SITH: Semantic Interpreter for Transformer Hierarchy

Anonymous ACL submission

Abstract

While Transformers and their derivatives have shown strong performance in various NLP tasks, understanding their internal mechanisms remains challenging. Mainstream interpretability research often focuses solely on numerical attributes, neglecting the complex semantic structure inherent in the model. We have developed the *SITH*(Semantic Interpreter for Transformer Hierarchy) framework to address this issue. We focus on creating universal text representation methods and uncovering the semantic principles of the Transformer's hierarchical structure. We use the convex hull method to represent sequence semantics in an n-dimensional Semantic Euclidean space and analyze semantic quality and quantity changes across the convex hull's three dimensions: point, line, and surface. Our analysis takes a dual perspective: a multi-layer cumulative perspective and an individual layer-to-layer shift perspective. When applied to machine translation, our results reveal potential semantic processes and emphasize the effectiveness of stacking and hierarchical differences. These insights are valuable for fine-tuning hyperparameters at the encoder and decoder layers.

1 Introduction

The Transformer architecture (Vaswani et al., 2017), acclaimed for its outstanding performance in many natural language processing tasks, is characterized by a modular encoder-decoder design. While this clever architecture of stacking encoder and decoder components improves the model's scalability, it poses a significant challenge in exploring model interpretability.

Traditionally, the attention mechanism in Transformer models has been considered intrinsic to their interpretability (Bibal et al., 2022). For instance, the integrated gradient-based self-attention attribution has illuminated the internal dynamics of Transformers (Hao et al., 2021), and attention-based visualization methods have clarified aspects of BERT's functioning (Clark et al., 2019). However, relying solely on attention mechanisms to explain the model is not enough (Jain and Wallace, 2019), which has drawn attention to other components of Transformer, such as the impact of the arrangement of feedforward layers on model performance (Press et al., 2020) and the importance of LayerNorm sublayers on model expression ability (Brody et al., 2023).

These current Transformer interpretation methods focus on the dissection of model numerical features and local components (e.g., attention weights) in the Transformer. While insightful, this quantitative perspective neglects the interpretable analysis of the model from a semantic perspective and a hierarchical stacking perspective.

Semantic Perspective: The Transformer attention weighting mechanism plays a crucial role. In addition to the intricate numerical features, the attention mechanism should also contain rich semantic information. Current research suggests that relying solely on attention weights for interpretation may overlook the subtle semantic changes presented by these models (Jain and Wallace, 2019). A more profound interpretation approach should delve into the semantic level of the models to reveal their cognitive processes and decision-making patterns from a semantic perspective.

Hierarchical stacking perspective: Focusing only on individual components is insufficient to elucidate the overall structural logic of the Transformer. Repeatedly stacking the model's uniquely modular components requires a macro-level interpretive perspective. This perspective is critical to deciphering the collective impact of the structure and understanding how the interactions of these stacked components shape the overall behavior of the model.

Addressing these gaps, our research pivots toward an enriched understanding of the Trans-

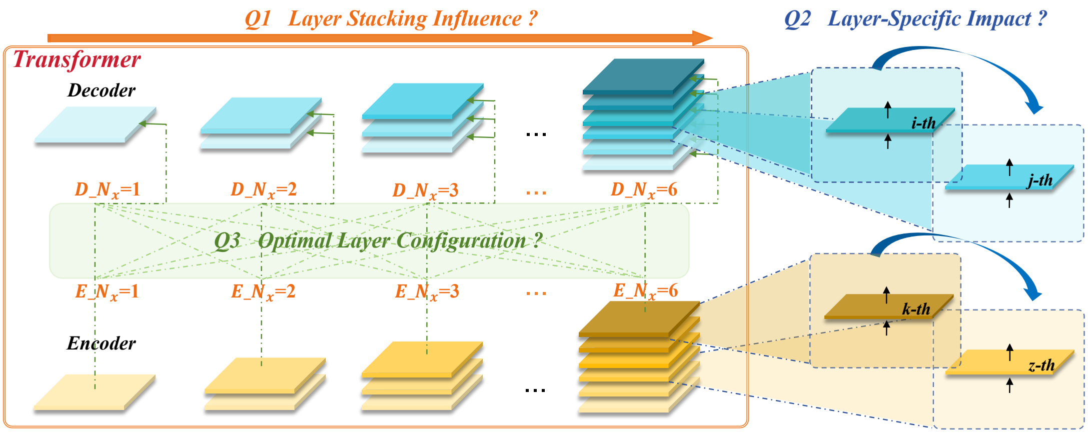

Figure 1: Three unresolved issues in the Transformer hierarchy

former's semantic complexity and architectural rationale, especially from a holistic perspective. This approach is pivotal in demystifying the strategic selection of layers in Transformer-based models, a process often guided more by intuition than systematic analysis. Our study is anchored around three critical inquiries, as depicted in Figure 1:

- Layer Stacking Influence: How does the Transformer's characteristic multi-layer stacking modulate the model's semantic processing and understanding?
- Layer-Specific Impact: What unique semantic contributions or alterations does each layer bring to the overall functioning of the Transformer model?
- Optimal Layer Configuration: What criteria or methodologies should be employed to determine the most effective number of layers for both the encoder and decoder components of the Transformer?

To tackle these pivotal questions, we introduce *SITH*(Semantic Interpreter for Transformer Hierarchy), a novel analytical framework that leverages the concept of ubiquitous text representation. *SITH* is specifically designed to unravel the semantic underpinnings of the Transformer's layered structure. By methodically extracting the model's output at each layer, we translate sequence semantics into an n-dimensional Semantic Euclidean space and then represent this data through a convex hull. This unique approach enables us to employ convex hull metrics to assess variations in the quality and quantity of semantics within the Transformer.

Our primary contributions through this work are threefold:

- Semantic Evaluation via Convex Hull Metrics: We have developed a novel method for assessing semantic quality and quantity, utilizing convex hull dimensions (points, lines, and surfaces) to analyze the semantic complexity inherent in Transformers.
- Dual-Perspective Hierarchical Analysis: Our approach introduces a two-pronged analysis of the Transformer's structure, encompassing both a multi-layer cumulative perspective and an individual layer-to-layer shift perspective, enabling a more comprehensive understanding of the model's semantic evolution.
- Insights into Encoding and Decoding Semantics: By exploring the nuances of semantic processes in encoding and decoding, our research demystifies the model's layering strategy, highlighting the effectiveness of its hierarchical structure and offering guidance for its optimization.

2 Related Work

The internal behavior of transformers is often considered a black box, which has sparked research on the interpretability of transformer models. Attention mechanism has always been an inherent way for Transformer interpretability. Clark et al. (2019) proposed attention-based visualization methods and detection classifiers to explain the behavior of models. Hao et al. (2021) introduced a heuristic algorithm to construct self-attention attribution trees and proposed an integrated gradient-based self-attention attribution method to explain the internal information interaction in Transformer. Tay et al. (2021) introduced a new model called SYNTHE-

SIZER, which can learn to synthesize self-attentive matrices to explain the importance and contributions of the dot-product self-attention mechanism to the performance of the Transformer model. The effect of multiple attention heads has also sparked discussions among researchers, Ma et al. (2021) exploring the relative importance of the number of attention heads in the model to help them achieve interpretability in cross-linguistic and multilingual tasks. In addition, some works have also extracted latent information from the hidden representations (Hewitt and Manning, 2019; Rosa and Mareek, 2019; Coenen et al., 2019) and attention weights (Mareček and Rosa, 2019) of the Transformer.

As many studies have shown that relying solely on attention to explain model predictions is not enough (Jain and Wallace, 2019), researchers have begun focusing on other local Transformer components. Domhan (2018) evaluated the importance of each component by retraining the model with other components removed. Wang and Tu (2020) conducted granularity analysis on the Transformer model components and studied each component's contribution to information flow and the critical phenomena of different components. In addition, the detailed study of encoder representations (Raganato and Tiedemann, 2018; Tang et al., 2019a,b,c), feed forward layers (Press et al., 2020), positional encoding (Chi et al., 2023), residual and normalization layers (Kobayashi et al., 2021; Brody et al., 2023) has also enhanced our understanding of Transformers.

3 Semantic Measurement Methods

Traditional word embedding techniques represent each word as a vector in an n-dimensional Euclidean space ($\mathbb{R}^n$), effectively capturing the meanings of words within predefined vocabulary lists. However, this approach often struggles to encapsulate implicit meanings and novel semantic combinations arising from word sequences. In contrast, Transformers, with their layered architecture, generate multiple hidden states that may not correspond directly to words in the existing vocabulary. Addressing this limitation, our study introduces the concept of an n-dimensional Semantic Euclidean space ($\mathbb{SR}^n$) as an extension of $\mathbb{R}^n$ to better represent sequence semantics (Zhang et al., 2020).

$$\mathbb{SR}^n = \{\forall x = (x_1, \dots, x_n) \in \mathbb{R}^n \mid x \rightarrow \textit{semantics}\} \tag{1}$$

The $\mathbb{SR}^n$ space encompasses the semantic correlations of all points in $\mathbb{R}^n$, offering a more nuanced representation of implicit semantic information. Each point in $\mathbb{SR}^n$ is an n-dimensional vector with semantic value. These semantic vectors are categorized into two types: 'abstract semantic points' and 'specific semantic points'. In the context of the Transformer model, words from the input and output sequences are represented as specific semantic points. Meanwhile, abstract semantic points refer to those elements that lack a direct vocabulary correspondence, typically aligning with the hidden states in intermediate layers of the Transformer. This representation enables a more comprehensive and dynamic understanding of the semantic content processed by Transformer models.

3.1 Convex Hull Representation of Semantics

Zhang et al. (2020) proposed representing the semantics of a text sequence as the convex hull in $\mathbb{SR}^n$. Given a sequence $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$, its meaning is defined as $\mathbf{ME}(\mathcal{X})$:

$$\mathrm{ME}(\mathcal{X}) = \mathrm{Conv}(\mathcal{X}) \tag{2}$$

Where $\mathrm{Conv}(\mathcal{X})$ is a set of convex combinations of all points in $\mathcal{X}$ (Faux and Pratt, 1979). Each point x_i in $\mathcal{X}$ is assigned a coefficient α_i, such that all these coefficients are non-negative, and their sum equals 1. The calculation is as follows:

$$\mathrm{Conv}(\mathcal{X}) = \left\{ \sum_{i=1}^{|\mathcal{X}|} \alpha_i x_i \mid \alpha_i \geq 0 \wedge \sum_{i=1}^{|\mathcal{X}|} \alpha_i = 1 \right\} \tag{3}$$

3.2 Evaluation Metrics for Semantics

We are mapping semantic relationships to convex hull relationships through the convex hull. We will use convex hull dimensions (points, lines, and surfaces) to evaluate and measure the semantic relationships between sequences before and after transformation.

Exploring the semantic 'quality' changes between sequences from the dimensions of 'points' and 'lines' in convex hulls:

Central Idea: Using convex hull centroids to represent the central idea of a sequence (Zhang et al., 2020). The formula is as follows:

$$\mathrm{CI}(\mathcal{X}) = \text{ Centroid } (\mathrm{ME}(\mathcal{X})) \tag{4}$$

Central Idea Offset: For two sequences $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ and $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$, where sequence $\mathcal{Y}$ is the semantic transformation of sequence $\mathcal{X}$. We model the distance between the

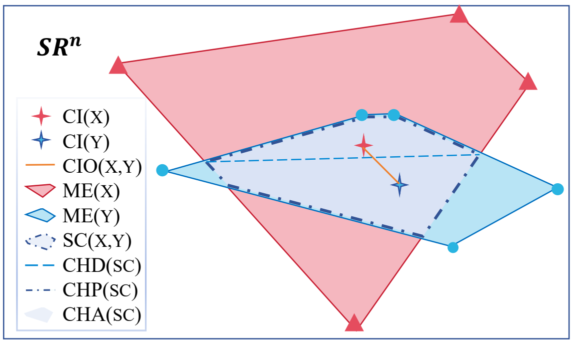

Figure 2: The sequence $\mathcal{X}$ is converted to $\mathcal{Y}$. During the conversion process, semantics' central idea (semantic quality) and coverage (semantic quantity) have changed, represented by solid orange lines and purple shadows.

central idea of two sequences as the Central Idea Offset(CIO). The formula is as follows:

$$\mathrm{CIO}(\mathcal{X}, \mathcal{Y}) = \| \, \mathrm{CI}(\mathcal{X}), \mathrm{CI}(\mathcal{Y}) \| \tag{5}$$

Exploring the semantic 'quantity' changes between sequences from the dimensions of 'lines' and 'surfaces' in convex hulls:

Semantic Coverage: Using semantic coverage (Zhang et al., 2021) to represent the overlap between two sequences, as shown in the purple shaded portion of Figure 2.

$$\mathrm{SC}(\mathcal{X}, \mathcal{Y}) = \mathrm{ME}(\mathcal{X}) \cap \mathrm{ME}(\mathcal{Y}) \tag{6}$$

Semantic Coverage Ratio: Semantic coverage, a common part between sequences before and after transformation, contains important semantic information, including shared semantics and symbiotic implicit semantics between sequences. We measure the proportion of the original semantics contained in the transformed sequence $\mathcal{Y}$ by calculating the ratio of the semantic coverage (SC) between sequences $\mathcal{X}$ and $\mathcal{Y}$ to the semantics quantity of the sequence $\mathcal{Y}$. The semantic quantity is represented by the different sizes and shapes of convex hulls, which are determined by their diameter, perimeter, and area. Therefore, we measure the proportion of original semantics in the transformed sequence from three aspects: the Semantic Coverage Diameter Ratio (SCDR), the Semantic Coverage Perimeter Ratio (SCPR), and the Semantic Coverage Area Ratio (SCAR). The formulas are as follows:

$$\mathrm{SCDR}(\mathcal{X}, \mathcal{Y}) = \frac{\mathrm{CHD}(\mathrm{SC}(\mathcal{X}, \mathcal{Y}))}{\mathrm{CHD}(\mathrm{ME}(\mathcal{Y}))} \tag{7}$$

$$\mathrm{SCPR}(\mathcal{X}, \mathcal{Y}) = \frac{\mathrm{CHP}(\mathrm{SC}(\mathcal{X}, \mathcal{Y}))}{\mathrm{CHP}(\mathrm{ME}(\mathcal{Y}))} \tag{8}$$

$$\mathrm{SCAR}(\mathcal{X}, \mathcal{Y}) = \frac{\mathrm{CHA}(\mathrm{SC}(\mathcal{X}, \mathcal{Y}))}{\mathrm{CHA}(\mathrm{ME}(\mathcal{Y}))} \tag{9}$$

We extract the semantic points represented by the vertices of the convex hull to form a sequence $\mathcal{V} = \{v_1, v_2, \ldots, v_m\}$ for simplifying calculations. As an example, for the convex hull constructed for SC, the methods for calculating the Convex Hull Diameter (CHD), Convex Hull Perimeter (CHP), and Convex Hull Area (CHA) are as follows:

$$\mathrm{CHD}(\mathcal{SC}) = \max_{v_i, v_j \in V} \|v_i - v_j\| \tag{10}$$

$$\mathrm{CHP}(\mathcal{SC}) = \sum_{i=1}^{m} ||v_i - v_{i+1}|| + ||v_m - v_1|| \tag{11}$$

$$\mathrm{CHA}(\mathcal{SC}) = \frac{1}{2} \left\| \sum_{i=1}^{m-1} (v_i - v_1) \times (v_{i+1} - v_1) \right\| \tag{12}$$

4 Semantic Interpreter for Transformer Hierarchy

This section will introduce an analysis framework called *SITH*(Semantic Interpreter for Transformer Hierarchy). Innovatively, we divide various sequences in Transformer into different dimensions and propose two analytical perspectives based on this. Each perspective is combined with the semantic evaluation metrics in Section 3.2 to form a comprehensive interpretable framework.

4.1 Sequence of Different Dimensions

The traditional Transformer architecture consists of a multi-layer stack of encoders and decoders. The input sequence is converted into various output sequences during the encoding and decoding process, including six encoder output sequences and six decoder output sequences. Previous studies have shown that each word in the sequence has its semantic meaning, and there are more abstract concepts at higher levels (Park et al., 2021). Therefore, we divide these sequences into dimensions, as shown in Figure 3.

The first encoder's input and the sixth decoder's output, the sequences closest to natural language, are grouped into the same dimension, defined as the '**language dimension**.' These two sequences are denoted as $\mathcal{NL}_{src}$ and $\mathcal{NL}_{tgt}$.

The output sequence of the sixth encoder undergoes the highest level of encoding and serves as the bridge for cross-lingual translation, containing the essential shared semantics between the source and target languages. This sequence is referred

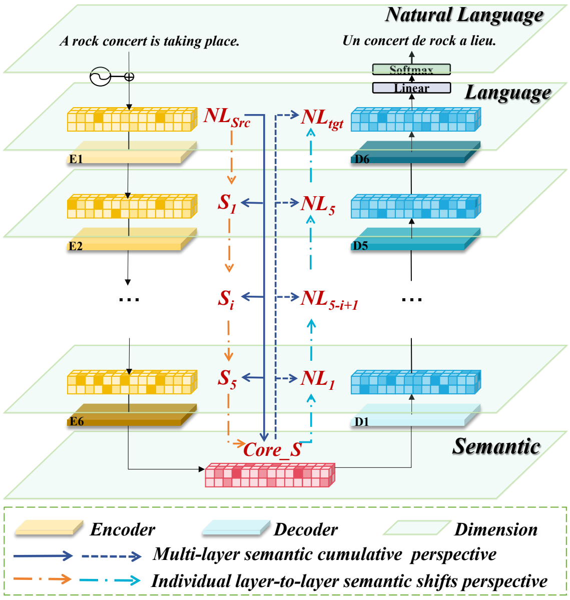

Figure 3: Two analytical perspectives of *SITH*(Semantic Interpreter for Transformer Hierarchy). The framework categorizes all sequences in the Transformer into different dimensions. The multi-layer cumulative perspective performs 'top-down' and 'bottom-up' semantic accumulation analysis on dimensions. In contrast, the independent layer-to-layer perspective analyzes the semantic relationships between sequences on adjacent dimensions.

to as *Core_S*, and its dimension is defined as the '**semantic dimension**'.

Sequences from intermediate encoders, where higher encoding corresponds to a closer proximity to the semantic dimension, are denoted as $\mathcal{S}_i$.

Sequences from intermediate decoders, where higher decoding corresponds to a closer proximity to the language dimension, are denoted as $\mathcal{NL}_j$. The relationship between them is j = 5 - i + 1.

4.2 Multi-layer Semantic Cumulative Perspective

The multi-layered semantic accumulation perspective aims to address the first issue raised in Section 1. The 'Layer Stacking Influence' was analyzed from two perspectives: semantic abstract accumulation and semantic concrete accumulation.

Semantic abstract accumulation perspective:

The semantic abstract accumulation perspective focuses on the transformation from the 'language dimension' to the 'semantic dimension' and aims to analyze how the stacking of encoders affects sequence semantics.

As the original sequence, we choose the encoder input $\mathcal{NL}_{src}$ in the language dimension. The encoder outputs $\mathcal{S}_i$ $(1{\le}i{\le}6, \mathcal{S}_6{=}Core\_S)$ in other dimensions as the transformation sequence. We evaluate the impact of i-layer encoders stacking by measuring the semantic relationship between $\mathcal{NL}_{src}$ and $\mathcal{S}_i$. Use SEI to represent semantic measurement methods, which involve different calculations in Section 3.2. An increase in i represents stacking, and $\mathcal{SAC\_T}$ indicates the change trend. The following formula reflects the analysis method of semantic abstract accumulation:

$$\mathcal{SAC\_T} = \Delta\{\mathrm{SEI}(\mathcal{NL}_{src}, \mathcal{S}_i)\}, \text{ for } 1 \le i \le 6 \tag{13}$$

Semantic concrete accumulation perspective:

The semantic concrete accumulation perspective focuses on the transformation from the 'semantic dimension' to the 'language dimension' and aims to analyze how the stacking of decoders affects sequence semantics.

We choose *Core_S* in the semantic dimension as the original sequence, and choose the decoder outputs $\mathcal{NL}_i$ $(1{\le}i{\le}6, \mathcal{NL}_6{=}\mathcal{NL}_{tgt})$ in the other dimensions as the transformation sequence. We evaluate the impact of i-layer decoders stacking on semantics by measuring the semantic relationship between *Core_S* and $\mathcal{NL}_i$. Using $\mathcal{SCC\_T}$ to represent the trend of semantic concrete accumulation, the formula is as follows:

$$\mathcal{SCC\_T} = \Delta\{\mathrm{SEI}(Core\_S, \mathcal{NL}_i)\}, \text{ for } 1 \le i \le 6 \tag{14}$$

Semantic measurement in the multi-layer accumulation perspective:

For the measurement of semantic 'quality,' the metric CIO is used. The semantic abstract accumulation perspective can be represented explicitly as $\mathcal{SAC\_T} = \Delta\{\mathrm{CIO}(\mathcal{NL}_{src}, \mathcal{S}_i)\}$, which reflects how the semantic center deviates from the language dimension during the transformation of a sequence from the 'language dimension' to the 'semantic dimension'. Similarly, in the semantic concrete accumulation perspective, $\mathcal{SCC\_T} = \Delta\{\mathrm{CIO}(Core\_S, \mathcal{NL}_i)\}$ can reflect how the semantic center deviates from the semantic dimension when a sequence evolves from the 'semantic dimension' to the 'language dimension.'

For the measurement of semantic 'quantity', indicators SCDR, SCPR, and SCAR are used to evaluate the proportion of the semantic quantity of the original sequence contained in the transformed sequence, where CHD, CHP, and CDA measure semantic quantity. Therefore, the semantic abstract cumulative perspective and the semantic concrete cumulative perspective can be expressed explicitly as the following formula, where

$$\text{SEI} \in \{\text{CHD}, \text{CHP}, \text{CHA}\}$$

$$\mathcal{SAC\_T} = \Delta\{\frac{\text{SEI}(\text{ME}(\mathcal{NL}_{src}) \cap \text{ME}(\mathcal{S}_i))}{\text{SEI}(\text{ME}(\mathcal{NL}_{src}))}\} \quad (15)$$

$$\mathcal{SCC\_T} = \Delta\{\frac{\text{SEI}(\text{ME}(Core\_\mathcal{S}) \cap \text{ME}(\mathcal{NL}_i))}{\text{SEI}(\text{ME}(Core\_\mathcal{S}))}\} \quad (16)$$

With the stacking of encoders, Equation 15 reflects the changes in semantic quantities containing language dimensions as the sequence approaches the semantic dimension.

With the stacking of decoders, Equation 16 reflects the changes in semantic quantities containing semantic dimensions as the sequence approaches the language dimension.

4.3 Individual Layer-to-layer Semantic Shifts Perspective

The independent layer-to-layer semantic shifts perspective aims to address the second issue raised in Section 1. The sequence is distributed in dimensions, and the changes in adjacent sizes are attributed to the role of the encoder or decoder between layers. From this perspective, the semantic relationship of sequences on adjacent dimensions is gradually calculated to evaluate the effectiveness of encoders and decoders at different levels.

For the encoding process, the transformation $\mathcal{S}_{i-1} \rightarrow \mathcal{S}_i$ $(1 \leq i \leq 6, \mathcal{S}_0 = \mathcal{NL}_{src}, \mathcal{S}_6 = Core\_\mathcal{S})$ is attributed to the effects of the i-th layer encoder. Similarly, for the decoding process, the transformation $\mathcal{NL}_{i-1} \rightarrow \mathcal{NL}_i$ $(1 \leq i \leq 6, \mathcal{NL}_0 = Core\_\mathcal{S}, \mathcal{NL}_6 = \mathcal{NL}_{tgt})$ is attributed to the effects of the i-th layer decoder. SEI is used to represent semantic evaluation metrics ($\text{SEI} \in \{\text{CIO}, \text{SCDR}, \text{SCPR}, \text{SCAR}\}$), and the effects of the i-th layer encoder and i-th layer decoder are denoted as Enc_i and Dec_i, respectively. Therefore, the effects of different layers under this perspective can be expressed as:

$$\text{Enc}_i = \text{SEI}(\mathcal{S}_{i-1}, \mathcal{S}_i), \text{Dec}_i = \text{SEI}(\mathcal{NL}_{i-1}, \mathcal{NL}_i) \quad (17)$$

5 Experiment

5.1 Experimental Setup

To ensure the simplicity of the analysis, we utilized a standard Transformer model as described in (Vaswani et al., 2017), with a layer size of 512, feedforward sub-layer of 2048, 8 attention heads, and a dropout rate of 0.1. The experiment focused on machine translation tasks using the multi30k dataset (Elliott et al., 2016), conducting interpretability analysis on four datasets, including the 2016_flickr and 2017_flickr test sets for English-German and English-French. Semantic analysis were integrated into the translation process, utilizing greedy decoding for text generation. For visualization, we employed t-SNE to reduce vector dimensions to two for convex hull calculations in this reduced space. Code will be available at https://anonymous.4open.science/r/SITH-39BE.

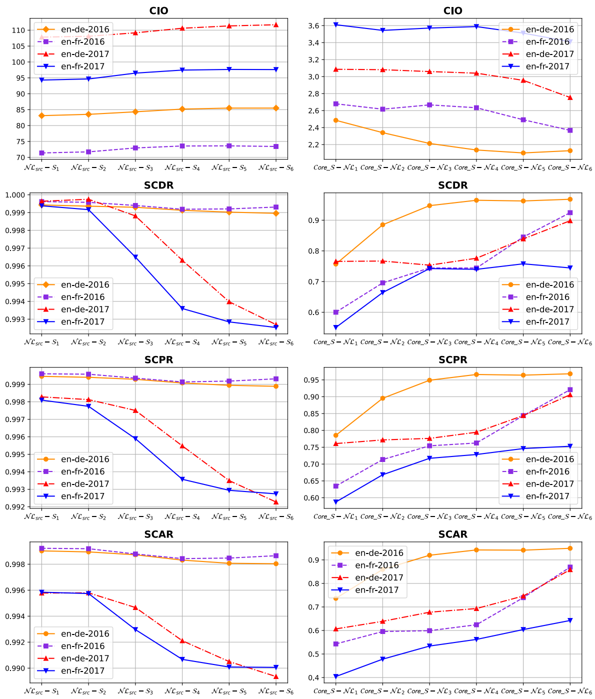

Figure 4: Results of multi-layer semantic cumulative perspective

5.2 Analysis from the Multi-layer Semantic Cumulative Perspective

The results are depicted in Figure 4. The left column represents the observation results of $\mathcal{SAC\_T}$, and the right column represents the observations of $\mathcal{SCC\_T}$. Based on this, we provide insights into the internal semantic transformation mechanism of the Transformer and demonstrate the effectiveness of stacking encoders and decoders:

- Encoder stacking results in an increasing deviation of the sequence's central idea from the source language, manifested as a highly abstract process of semantics.

- Decoder stacking results in a broader coverage of core semantics. The central idea aligns more closely with the core semantics, thereby improving the accuracy and semantic richness of the target language. This manifests as a process of semantic determination.

A opposite trend is shown in $\mathcal{SAC\_T}$ and $\mathcal{SCC\_T}$. In the perspective of semantic abstraction accumulation, there is a 'top-down' transformation of sequence dimensions, where the semantic deviation of each dimension from the initial language dimension increases ($\Delta\{\text{CIO}(\mathcal{NL}_{src}, \mathcal{S}_i)\}$shows an upward trend), and the semantic quantity containing the language dimension decreases (observed in the left column SCDR, SCPR, SCAR). On the other hand, in the perspective of semantic concretization accumulation, the sequence dimensions undergo a 'bottom-up' transformation. In the process of approaching the language dimension, the central idea of the sequence becomes increasingly aligned with the essential core semantics of the semantic dimension ($\Delta\{\text{CIO}(\mathcal{NL}_{src}, \mathcal{S}_i)\}$ shows a downward trend), and the quantity of semantic dimension contained in the sequence increases (observed in the right column SCDR, SCPR, SCAR).

Therefore, the process of encoder stacking is a semantic abstraction process. As the number of stacking layers increases, the rich semantic information is abstracted into higher-level representations, corresponding to a greater deviation from the semantic center of the language dimension. Usually, we consider the original sequence to be 'concrete'. Hence, each layer of the encoder aims to extract more advanced, universal, and concise semantic information, while ignoring certain specific and unnecessary details of the input sequence. Therefore, the semantic quantity of language dimensions gradually decreases during the superposition process, but becomes more general and abstract.

Decoder stacking is considered a process of semantic determination. *Core_ S*, as the input for each decoder layer, encapsulates the highly universal and advanced semantic representation of $\mathcal{NL}_{src}$, fundamentally reflecting the core semantics. The source language and target language share this core semantics. The decoder is responsible for generating the target language. In the process of layer-by-layer stacking, each decoder layer fine-tunes the sequence around *Core_ S*. On the one hand, the semantic center is more consistent with the core semantics, ensuring the accuracy of the meaning in the target language. On the other hand, the translation results are gradually refined and concretized, aiming to cover as much core semantic content as possible, ensuring that the generated text has expressive power. This gradually leads the target language towards a deterministic direction consistent with the core semantic expression.

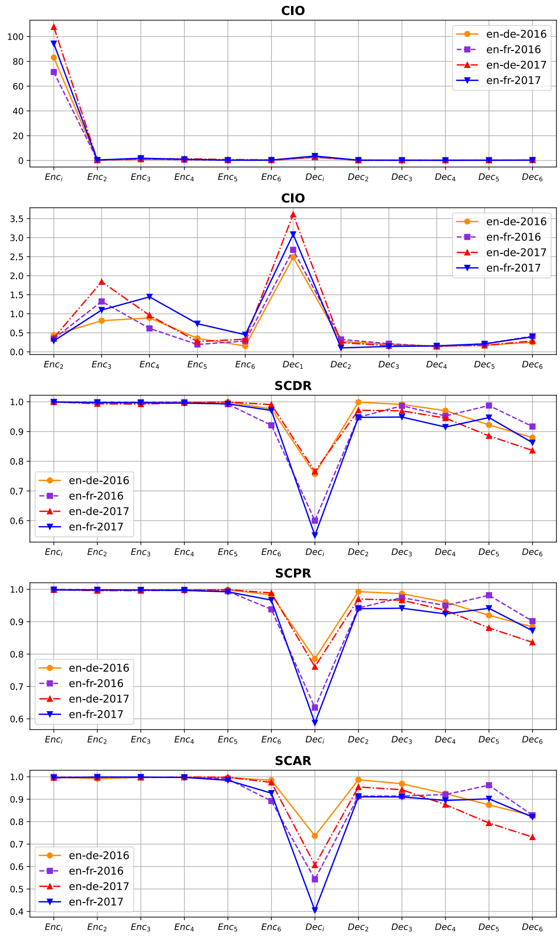

Figure 5: Results of individual layer-to-layer semantic shifts perspective

5.3 Analysis from the individual layer-to-layer semantic shifts perspective

Results of individual layer-to-layer semantic shifts perspective are shown in Figure 5 and Table 1. Calculate the effect of different layers by measuring the convex hull between adjacent dimensions. According to 5.2, the encoding process is a continuous abstraction of semantics. For semantic quality, a larger deviation indicates a higher level of abstraction. In terms of semantic quantity, the lower the degree of inclusion of the original sequence, the higher the level of abstraction. Therefore, larger CIOs and smaller SCDR, SCPR, and SCAR represent better hierarchical effects in the encoder section. On the other hand, the stacking of decoders leads to the continuous determination of semantics, which makes semantic expression more specific and reduces deviations from the semantic core. Therefore, smaller CIOs are preferred in the decoder section, while larger SCDR, SCPR, and SCAR values represent better layering effects. The

	en-de						en-fr					
	D_1	**D_2**	**D_3**	**D_4**	**D_5**	**D_6**	**D_1**	**D_2**	**D_3**	**D_4**	**D_5**	**D_6**
E_1	19.28	22.01	22.12	21.82	22.31	21.61	29.78	34.93	36.14	34.86	33.88	34.67
E_2	19.55	22.60	22.59	22.79	22.21	22.74	32.41	35.24	36.37	35.84	**36.62**	35.76
E_3	20.07	23.34	**23.57**	23.04	22.49	22.49	31.72	**37.34**	**37.70**	36.30	**37.07**	35.70
E_4	22.49	22.33	23.20	22.40	22.85	23.00	31.67	36.42	**38.32**	**37.89**	**37.27**	36.43
E_5	19.37	22.16	22.79	22.84	23.27	23.13	31.71	**36.96**	**37.63**	**37.48**	**36.96**	36.23
E_6	19.17	22.55	22.60	23.03	23.30	23.46	31.25	**36.86**	**38.05**	**38.07**	**36.81**	36.53

Table 1: Translation BLEU scores for Transformers of different sizes

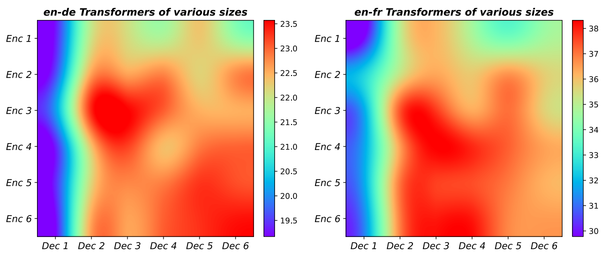

Figure 6: Translation BLEU scores for Transformers of different sizes

experimental results indicate that:

- The stacking effectiveness of Transformer is not a simple accumulation of equal effects, nor does it follow that higher layers are always more effective. Instead, it exhibits clear hierarchical differences.

- The hierarchical difference in the impact of encoders on semantic quality is significant, while the hierarchical difference in the impact of decoders on semantic quantity is significant.

Analyze the hierarchical differences of encoders: The second-row of Figure 5 has been added to highlight the differences. For semantic quality, the hierarchical effect first increases and then decreases, and the best performance occurs in Enc_3 or Enc_4. Higher layers cannot function more effectively. The high level of the encoder mainly affects the quantity of semantics.

As for the hierarchical differences in decoders, a turning point can be seen in Dec_1, this is attributed to the influence of cross-language and the introduction of Dec_2. It can be seen that Dec_2 has a significant positive impact on both semantic quality and quantity. Therefore, we believe that a layer of decoder is not enough. For semantic quality, the best performance occurs in Dec_3. For semantic quantities, the hierarchical effects of different language pairs vary. For English German, Dec_2 has the greatest impact, while for English French, Dec_3 and Dec_5 show the best performance.

To verify the correctness of the explanation for hierarchical differences mentioned above, we conducted 36 experiments on the English-German and English-French datasets, respectively, and obtained the BLEU scores of transformers of different sizes in machine translation tasks, as shown in Figure 6. Tacking the encoder when there is only one decoder layer cannot optimize performance, which is consistent with our previous analysis that more than one decoder layer is required. Both datasets exhibit similar characteristics at the best performance point, approximately at three layers. Using the original 6-layers model as the baseline, in the en-de, the 3-layers encoder and 2-layers decoder, as well as the 3-layers encoder and 3-layer decoder, performed similarly to the baseline. On the en-fr, the best performance occurred on the 4-layers encoder and 3-layers decoder, with 1.79 BLEU higher than the baseline. The performance of the 3-layers encoders and 3-layers decoders surpasses the baseline by 1.17 BLEU points, which aligns with our calculation above results for hierarchical effects.

6 Conclusions

In this work, we introduce *SITH*, a new framework designed to explore text representation in Transformer models. *SITH* delves into the semantic intricacies of Transformers' hierarchical structure, analyzing how layer stacking and different levels affect semantic transformation. It highlights the importance of the model's architecture in semantic processing and offers insights for optimizing hyperparameters in Transformer encoders and decoders, thus effectively linking theoretical concepts with practical applications.

Limitations

Due to limitations in computing power, this article only constructed convex hulls in a two-dimensional space and conducted semantic measurements. For simplicity in analysis, this article only verifies and analyzes the traditional structure of Transformers. In the future, we will conduct experiments on larger Transformer based model structures, while incorporating high-dimensional convex hull calculations as much as possible to solve the semantic problems in Transformers.

References

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3889–3900, Dublin, Ireland. Association for Computational Linguistics.

Shaked Brody, Uri Alon, and Eran Yahav. 2023. On the expressivity role of layernorm in transformers' attention.

Ta-Chung Chi, Ting-Han Fan, Li-Wei Chen, Alexander Rudnicky, and Peter Ramadge. 2023. Latent positional information is in the self-attention variance of transformer language models without positional embeddings. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1183–1193, Toronto, Canada. Association for Computational Linguistics.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. Visualizing and measuring the geometry of bert.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Desmond Elliott, Stella Frank, Khalil Sima'An, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions.

Faux and I. D. Pratt. 1979. Computational geometry for design and manufacture.

Yaru Hao, Li Dong, Furu Wei, and Ke Xu. 2021. Self-attention attribution: Interpreting information interactions inside transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12963–12971.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *North American Chapter of the Association for Computational Linguistics*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation.

Goro Kobayashi, Tatsuki Kuribayashi, Sho Yokoi, and Kentaro Inui. 2021. Incorporating residual and normalization layers into analysis of masked language models. *Association for Computational Linguistics*.

Weicheng Ma, Kai Zhang, Renze Lou, Lili Wang, and Soroush Vosoughi. 2021. Contributions of transformer attention heads in multi- and cross-lingual tasks. 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 1956–1966, Online. Association for Computational Linguistics.

David Mareček and Rudolf Rosa. 2019. From balustrades to pierre vinken: Looking for syntax in transformer self-attentions.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. *CoRR*, abs/2109.08359.

Ofir Press, Noah A. Smith, and Omer Levy. 2020. Improving transformer models by reordering their sublayers.

Alessandro Raganato and Jörg Tiedemann. 2018. An analysis of encoder representations in transformer-based machine translation. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: analyzing and interpreting neural networks for NLP*. The Association for Computational Linguistics.

Rudolf Rosa and David Mareek. 2019. Inducing syntactic trees from bert representations.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019a. Encoders help you disambiguate word senses in neural machine translation. *arXiv preprint arXiv:1908.11771*.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019b. Encoders help you disambiguate word senses in neural machine translation.

Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019c. Understanding neural machine translation by simplification: The case of encoder-free models. *arXiv preprint arXiv:1907.08158*.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. 2021. Synthesizer: Rethinking self-attention for transformer models. In *International conference on machine learning*, pages 10183–10192. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wenxuan Wang and Zhaopeng Tu. 2020. Rethinking the value of transformer components.

Chen Zhang, Qiuchi Li, Li Min Hua, and Dawei Song. 2021. How does attention affect the model? In *Findings*.

Cheng Zhang, Qiuchi Li, Li Hua, and Dawei Song. 2020. Assessing the memory ability of recurrent neural networks. *ArXiv*, abs/2002.07422.