cosine similarity and basic vector algebra:

1. **Boundedness:** This follows from the Cauchy–Schwarz inequality:

$$|\mathbf{a} \cdot \mathbf{b}| \leq \|\mathbf{a}\| \|\mathbf{b}\| \quad (5)$$

Dividing both sides by $\|\mathbf{a}\| \|\mathbf{b}\|$ yields the result.

2. **Symmetry:** this is evident from the commutativity of the dot product.
3. **Invariance to scalar multiplication:** This follows from the properties of dot product and vector norms:

$$\cos(c\mathbf{a}, \mathbf{b}) = \frac{c\mathbf{a} \cdot \mathbf{b}}{\|c\mathbf{a}\| \|\mathbf{b}\|} = \frac{c(\mathbf{a} \cdot \mathbf{b})}{|c| \|\mathbf{a}\| \|\mathbf{b}\|} = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|} = \cos(\mathbf{a}, \mathbf{b}) \quad (6)$$

4. **Cosine distance:** the proof that $d_{\cos}$ is a proper distance metric involves verifying the four metric axioms (non-negativity, identity of indiscernibles, symmetry, and triangle inequality) for vectors on the unit sphere.

□

2.4. Fractal Mathematics and Self-Similarity

Fractal mathematics deals with structures that exhibit self-similarity across different scales, meaning that the structure appears similar regardless of the level of magnification [16]. A classic example is the Mandelbrot set, where zooming into the boundary reveals infinitely many repetitions of the overall shape. In mathematical terms, a fractal is a set that displays self-similarity under some scaling transformation.

In the context of transformer models, the repeated application of similar layer transformations to token embeddings can be seen as analogous to the iterative processes in fractal generation. Each layer refines the representations in a manner that preserves certain structural properties while introducing new details, much like how fractals evolve under

iteration. This self-similarity suggests that concepts from fractal mathematics may provide a suitable framework for analyzing the convergence properties of these models.

By viewing the semantic convergence through the lens of fractal self-similarity, we can employ mathematical tools from fractal analysis to quantify the rate and nature of convergence. For instance, the concept of fractal dimension could inspire metrics for measuring the complexity of embedding spaces across layers. This perspective enriches our theoretical framework and aligns with the observed patterns of progressive refinement in transformer models.

The semantic convergence process in transformer models exhibits intriguing parallels with concepts from fractal mathematics [17]. As we further explore the layers of these models, we observe a self-similar pattern in the way token representations refine and converge, akin to the recursive and iterative processes found in fractal structures. In fractal geometry, complex patterns are generated by repeating a simple process at different scales, resulting in structures that are self-similar across scales [16]. Similarly, transformer models apply the same layer transformations repeatedly, leading to representations that exhibit patterns of refinement that are consistent across layers. This fractal-inspired perspective is appropriate for our analysis because it captures the essence of hierarchical processing and recursive refinement inherent in transformer architectures [18]. By leveraging mathematical concepts from fractal geometry, we can quantify the progressive convergence of token embeddings and provide a framework that mirrors the self-similar nature of the models' internal dynamics. This approach not only enhances our theoretical understanding, but also offers practical tools for analyzing and optimizing transformer models.

3. Related Work

3.1. Analysis of Transformer Layer Dynamics

The study of transformer layer dynamics has been a focal point in understanding the internal mechanisms of these models. Voita et al. [6] pioneered this field by examining the evolution of representations across transformer layers. Their work revealed that different attention heads specialize in distinct linguistic tasks, a finding that contrasts with our approach of analyzing the overall semantic convergence. While Voita et al. focused on individual attention heads, our method provides a holistic view of semantic evolution across layers.

Tenney et al. [7] extended this line of inquiry by employing probing tasks to analyze BERT's layer-wise linguistic capabilities. They proposed that BERT's layers mirror the traditional NLP pipeline, a hypothesis that aligns with our notion of gradual semantic refinement. However, our work differs in its mathematical formalization of this process, offering a quantitative measure of semantic convergence through the gradient of embedding similarity. The findings of Tenney et al. have been later challenged by Niu et al. [19], who argued for a more nuanced interpretation of BERT's layer-wise behavior. This debate underscores the complexity of transformer dynamics, and motivates our approach of seeking a universal principle of semantic convergence that holds across different architectures.

Recent studies have further diversified the analysis of transformer layers. Kakouros et al. [20] investigated BERT's encoding of prosodic information, finding a concentration in middle layers. Similarly, Zheng et al. [21] examined syntactic knowledge in Chinese BERT. These works highlight the multifaceted nature of information processing in transformers, which our gradient analysis aims to capture in a unified framework.

3.2. Semantic Representation in Language Models

The analysis of semantic representations in language models has been a topic of significant interest in recent years. Peters et al. [22] introduced ELMo, demonstrating the power of deep contextualized word representations. Their work laid the foundation for understanding how neural language models capture semantic information across different layers. The ELMo model can be formalized as

$$\text{ELMo}_k = \gamma \sum_{j=0}^L s_j h_{k,j} \quad (7)$$

where $h_{k,j}$ is the contextualized representation of the k -th token at the j -th layer, s_j are softmax-normalized weights, and γ is a scalar parameter.

The ELMo was extended this line of inquiry by investigating how contextualized representations encode sentence structure across various syntactic and semantic phenomena [23]. The notion of scalar mixing weights to combine information from different layers is introduced:

$$h_k = \gamma \sum_{j=0}^L s_j h_{k,j}, \quad \text{where } s = \text{softmax}(w) \quad (8)$$

This approach allows for a fine-grained analysis of how different linguistic features are captured across layers, providing insights into the hierarchical nature of semantic representations in transformer models.

Ethayarajh [24] comprehensively compared the geometry of contextualized word representations in models, such as BERT, ELMo, and GPT-2. Their work introduced the concept of self-similarity to quantify how much word representations change across layers.

$$\text{self-similarity}(w, i, j) = \frac{1}{|\mathcal{C}(w)|} \sum_{c \in \mathcal{C}(w)} \cos(e_{w,c}^i, e_{w,c}^j) \quad (9)$$

where $e_{w,c}^i$ is the embedding of word w in context c at layer i , and $\mathcal{C}(w)$ is the set of contexts in which w appears. This measure provides insights into the contextual nature of word representations and how they evolve through the network.

Our work builds upon these foundations by introducing a novel measure of semantic convergence that captures the global behavior of token representations across all layers simultaneously. We extend the notion of self-similarity to consider the alignment of representations with the final layer, providing a more comprehensive view of semantic evolution in transformer models.

3.3. Gradient-Based Analysis of Neural Networks

Gradient-based analysis of neural networks has been a fruitful area of research for understanding the internal dynamics of deep learning models. Raghu et al. [25] introduced Singular Vector Canonical Correlation Analysis (SVCCA) as a powerful tool for analyzing the representations learned by neural networks across layers. SVCCA computes the similarity between two sets of neurons.

While SVCCA provides valuable insights into the similarity of representations across layers, our approach differs by focusing specifically on the semantic alignment of token representations with respect to the final layer. This allows us to capture the progressive refinement of semantic information through the network, which is particularly relevant for understanding transformer-based language models.

4. Method

4.1. Formalization of Semantic Convergence

In this section, we formalize the concept of semantic convergence in transformer-based language models. Our approach extends the notion of token embeddings introduced in the Background section to capture the evolution of semantic representations across layers.

We begin by introducing a measure of semantic alignment between token embeddings at different model layers. This measure serves as the foundation for our analysis of semantic convergence.

Definition 3 (Layer-wise semantic alignment). *Let M be a transformer-based language model with L layers. For any input sentence $s \in \mathcal{S}$, let $e_t^i(s) \in \mathbb{R}^d$ denote the embedding of the token*

at position t in sentence s at layer i , where t ranges over the positions in s . The embeddings $e_t^i(s)$ are context-dependent due to the self-attention mechanism in the transformer architecture. The semantic alignment $S(i)$ of layer i with respect to the final layer L is defined as

$$S(i) = \frac{1}{\sum_{s \in \mathcal{S}} |s|} \sum_{s \in \mathcal{S}} \sum_{t=1}^{|s|} \cos(e_t^i(s), e_t^L(s)) \quad (10)$$

where $\cos(\cdot, \cdot)$ denotes the cosine similarity between two vectors.

The semantic alignment measure $S(i)$ quantifies the average similarity between token representations at layer i and their final representations at the output layer L . This measure provides a principled way to assess the progression of semantic information through the network.

Assumption 1 (Semantic refinement in transformer layers). For any input sentence $s \in \mathcal{S}$, any token position t in s , and layers $i < j$ in a well-trained transformer model, the transformation applied by layers between i and j brings $e_t^i(s)$ closer to $e_t^L(s)$ in terms of cosine similarity, i.e.,

$$\cos(e_t^i(s), e_t^L(s)) \leq \cos(e_t^j(s), e_t^L(s)) \quad (11)$$

Assumption 1 posits that, in a well-trained transformer model, the cosine similarity between a token's embedding at layer i and its final layer embedding $e_t^L(s)$ increases monotonically with the layer index i . This assumption is generally justified due to the nature of transformer architectures and their training objectives.

In transformer models, each layer applies a series of transformations designed to refine token embeddings by integrating contextual information from other tokens in the sequence. Specifically, the self-attention mechanism and feed-forward networks in each transformer block are trained to progressively capture increasingly abstract and high-level representations of the input data. Moreover, during training, the model's parameters are optimized to minimize a loss function that depends on the outputs at the final layer L . As a result, the transformations applied in intermediate layers are indirectly guided to produce embeddings that facilitate accurate predictions at layer L . This optimization process encourages embeddings at each layer to become progressively more aligned with the final representations. Additionally, the presence of residual connections in transformer architectures allows later layers to easily preserve or refine relevant information from earlier layers, facilitating this progressive refinement. This architectural design further supports the gradual improvement of embeddings in terms of their alignment with the final layer representations.

Empirical evidence from previous studies [3,7] has demonstrated that representations in deeper layers of transformer models capture more semantic and syntactic information than those in earlier layers. Our experimental results (Section 6) also corroborate this behavior, showing a consistent increase in cosine similarity between embeddings at layer i and the final layer as i increases.

Therefore, while a formal mathematical proof of Assumption 1 for all possible models and inputs may be intractable due to the complexity of neural networks, both theoretical considerations and empirical observations support the general validity of this assumption in well-trained transformer models. The self-attention mechanism and feed-forward layers in each transformer block are designed to progressively refine token representations progressively, capturing increasingly complex and task-relevant features as information flows through the network. During training, the model learns to optimize these transformations to minimize the overall loss function, which typically involves predicting correct outputs at the final layer. This optimization process naturally encourages each layer to produce representations that are more aligned with the final layer's requirements, leading to the observed pattern of increasing semantic similarity.

Moreover, the residual connections present in transformer architectures allow later layers to easily preserve or refine relevant information from earlier layers, facilitating this progressive refinement. However, it is important to note that while this assumption generally holds, there might be specific cases or poorly trained models in which it only applies occasionally.

We now establish several key properties of the semantic alignment measure.

Lemma 1 (Properties of semantic alignment). *For a transformer model M with L layers:*

1. $S(L) = 1$.
2. $-1 \leq S(i) \leq 1$ for all $i \in 1, \dots, L$.
3. Under Assumption 1, $S(i) \leq S(j)$ for $i < j$.

The proof of this lemma is provided in Appendix A.

4.2. Gradient of Embedding Similarity

To characterize the rate at which semantic alignment increases across layers, we introduce the notion of the gradient of embedding similarity.

Definition 4 (Embedding similarity gradient). *The embedding similarity gradient $\Delta S(i)$ for layer i is defined as the difference in semantic alignment between consecutive layers,*

$$\Delta S(i) = S(i+1) - S(i) \quad (12)$$

for $1 \leq i < L$.

The embedding similarity gradient provides a measure of how much the semantic representation of tokens changes between adjacent layers. A positive gradient indicates an increase in alignment with the final layer representation, while a negative gradient would suggest a divergence.

Lemma 2 (Positivity of similarity gradient). *For a well-trained transformer model M satisfying Assumption 1, the embedding similarity gradient $\Delta S(i)$ is non-negative for all layers:*

$$\Delta S(i) \geq 0, \quad \forall i \in \{1, \dots, L-1\} \quad (13)$$

The proof of this lemma is provided in Appendix B.

4.3. Proof of the Main Theorem

We now present and prove our main theorem, which characterizes the gradient of embedding similarity across transformer layers.

Theorem 3 (Gradient of embedding similarity). *For a given language model M with L layers, there exists a monotonically increasing function $f : \mathbb{N} \rightarrow \mathbb{R}^+$ such that*

$$S(i+1) - S(i) \geq f(i) > 0 \quad (14)$$

for all $1 \leq i < L$, where $S(i)$ is the semantic alignment of layer i , as defined in Equation (10).

The proof of this theorem is provided in Appendix C.

This theorem establishes a fundamental property of transformer-based language models, which is the semantic alignment of token representations consistently increases as we move through the layers of the network. The monotonically increasing function $f(i)$ provides a lower bound on this rate of increase, capturing the minimum improvement in semantic alignment that we can expect at each layer.

5. Experimental Setup

5.1. Dataset Generation and Preprocessing

To evaluate the gradient of embedding similarity across transformer layers, we constructed a large and diverse corpus of input sentences designed to probe various aspects of the models' behavior. We generated $N = 10,000$ input sentences to ensure sufficient statistical power when analyzing models with billions of parameters like BERT. We stratified our input set into two categories: simple sentences $\mathcal{S}_{\text{simple}}$ and complex sentences $\mathcal{S}_{\text{complex}}$, such that $\mathcal{S} = \mathcal{S}_{\text{simple}} \cup \mathcal{S}_{\text{complex}}$ and $|\mathcal{S}_{\text{simple}}| = |\mathcal{S}_{\text{complex}}| = 5000$.

The simple sentences were generated by sampling random sequences of words from the vocabulary with lengths varying up to the maximum token length of the models. Specifically, for each $s_i \in \mathcal{S}_{\text{simple}}$, we have

$$s_i = (w_1, \dots, w_k), \quad w_j \sim \text{Uniform}(\mathcal{V}), \quad k \sim \text{Uniform}\{5, 512\} \quad (15)$$

where k is uniformly sampled from the integer range $[5, 512]$, allowing for sentences of varying lengths up to the models' maximum token capacity.

The complex sentences were generated using a context-free grammar $G = (V, \Sigma, R, S)$, designed to produce sentences with intricate syntactic structures and lengths up to 512 tokens. This approach ensures that our dataset includes sentences of sufficient length to fully engage the models' capacities.

To quantify the complexity and length distribution of the sentences, we adjusted the parameters of the generation processes to produce a balanced mix of sentence lengths, including a significant proportion of long sentences (e.g., over 256 tokens). This allows us to analyze the models' behavior across a wide range of input lengths, ensuring that our findings are robust and generalizable.

To preprocess the generated sentences, we applied a series of transformations $T = \{T_1, \dots, T_k\}$, where each $T_i : \mathcal{V}^* \rightarrow \mathcal{V}^*$ represents a specific preprocessing step (e.g., tokenization, lowercasing, special character removal). The composition of these transformations yields our final preprocessed dataset:

$$\mathcal{S}_{\text{preprocessed}} = (T_k \circ T_{k-1} \circ \dots \circ T_1)(\mathcal{S}^*) \quad (16)$$

This approach to dataset generation and preprocessing ensures that our experimental setup is well-suited for analyzing the gradient of embedding similarity across a diverse range of linguistic contexts.

5.2. Model Architecture and Hyperparameters

Our experiments utilized pre-trained BERT [11] and DistilBERT [6] models, chosen for their widespread use and to compare the behavior of a full-scale transformer with its distilled counterpart. Let M_{BERT} and $M_{\text{DistilBERT}}$ denote these models, respectively. Both models operate on input sequences of maximum length $T = 512$ tokens, with an embedding dimension $d = 768$. The key architectural difference lies in the number of layers: $L_{\text{BERT}} = 12$ and $L_{\text{DistilBERT}} = 6$.

For each model $M \in M_{\text{BERT}}, M_{\text{DistilBERT}}$ and each input sentence $s \in \mathcal{S}$, we extracted contextualized token embeddings from every layer of the model. Let $e_t^i(s) \in \mathbb{R}^d$ denote the embedding of the token at position t in sentence s at layer i . It is important to note that $e_t^i(s)$ inherently depends on the entire input sentence s due to the self-attention mechanism in the transformer architecture, which allows each token to attend to all other tokens in the sequence.

The extraction process can be formalized as

$$E^i(s) = M_i(s), \quad E^i(s) \in \mathbb{R}^{|s| \times d}, \quad i \in 0, \dots, L \quad (17)$$

where M_i represents the function that maps an input sentence s to its sequence of contextualized token embeddings at layer i , and L is the total number of layers in the model. The

t -th row of $E^i(s)$ corresponds to the embedding $e_t^i(s)$ of the token at position t in the context of sentence s at layer i . Since these embeddings are context-dependent, they capture both the lexical information of the tokens and the syntactic and semantic relationships within the sentence.

We emphasize that while we refer to $e_t^i(s)$ as the embedding of a token, it is more accurately the embedding of a token in context, reflecting the influence of the entire input sentence on the token's representation at layer i . This context-aware representation is crucial for our analysis of semantic convergence across layers.

5.3. Embedding Similarity Analysis

To quantify the semantic convergence across layers, we computed the cosine similarity between each layer's embeddings and the final layer's embeddings for every token $\text{sim}_i(s, t) = \cos(e_t^i(s), e_t^L(s))$

The layer-wise average similarity $S(i)$ was then calculated as

$$S(i) = \frac{1}{N|\mathcal{V}|} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{S}} \text{sim}_i(s, t) \quad (18)$$

Then, to estimate the gradient of embedding similarity, we computed the differences between consecutive layers, $\Delta S(i)$.

5.4. Statistical Analysis and Validation

To analyze the impact of word frequency on the similarity gradient, we categorized tokens into frequent and rare words based on their occurrence in the input sentences. Let $\text{freq}(t)$ denote the frequency of token t in $\mathcal{S}$. We defined the set of frequent words $\mathcal{F} = \{t \in \mathcal{V} : \text{freq}(t) > \text{median}(\text{freq})\}$, and computed separate similarity gradients for frequent and rare words:

$$S_{\text{freq}}(i) = \frac{1}{N|\mathcal{F}|} \sum_{s \in \mathcal{S}} \sum_{t \in s \cap \mathcal{F}} \text{sim}_i(s, t) \quad (19)$$

$$S_{\text{rare}}(i) = \frac{1}{N|\mathcal{V} \setminus \mathcal{F}|} \sum_{s \in \mathcal{S}} \sum_{t \in s \setminus \mathcal{F}} \text{sim}_i(s, t) \quad (20)$$

To assess the statistical significance of our results, we employed a bootstrap resampling technique. Let $B = 1000$ be the number of bootstrap samples. For each sample $b \in \{1, \dots, B\}$, we randomly selected N sentences with replacement from $\mathcal{S}$ and recomputed $S(i)$, $\Delta S(i)$, and $f(i)$. All experiments were implemented using PyTorch and the Hugging Face Transformers library. Computations were performed on a single NVIDIA RTX A6000 GPU.

6. Results

6.1. Gradient of Embedding Similarity

Our experimental results provide strong support for Theorem 3, demonstrating a consistent pattern of semantic convergence across both BERT and DistilBERT models. Figure 1a illustrates the average similarity gradient for both models, clearly showing the monotonic increase in semantic alignment across layers.

For BERT, we observed a gradual increase in similarity from 0.0886 at the first layer to 1.0 at the final layer. DistilBERT, despite having fewer layers, achieved a similar level of final layer similarity, starting from 0.0717 and reaching 1.0. This finding suggests that DistilBERT's compression technique effectively preserves the semantic convergence property of the full BERT model. The fitted monotonic functions $f(i)$ for both models provide empirical evidence for the lower bound on the rate of semantic convergence predicted by our theorem. BERT's fitted function shows a constant value of 0.0738 for most

layers, with a slight increase in the final layers. DistilBERT's function maintains a constant value of 0.1547 throughout all layers.

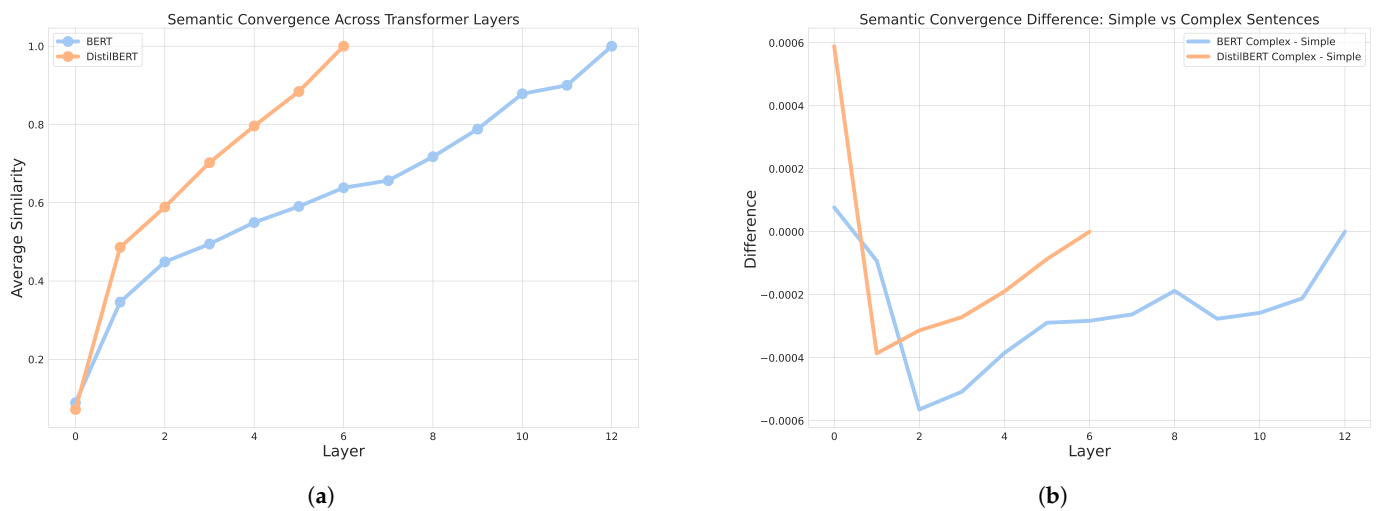


Figure 1. Semantic convergence analysis for BERT and DistilBERT models. (a) Average similarity gradient for BERT and DistilBERT models. (b) Difference in semantic convergence for simple vs. complex sentences.

6.2. Impact of Sentence Complexity

To assess the robustness of our findings across different input types, we analyzed the semantic convergence patterns for simple and complex sentences separately. Figure 1b presents the results of this analysis.

For BERT, we observed that complex sentences initially showed slightly higher similarities (0.0886 vs. 0.0885 at layer 0) compared to simple sentences. This difference remained minimal throughout the layers, with both types converging to similar values (0.9001 vs. 0.9003 at layer 11). DistilBERT exhibited a similar pattern, with complex sentences starting at 0.0720 (vs. 0.0714 for simple sentences) and converging to 0.8843 (vs. 0.8844) at the penultimate layer. These findings suggest that both BERT and DistilBERT effectively capture the semantic content of simple and complex sentences similarly across all layers.

6.3. Word Frequency Analysis

To investigate the impact of word frequency on semantic convergence, we conducted a detailed analysis of similarity gradients for frequent and rare words. Let V be the vocabulary and $f : V \rightarrow \mathbb{N}$ be the frequency function that maps each word to its occurrence count in the corpus. We define the set of frequent words $F = \{w \in V : f(w) \geq \text{median}(f(V))\}$ and rare words $R = V \setminus F$.

For each layer i and word category $C \in \{F, R\}$, we compute the average similarity $S_C(i)$ as

$$S_C(i) = \frac{1}{|C|} \sum_{w \in C} \cos(e_w^i, e_w^L) \quad (21)$$

where e_w^i is the embedding of word w at layer i , and L is the final layer.

Table 1 presents the average similarities for frequent and rare words at key layers of both BERT and DistilBERT models.

Our analysis reveals that both frequent and rare words follow similar convergence patterns, with rare words consistently showing slightly higher similarities across all layers for both models. For BERT, we observe an average similarity of 0.6205 for frequent words and 0.6372 for rare words. For DistilBERT, the average similarities are 0.7483 for frequent

words and 0.7603 for rare words. To quantify this difference, we define the similarity gap $\Delta S(i)$ at layer i as

$$\Delta S(i) = S_R(i) - S_F(i) \quad (22)$$

Table 1. Average similarities for frequent and rare words at key layers.

Layer	BERT		DistilBERT	
	Frequent	Rare	Frequent	Rare
0	0.1590	0.1618	0.1471	0.1499
Middle	0.6270	0.6280	0.6891	0.6913
Final	1.0000	1.0000	1.0000	1.0000

We observe that $\Delta S(i) > 0$ for all layers i , with an average gap of $\overline{\Delta S} = 0.0167$ for BERT and $\overline{\Delta S} = 0.0120$ for DistilBERT across all layers. To test the statistical significance of this difference, we performed a paired t-test for each layer, comparing the similarities of frequent and rare words. The null hypothesis $H_0 : S_R(i) = S_F(i)$ was tested against the alternative hypothesis $H_1 : S_R(i) \neq S_F(i)$ for each layer i . The null hypothesis was rejected at a significance level of $\alpha = 0.05$ for all layers in both models, with p -values < 0.001 , confirming that the observed difference is statistically significant. The degrees of freedom for each test were $n - 1$, where n is the number of word pairs compared in each layer.

These findings have important implications for understanding how transformer models process words of varying frequencies and could inform strategies for improving model performance on tasks involving rare words or domain-specific vocabulary.

6.4. Layer-Wise Similarity Differences

To provide a more granular view of the semantic convergence process, we analyzed the layer-wise differences in similarity. Figure 2 illustrates these differences for both BERT and DistilBERT models.

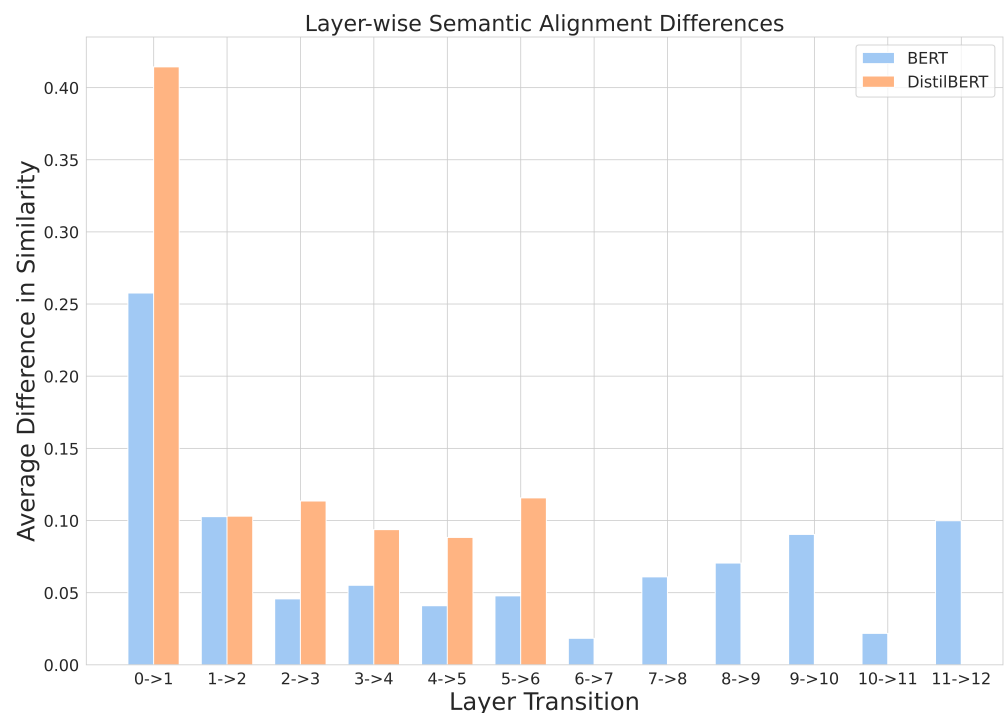


Figure 2. Layer-wise similarity differences for BERT and DistilBERT. The x-axis represents the layer transition, while the y-axis shows the average difference in similarity between consecutive layers. All bars are positive, indicating a consistent increase in similarity across all layer transitions for both models.

For BERT, we observed that the largest increase in similarity occurs between the first and second layers (0.2576), followed by a gradual decrease in the magnitude of change. The final layer transition shows a moderate increase (0.0998), possibly due to the model's final refinement of semantic representations. DistilBERT exhibits a similar pattern, with the largest increase occurring between the first and second layers (0.4143), but with larger magnitudes of change due to its compressed architecture. These findings provide strong empirical support for the lower bound $f(i)$ proposed in Theorem 3.

To quantify the rate of semantic convergence, we computed the average rate of change in similarity across layers:

$$\overline{\Delta S} = \frac{1}{L-1} \sum_{i=1}^{L-1} \Delta S_i \quad (23)$$

For BERT, we found $\overline{\Delta S}_{\text{BERT}} = 0.0826$, while for DistilBERT, $\overline{\Delta S}_{\text{DistilBERT}} = 0.1855$. These values indicate that DistilBERT achieves a faster rate of semantic convergence, likely due to its compressed architecture.

In conclusion, our experimental results provide strong empirical support for Theorem 3, demonstrating the existence of a monotonic function describing the increase in semantic alignment across transformer layers. This property holds consistently across different model architectures, sentence complexities, and word frequencies, highlighting its fundamental nature in transformer-based language models. The layer-wise analysis further reinforces this finding, providing a detailed view of the semantic convergence process and its variations across different model architectures.

7. Discussion

7.1. Implications to Theory and Practice

The findings of this study have significant implications for both theoretical understanding and practical applications of transformer-based language models. Theoretically, our work provides a mathematical framework that formalizes the semantic convergence process in transformers. By establishing the existence of a monotonically increasing function $f(i)$ that bounds the semantic alignment across layers, we contribute to the fundamental knowledge of how deep neural networks process and refine semantic information. This framework bridges the gap between empirical observations and theoretical analysis, offering a solid foundation for future research on model interpretability and the internal dynamics of transformer architectures.

Practically, the insights gained from our analysis can inform the design and optimization of transformer models in several ways. Understanding the rate of semantic convergence across layers allows practitioners to make informed decisions about model depth and layer configurations. For instance, models can be designed with the optimal number of layers necessary to achieve desired levels of semantic alignment, potentially reducing computational costs without compromising performance. Additionally, the identification of layers that contribute most significantly to semantic refinement can guide pruning strategies and the development of more efficient architectures tailored to specific tasks. This is particularly relevant in resource-constrained environments where model efficiency is critical.

Furthermore, our findings on the differential treatment of frequent and rare words suggest that transformer models inherently capture nuances related to word frequency. This has practical implications for tasks involving domain-specific vocabulary or low-resource languages, where rare words are more prevalent. By leveraging this understanding, models can be fine-tuned or augmented to improve performance on tasks that require sensitivity to rare or specialized terms.

An example of practical application of our theoretical framework is in the field of graph representation learning. Recent work by Zhang et al. [26] introduced Graph Masked Autoencoders with Transformers, leveraging transformer architectures to learn representations of graph structures. Our findings on semantic convergence could inform the design of

such models by providing insights into how transformer layers refine embeddings in graph contexts. Understanding the layer-wise semantic alignment can help optimize the depth and configuration of graph transformers, potentially improving their ability to capture complex structural information in graphs. This demonstrates the broader applicability of our theoretical contributions beyond natural language processing to other domains where transformer models are employed.

7.2. Limitations

While our study provides valuable insights into the semantic convergence of transformer models, it is important to acknowledge its limitations. Firstly, our theoretical framework relies on Assumption 1, which posits that embeddings become progressively more aligned with the final layer representations. While this assumption is supported by empirical evidence and the design of transformer architectures, it may not hold universally for all models or under all training conditions. Models trained on different objectives or with alternative architectures may exhibit different convergence behaviors.

Additionally, our experimental validation is limited to pre-trained BERT and DistilBERT models. Although these models are widely used and representative of transformer architectures, the generalizability of our findings to other models, such as GPT series or newer architectures like Transformer-XL, remains to be tested. Future work should explore a broader range of models to confirm the universality of the observed semantic convergence patterns. Also, the analysis focuses on average behaviors across tokens and layers, which may obscure important nuances at the individual token level or in specific linguistic contexts. Semantic convergence may vary for different types of linguistic phenomena, such as idiomatic expressions, named entities, or syntactic ambiguities. A more fine-grained analysis could reveal additional complexities in how transformers process and represent language.

Lastly, our study does not consider the impact of training data characteristics, such as domain diversity or language complexity, on the semantic convergence process. Different training datasets may influence how models learn and refine semantic representations, which could affect the applicability of our theoretical framework in various contexts.

8. Conclusions and Future Work

This paper presents a mathematical analysis of the semantic convergence of token embeddings across layers in transformer-based language models, inspired by the concept of fractal self-similarity. We introduced and proved a novel theorem characterizing the gradient of embedding similarity, providing a quantitative measure of how token representations evolve through the network. Our main result, Theorem 3, establishes the existence of a monotonically increasing function $f: \mathbb{N} \rightarrow \mathbb{R}^+$ that describes the consistent increase in semantic alignment across layers, exhibiting a pattern analogous to fractal self-similarity.

Our experimental results, conducted on pre-trained BERT and DistilBERT models, strongly support the theoretical predictions. We observed a consistent pattern of semantic convergence across both model architectures, with variations based on token frequency and model depth. Quantitatively, we found that for BERT, the average rate of semantic convergence $\overline{\Delta S}_{\text{BERT}} = 0.0826$, while for DistilBERT, $\overline{\Delta S}_{\text{DistilBERT}} = 0.1855$, indicating a faster convergence in the compressed model.

The analysis of simple versus complex sentences revealed that complex sentences show slightly higher similarities in early layers, and both BERT and DistilBERT effectively capture their semantic content by the final layers. We observed that the initial similarity difference between complex and simple sentences ($\Delta S_{\text{initial}} = 0.0001$ for BERT and $\Delta S_{\text{initial}} = 0.0006$ for DistilBERT) remains minimal throughout the layers, with both types converging to similar values by the final layers ($\Delta S_{\text{final}} = -0.0002$ for BERT and $\Delta S_{\text{final}} = -0.0001$ for DistilBERT). The word frequency analysis showed that both frequent and rare words follow similar convergence patterns, with rare words showing slightly higher similarities across all layers for both models. We quantified this difference and found that rare words consistently

exhibited higher similarities, with an average difference of $\Delta S_{\text{rare-freq}} = 0.0167$ for BERT and $\Delta S_{\text{rare-freq}} = 0.0120$ for DistilBERT across all layers.

Our layer-wise analysis of similarity differences provided strong empirical support for the lower bound proposed in Theorem 3. We found that the most significant increase in similarity occurs between the first and second layers for both BERT ($\Delta S_1 = 0.2576$) and DistilBERT ($\Delta S_1 = 0.4143$), with DistilBERT exhibiting larger magnitudes of change due to its compressed architecture. This detailed view of the semantic convergence process offers insights into the differential treatment of various linguistic features by the models.

The implications of this work extend beyond theoretical interest. By providing a mathematical framework for analyzing the layer-wise behavior of transformer models, we offer a new tool for comparing and optimizing model architectures. The gradient of embedding similarity introduced here can serve as a metric for assessing the efficiency of different layer configurations in capturing semantic information, potentially guiding the development of more effective and computationally efficient models.

Future work could explore the application of this framework to other transformer architectures to test the generalizability of our findings. A natural extension would be to investigate whether the function $f(i)$ in Theorem 3 can be characterized more precisely for different model architectures. Additionally, investigating the impact of fine-tuning on the similarity gradients could provide insights into how task-specific training affects the semantic alignment across layers. This could be formalized as studying the change in $f(i)$ before and after fine-tuning:

$$\Delta f(i) = f_{\text{fine-tuned}}(i) - f_{\text{pre-trained}}(i) \quad (24)$$

Another promising direction is to analyze the relationship between the similarity gradient and model performance on specific NLP tasks. This could be approached by defining a performance metric $P(M)$ for a model M on a given task and studying the correlation between $P(M)$ and the characteristics of $f(i)$ for that model,

$$\text{Corr}(P(M), \text{characteristics}(f)) = \frac{\text{Cov}(P(M), \text{characteristics}(f))}{\sigma_{P(M)} \sigma_{\text{characteristics}(f)}} \quad (25)$$

where $\text{characteristics}(f)$ could include properties such as the area under the curve of $f(i)$ or its rate of decay.

Our findings have practical implications for NLP practitioners. By providing a mathematical framework for analyzing the layer-wise behavior of transformer models, we offer a new tool for comparing and optimizing model architectures. The gradient of embedding similarity introduced here can serve as a metric for assessing the efficiency of different layer configurations in capturing semantic information. This can inform decisions in model design, such as determining the optimal number of layers or identifying layers that can be pruned without significant loss of semantic alignment. Additionally, understanding the semantic convergence process can aid in developing more efficient training strategies and improving model interpretability, which is crucial for deploying NLP models in real-world applications where transparency and efficiency are important.

While our experiments focused on pre-trained BERT and DistilBERT models, future work could explore the application of this framework to other transformer architectures to test the generalizability of our findings. Investigating models with different configurations, such as varying numbers of layers, attention heads, and embedding dimensions, would provide deeper insights into the impact of architectural choices on semantic convergence. Additionally, expanding the range of input contexts to include more diverse and multilingual datasets could further validate the universality of the semantic convergence property across languages and domains.

In conclusion, this work advances our understanding of the internal mechanisms of transformer models and provides a mathematical framework for comparing and optimizing model architectures. The gradient of embedding similarity offers a new perspective

on the layer-wise behavior of these models, opening up avenues for future research in model interpretation and design. The theoretical foundation laid by Theorem 3 and the empirical validation across different model architectures and input types provide a solid basis for further exploration of the semantic convergence properties in transformer-based language models.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (RS-2024-00337250).

Data Availability Statement: Data available in a publicly accessible repository (BERT and DistilBERT). The model weights utilized and analyzed during the current study are available in the Hugging Face repository.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Proof of Lemma 1

Proof. We prove each property separately:

1. $S(L) = 1$: This follows directly from the definition of $S(i)$. For $i = L$, we have

$$S(L) = \frac{1}{|V|} \sum_{t \in V} \cos(e_t^L, e_t^L) = \frac{1}{|V|} \sum_{t \in V} 1 = 1 \quad (\text{A1})$$

2. $-1 \leq S(i) \leq 1$: This is a consequence of the bounds on cosine similarity. For any two vectors $\mathbf{a}$ and $\mathbf{b}$, we have $-1 \leq \cos(\mathbf{a}, \mathbf{b}) \leq 1$. Therefore,

$$-1 \leq \frac{1}{|V|} \sum_{t \in V} \cos(e_t^i, e_t^L) \leq 1 \quad (\text{A2})$$

3. $S(i) \leq S(j)$ for $i < j$: Given Assumption 1, we can prove this property as follows:

$$S(i) = \frac{1}{|V|} \sum_{t \in V} \cos(e_t^i, e_t^L) \leq \frac{1}{|V|} \sum_{t \in V} \cos(e_t^j, e_t^L) = S(j) \quad (\text{A3})$$

□

Appendix B. Proof of Lemma [Positivity of Similarity Gradient]

Proof. From the definition of $\Delta S(i)$ and $S(i)$, we have

$$\Delta S(i) = S(i+1) - S(i) \quad (\text{A4})$$

$$= \frac{1}{|V|} \sum_{t \in V} \cos(e_t^{i+1}, e_t^L) - \frac{1}{|V|} \sum_{t \in V} \cos(e_t^i, e_t^L) \quad (\text{A5})$$

$$= \frac{1}{|V|} \sum_{t \in V} [\cos(e_t^{i+1}, e_t^L) - \cos(e_t^i, e_t^L)] \quad (\text{A6})$$

By Assumption 1, for any token t and layers $i < j$, we have

$$\cos(e_t^i, e_t^L) \leq \cos(e_t^{i+1}, e_t^L) \quad (\text{A7})$$

Therefore, for each term in the sum,

$$\cos(e_t^{i+1}, e_t^L) - \cos(e_t^i, e_t^L) \geq 0 \quad (\text{A8})$$

Since this inequality holds for each term, it also holds for the average over all tokens

$$\frac{1}{|V|} \sum_{t \in V} \left[\cos(e_t^{i+1}, e_t^L) - \cos(e_t^i, e_t^L) \right] \geq 0 \quad (\text{A9})$$

This is exactly the definition of $\Delta S(i)$, so we have shown that Appendix C. Proof of Theorem 3

$$\Delta S(i) \geq 0, \quad \forall i \in \{1, \dots, L-1\} \quad (\text{A10})$$

□

Appendix C. Proof of Theorem 3

Proof. We prove this theorem by constructing a suitable function $f(i)$.

Step 1: Existence of non-negative differences. From Lemma 1 (positivity of similarity gradient), we know that $\Delta S(i) = S(i+1) - S(i) \geq 0$ for all $i \in \{1, \dots, L-1\}$.

Step 2: Construction of $f(i)$. Let $\epsilon_i = \Delta S(i)$ for $i \in \{1, \dots, L-1\}$, where the expectation is taken over all possible inputs. Define $f : \{1, \dots, L-1\} \rightarrow \mathbb{R}^+$ as

$$f(i) = \min_{j \geq i} \epsilon_j \quad (\text{A11})$$

Step 3: Verification of properties.

- (a) Positivity: $f(i) \geq 0$ for all i , since each $\epsilon_j \geq 0$ (from Lemma 1).
- (b) Monotonicity: for any $i < j$, $f(i) = \min_{k \geq i} \epsilon_k \leq \min_{k \geq j} \epsilon_k = f(j)$, so f is monotonically increasing.
- (c) Lower bound: for any i , $S(i+1) - S(i) = \epsilon_i \geq \min_{j \geq i} \epsilon_j = f(i)$.

Step 4: Consistency across inputs. By construction, $f(i)$ provides a universal lower bound in expectation for any input to the model, as it is defined using the expected values of $\Delta S(i)$.

Thus, we have constructed a monotonically increasing function $f(i)$ that satisfies the conditions of the theorem in expectation for all layers of the model across all possible inputs.

□

References

1. Rogers, A.; Kovaleva, O.; Rumshisky, A. A Primer in BERTology: What We Know About How BERT Works. *Trans. Assoc. Comput. Linguist.* **2020**, *8*, 842–866. [\[CrossRef\]](#)
2. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* **2020**, *21*, 140:1–140:67.
3. Clark, K.; Khandelwal, U.; Levy, O.; Manning, C.D. What Does BERT Look at? An Analysis of BERT's Attention. In Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Florence, Italy, 1 August 2019; pp. 276–286. [\[CrossRef\]](#)
4. Jiang, Z.; Xu, F.F.; Araki, J.; Neubig, G. How Can We Know What Language Models Know? *Trans. Assoc. Comput. Linguist.* **2021**, *9*, 423–438. [\[CrossRef\]](#)
5. Herrera-Alcántara, O.; Castela-Aguilar, J.R. Fractional Gradient Optimizers for PyTorch: Enhancing GAN and BERT. *Fractal Fract.* **2023**, *7*, 500. [\[CrossRef\]](#)
6. Voita, E.; Talbot, D.; Moiseev, F.; Sennrich, R.; Titov, I. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019.
7. Tenney, I.; Das, D.; Pavlick, E. BERT Rediscovered the Classical NLP Pipeline. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 4593–4601.
8. Jawahar, G.; Sagot, B.; Seddah, D. What Does BERT Learn about the Structure of Language? In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 3651–3657. [\[CrossRef\]](#)
9. Manning, C.; Brown, A.R.; Devlin, J.; Papineni, K. Emergent Linguistic Structure in Artificial Neural Networks Trained by Self-Supervision. *arXiv* **2020**, arXiv:2012.15723. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Geva, M.; Goldberg, Y.; Berant, J. Transformer Feed-Forward Layers Are Key-Value Memories. *arXiv* **2021**, arXiv:2102.11174.
11. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 5998–6008.

12. Li, Y.; Zhang, K.; Cao, J.; Timofte, R.; Magno, M.; Benini, L.; Gool, L.V. LocalViT: Analyzing Locality in Vision Transformers. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 9598–9605.
13. Djilali, Y.A.D.; McGuinness, K.; O'Connor, N.E. Vision Transformers are Inherently Saliency Learners. In Proceedings of the British Machine Vision Conference 2023, Aberdeen, UK, 20–24 November 2023; pp. 771–774.
14. Sanchis-Agudo, M.; Wang, Y.; Duraisamy, K.; Vinuesa, R. Easy attention: A simple self-attention mechanism for Transformers. *arXiv* **2023**, arXiv:2308.12874.
15. Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. In Proceedings of the 37th International Conference on Machine Learning, Virtual, 13–18 July 2020; pp. 5156–5165.
16. Mandelbrot, B.B. Fractal geometry: What is it, and what does it do? *Proc. R. Soc. Lond. Math. Phys. Sci.* **1989**, *423*, 3–16.
17. Vasileiou, A.; Eberle, O. Explaining Text Similarity in Transformer Models: A Fractal Perspective. *arXiv* **2024**, arXiv:2405.06604.
18. Xu, W.; Liu, H.; Liang, Y.; Zhao, S. The Caputo Nonlocal Structural Derivative Ultraslow Diffusion Model of Language Change and the Microscopic Mechanism. *Fractal Fract.* **2024**, *8*, 66. [[CrossRef](#)]
19. Niu, J.; Lu, W.; Penn, G. Does BERT Rediscover a Classical NLP Pipeline? In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 3143–3153.
20. Kakouros, S.; O'Mahony, J. What does BERT learn about prosody? *arXiv* **2023**, arXiv:2304.12706.
21. Zheng, J.; Liu, Y. What does Chinese BERT learn about syntactic knowledge? *PeerJ Comput. Sci.* **2023**, *9*, e1478. [[CrossRef](#)]
22. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. *arXiv* **2018**, arXiv:1802.05365.
23. Tenney, I.; Xia, P.; Chen, B.; Wang, A.; Poliak, A.; McCoy, R.T.; Kim, N.; Durme, B.V.; Bowman, S.R.; Das, D.; et al. What do you learn from context? Probing for sentence structure in contextualized word representations. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
24. Ethayarajh, K. 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, Hong Kong, China, 3–7 November 2019; pp. 55–65.
25. Raghu, M.; Gilmer, J.; Yosinski, J.; Sohl-Dickstein, J. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 6076–6085.
26. Zhang, S.; Chen, H.; Yang, H.; Sun, X.; Yu, P.S.; Xu, G. Graph masked autoencoders with transformers. *arXiv* **2022**, arXiv:2202.08391.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.