Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors

Zeyu Yun[*2] **Yubei Chen**[*1,2] **Bruno A Olshausen**[2,4] **Yann LeCun**[1,3]

[1] Facebook AI Research

[2] Berkeley AI Research (BAIR), UC Berkeley

[3] New York University

[4] Redwood Center for Theoretical Neuroscience, UC Berkeley

Abstract

Transformer networks have revolutionized NLP representation learning since they were introduced. Though a great effort has been made to explain the representation in transformers, it is widely recognized that our understanding is not sufficient. One important reason is that there lack enough visualization tools for detailed analysis. In this paper, we propose to use dictionary learning to open up these 'black boxes' as linear superpositions of transformer factors. Through visualization, we demonstrate the hierarchical semantic structures captured by the transformer factors, e.g., word-level polysemy disambiguation, sentence-level pattern formation, and long-range dependency. While some of these patterns confirm the conventional prior linguistic knowledge, the rest are relatively unexpected, which may provide new insights. We hope this visualization tool can bring further knowledge and a better understanding of how transformer networks work. The code is available at https://github.com/zeyuyun1/TransformerVis.

1 Introduction

Though the transformer networks (Vaswani et al., 2017; Devlin et al., 2018) have achieved great success, our understanding of how they work is still fairly limited. This has triggered increasing efforts to visualize and analyze these "black boxes". Besides a direct visualization of the attention weights, most of the current efforts to interpret transformer models involve "probing tasks". They are achieved by attaching a light-weighted auxiliary classifier at the output of the target transformer layer. Then only the auxiliary classifier is trained for well-known NLP tasks like part-of-speech (POS) Tagging, Named-entity recognition (NER) Tagging, Syntactic Dependency, etc. Tenney et al. (2019) and Liu et al. (2019) show transformer models have excellent performance in those probing tasks. These results indicate that transformer models have learned the language representation related to the probing tasks. Though the probing tasks are great tools for interpreting language models, their limitation is explained in Rogers et al. (2020). We summarize the limitation into three major points:

- Most probing tasks, like POS and NER tagging, are too simple. A model that performs well in those probing tasks does not reflect the model's true capacity.
- Probing tasks can only verify whether a certain prior structure is learned in a language model. They can not reveal the structures beyond our prior knowledge.
- It's hard to locate where exactly the related linguistic representation is learned in the transformer.

Efforts are made to remove those limitations and make probing tasks more diverse. For instance, Hewitt and Manning (2019) proposes "structural probe", which is a much more intricate probing task. Jiang et al. (2020) proposes to generate specific probing tasks automatically. Non-probing methods are also explored to relieve the last two limitations. For example, Reif et al. (2019) visualizes embedding from BERT using UMAP and shows that the embeddings of the same word under different contexts are separated into different clusters. Ethayarajh (2019) analyzes the similarity between embeddings of the same word in different contexts. Both of these works show transformers provide a context-specific representation.

Faruqui et al. (2015); Arora et al. (2018); Zhang et al. (2019) demonstrate how to use dictionary learning to explain, improve, and visualize the uncontextualized word embedding representations. In

[*] equal contribution. Correspondence to: Zeyu Yun <chobitstian@berkeley.edu>, Yubei Chen <yubeic@{fb.com, berkeley.edu}>

Proceedings of Deep Learning Inside Out (DeeLIO):
The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 1–10
Online, June 10, 2021.

this work, we propose to use dictionary learning to alleviate the limitations of the other transformer interpretation techniques. Our results show that dictionary learning provides a powerful visualization tool, leading to some surprising new knowledge.

2 Method

Hypothesis: contextualized word embedding as a sparse linear superposition of transformer factors. It is shown that word embedding vectors can be factorized into a sparse linear combination of word factors (Arora et al., 2018; Zhang et al., 2019), which correspond to elementary semantic meanings. An example is:

$$\begin{aligned}\text{apple} =& 0.09\text{“dessert”} + 0.11\text{“organism”} + 0.16\\ &\text{“fruit”} + 0.22\text{“mobile\&IT”} + 0.42\text{“other”}.\end{aligned}$$

We view the latent representation of words in a transformer as contextualized word embedding. Similarly, we hypothesize that a contextualized word embedding vector can also be factorized as a sparse linear superposition of a set of elementary elements, which we call *transformer factors*. The exact definition will be presented later in this section.

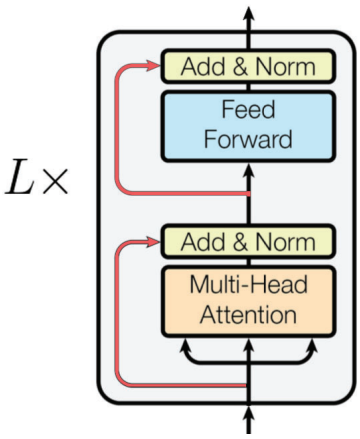


Figure 1: Building block (layer) of transformer

Due to the skip connections in each of the transformer blocks, we hypothesize that the representation in any layer would be a superposition of the hierarchical representations in all of the lower layers. As a result, the output of a particular transformer block would be the sum of all of the modifications along the way. Indeed, we verify this intuition with the experiments. Based on the above observation, we propose to learn a single dictionary for the contextualized word vectors from different layers' output.

To learn a dictionary of transformer factors with non-negative sparse coding.

Given a set of tokenized text sequences, we collect the contextualized embedding of every word using a transformer model. We define the set of all word embedding vectors from lth layer of transformer model as $X^{(l)}$. Furthermore, we collect the embeddings across all layers into a single set $X = X^{(1)} \cup X^{(2)} \cup \cdots \cup X^{(L)}$.

By our hypothesis, we assume each embedding vector $x \in X$ is a sparse linear superposition of *transformer factors*:

$$x = \Phi\alpha + \epsilon,\ s.t.\ \alpha \succeq 0, \tag{1}$$

where $\Phi \in \mathbb{R}^{d\times m}$ is a dictionary matrix with columns $\Phi_{:,c}$, $\boldsymbol{\alpha} \in \mathbb{R}^m$ is a sparse vector of coefficients to be inferred and $\boldsymbol{\epsilon}$ is a vector containing independent Gaussian noise samples, which are assumed to be small relative to $\boldsymbol{x}$. Typically $m > d$ so that the representation is *overcomplete*. This inverse problem can be efficiently solved by FISTA algorithm (Beck and Teboulle, 2009). The dictionary matrix Φ can be learned in an iterative fashion by using non-negative sparse coding, which we leave to the appendix section C. Each column $\Phi_{:,c}$ of Φ is a *transformer factor* and its corresponding sparse coefficient $\boldsymbol{\alpha}_c$ is its activation level.

Visualization by top activation and LIME interpretation. An important empirical method to visualize a feature in deep learning is to use the input samples, which trigger the top activation of the feature (Zeiler and Fergus, 2014). We adopt this convention. As a starting point, we try to visualize each of the dimensions of a particular layer, $X^{(l)}$. Unfortunately, the hidden dimensions of transformers are not semantically meaningful, which is similar to the uncontextualized word embeddings (Zhang et al., 2019).

Instead, we can try to visualize the transformer factors. For a transformer factor $\Phi_{:,c}$ and for a layer-l, we denote the 1000 contextualized word vectors with the largest sparse coefficients $\alpha_c^{(l)}$ as $X_c^{(l)} \subset X^{(l)}$, which correspond to 1000 different sequences. For example, Figure 3 shows the top 5 words that activated transformer factor-17 $\Phi_{:,17}$ at layer-0, layer-2, and layer-6 respectively. Since a contextualized word vector is generally affected by many tokens in the sequence, we can use LIME (Ribeiro et al., 2016) to assign a weight to each token in the sequence to identify their relative

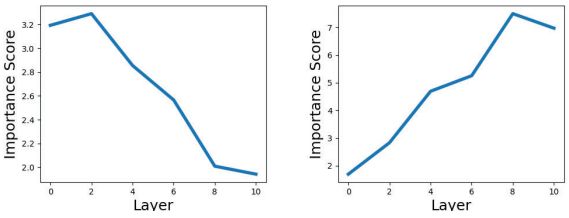
importance to α_c. The detailed method is left to Section 3.

To determine low-, mid-, and high-level transformer factors with importance score. As we build a single dictionary for all of the transformer layers, the semantic meaning of the transformer factors has different levels. While some of the factors appear in lower layers and continue to be used in the later stages, the rest of the factors may only be activated in the higher layers of the transformer network. A central question in representation learning is: "where does the network learn certain information?" To answer this question, we can compute an "importance score" for each transformer factor $\Phi_{:,c}$ at layer-l as $I_c^{(l)}$. $I_c^{(l)}$ is the average of the largest 1000 sparse coefficients $\alpha_c^{(l)}$'s, which correspond to $X_c^{(l)}$. We plot the importance scores for each transformer factor as a curve is shown in Figure 2. We then use these importance score (IS) curves to identify which layer a transformer factor emerges. Figure 2a shows an IS curve peak in the earlier layer. The corresponding transformer factor emerges in the earlier stage, which may capture lower-level semantic meanings. In contrast, Figure 2b shows a peak in the higher layers, which indicates the transformer factor emerges much later and may correspond to mid- or high-level semantic structures. More subtleties are involved when distinguishing between mid-level and high-level factors, which will be discussed later.

An important characteristic is that the IS curve for each transformer factor is relatively smooth. This indicates if a vital feature is learned in the beginning layers, it won't disappear in later stages. Instead, it will be carried all the way to the end with gradually decayed weight since many more features would join along the way. Similarly, abstract information learned in higher layers is slowly developed from the early layers. Figure 3 and 5 confirm this idea, which will be explained in the next section.

3 Experiments and Discoveries

We use a 12-layer pre-trained BERT model (Pre; Devlin et al., 2018) and freeze the weights. Since we learn a single dictionary of transformer factors for all of the layers in the transformer, we show that these transformer factors correspond to different levels of semantic or syntactic patterns. The patterns can be roughly divided into three categories:



(a) (b)

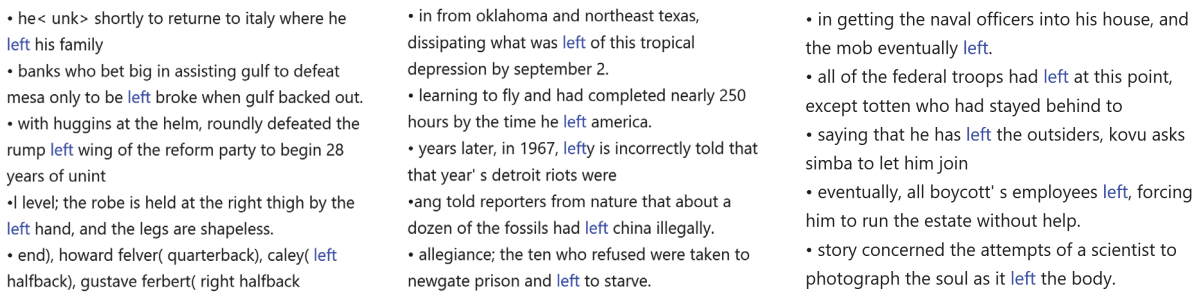
Figure 2: Importance score (IS) across all layers for two different transformer factors. (a) This figure shows a typical IS curve of a transformer factor corresponding to low-level information. (b) This figure shows a typical IS curve of a transformer factor corresponds to mid-level information.

word-level disambiguation, sentence-level pattern formation, and long-range dependency. In the following, we provide detailed visualization for each pattern category. Due to the space limit, only a small amount of the factors are demonstrated in the paper. To alleviate the "cherry-picking" bias, we also build a website for the interested readers to play with these results.

Low-level: word-level polysemy disambiguation. While the input embedding of a token contains polysemy, we find transformer factors with early IS curve peaks usually correspond to a specific word-level meaning. By visualizing the top activation sequences, we can see how word-level disambiguation is gradually developed in a transformer.

We show how the disambiguation effect develops progressively through each layer in Figure 3. In Figure 3, the top 5 activated words and their contexts for transformer factor $\Phi_{:,30}$ in different layers are listed. The top activated words in layer 0 contain the word "left" varying senses, which is being mostly disambiguated in layer 2 albeit not completely. In layer 4, the word "left" is fully disambiguated since the top-activated word contains only "left" with the word sense "leaving, exiting." We also show more examples of those types of transformer factors in Table 1: for each transformer factor, we list out the top 3 activated words and their contexts in layer 4. As shown in the table, nearly all top-activated words are disambiguated into a single sense.

Further, we can quantify the quality of the disambiguation ability of the transformer model. In the example above, since the top 1000 activated words



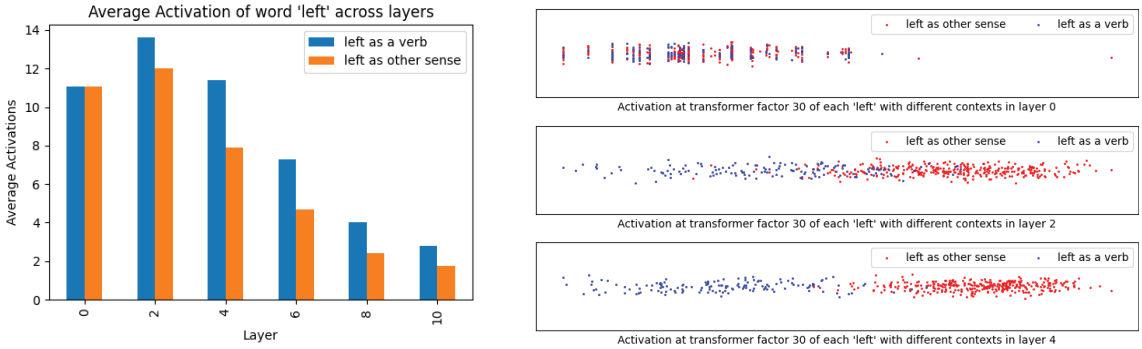
(a) layer 0 (b) layer 2 (c) layer 6

Figure 3: Visualization of a low-level transformer factor, $\Phi_{:,30}$ at different layers. (a), (b) and (c) are the top-activated words and contexts for $\Phi_{:,30}$ in layer-0, 2 and 4 respectively. We can see that at layer-0, this transformer factor corresponds to word vectors that encode the word "left" with different senses. In layer-2, a majority of the top activated words "left" correspond to a single sense, "leaving, exiting." In layer 4, all of the top-activated words "left" have corresponded to the same sense, "leaving, exiting." Due to space limitations, we invite the readers to use our website to see more of those disambiguation effects.

	Top 3 activated words and their contexts	Explanation
$\Phi_{:,2}$	• that snare shot sounded like somebody' d kicked open the door to your mind". • i became very frustrated with that and finally made up my mind to start getting back into things." • when evita asked for more time so she could make up her mind, the crowd demanded," ¡ ahora, evita,<	• Word "mind" • Noun • Definition: the element of a person that enables them to be aware of the world and their experiences.
$\Phi_{:,16}$	•nington joined the five members xero and the band was renamed to linkin park. • times about his feelings about gordon, and the price family even sat away from park' s supporters during the trial itself. • on 25 january 2010, the morning of park' s 66th birthday, he was found hanged and unconscious in his	• Word "park" • Noun • Definition: a common first and last name
$\Phi_{:,30}$	• saying that he has left the outsiders, kovu asks simba to let him join his pride • eventually, all boycott' s employees left, forcing him to run the estate without help. • the story concerned the attempts of a scientist to photograph the soul as it left the body.	• Word "left" • Verb • Definition: leaving, exiting
$\Phi_{:,33}$	• forced to visit the sarajevo television station at night and to film with as little light as possible to avoid the attention of snipers and bombers. • by the modest, cream@-@ colored attire in the airy, light@-@ filled clip. • the man asked her to help him carry the case to his car, a light@-@ brown volkswagen beetle.	• Word "light" • Noun • Definition: the natural agent that stimulates sight and makes things visible

Table 1: Several examples of low-level transformer factors. Their top-activated words in layer 4 are marked blue, and the corresponding contexts are shown as examples for each transformer factor. As shown in the table, nearly all of the top-activated words are disambiguated into a single sense. Please note the last example of $\Phi_{:,33}$ is a rare exception, the reader may check the appendix to see a more complete list. More examples, top-activated words and contexts are provided in Appendix.

and contexts are "left" with only the word sense "leave, exiting", we can assume "left" when used as a verb, triggers higher activation in $\Phi_{:,30}$ than "left" used as other sense of speech. We can verify this hypothesis using a human-annotated corpus: Brown corpus (Francis and Kucera, 1979). In this corpus, each word is annotated with its corresponding part-of-speech. We collect all the sentences contains the word "left" annotated as a verb in one set and sentences contains "left" annotated as other part-of-speech. As shown in Figure 4a, in layer 0, the average activation of $\Phi_{:,30}$ for the word "left" marked as a verb is no different from "left" as other senses. However, at layer 2, "left" marked as a verb triggers a higher activation of $\Phi_{:,30}$. In layer 4, this difference further increases, indicating disambiguation develops progressively across layers. In fact, we plot the activation of "left" marked as verb and the activation of other "left" in Figure 4b. In layer 4, they are nearly linearly separable by this



(a) (b)

Figure 4: (a) Average activation of $\Phi_{:,30}$ for word vector "left" across different layers. (b) Instead of averaging, we plot the activation of all "left" with different contexts in layer-0, 2, and 4. Random noise is added to the y-axis to prevent overplotting. The activation of $\Phi_{:,30}$ for two different word senses of "left" is blended together in layer-0. They disentangle to a great extent in layer-2 and nearly separable in layer-4 by this single dimension.

• wrote that" stankonia reeks of artful ambition rendered with impeccable skill"
• stated similar pros, describing the soundtrack as" suspenseful, dynamic and always adrenaline charged."
• with guest performances from vocalists frances maya and susan calloway, among others. the concert premiered several
•. it was an attitude of mind which tempered the sternness of his approach with an engaging humour and
•" all very effectively done, creepily atmospheric and splendidly gruesome"

(a) layer 4

• a film that' s stylish, breezily entertaining, and surprisingly sweet." on metacritic
• consensus reads" charming, audacious, and timely, wall@-@ e' s light
•.@ 1 out of 10, as being" beautiful, charismatic, engaging and one of the most
• that sinatra is" simply superb, comical, pitiful, childishly brave,
• everything madonna has been denounced for being — meticulous, calculated, domineering and artificial.

(b) layer 6

• new york times called it a" zany, lively, uninhibited, sexual odyssey
• she stated that the show was" breezy and entertaining and reasonably clever, at least when its sherlock
• of five stars, called it" a dark and delicious delight[and] a must@-@
• episodes, saying," it ' s smart, entertaining, and has moments so shocking that you '
•" full of exhilarating, ecstatic, thrilling, fun and sometimes downright silly songs"

(c) layer 8

Figure 5: Visualization of a mid-level transformer factor. (a), (b), (c) are the top 5 activated words and contexts for this transformer factor in layer-4, 6, and 8 respectively. Again, the position of the word vector is marked blue. Please notice that sometimes only a part of a word is marked blue. This is due to that BERT uses word-piece tokenizer instead of whole word tokenizer. This transformer factor corresponds to the pattern of "consecutive adjective". As shown in the figure, this feature starts to develop at layer-4 and fully develops at layer-8.

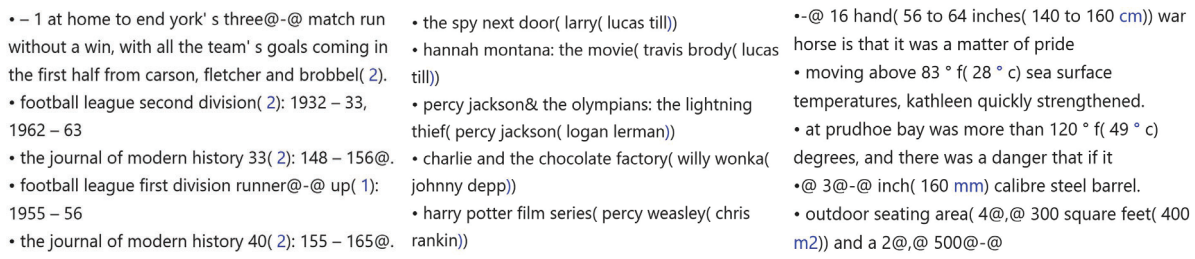
	Precision (%)	Recall (%)	F1 score (%)
Average perceptron POS tagger	92.7	95.5	94.1
Finetuned BERT base model for POS task	97.5	95.2	96.3
Logistic regression classifier with activation of $\Phi_{:,30}$ at layer 4	97.2	**95.8**	**96.5**

Table 2: Evaluation of binary POS tagging task: predict whether or not "left" in a given context is a verb.

single feature. Since each word "left" corresponds to an activation value, we can perform a logistic regression classification to differentiate those two types of "left". From the result shown in Figure 4a, it is pretty fascinating to see that the disambiguation ability of just $\Phi_{:,30}$ is better than the other two classifiers trained with supervised data. This result confirms that disambiguation is indeed done in the early part of pre-trained transformer model and we are able to detect it via dictionary learning.

Mid level: sentence-level pattern formation. We find most of the transformer factors, with an IS curve peak after layer 6, capture mid-level or high-level semantic meanings. In particular, the mid-level ones correspond to semantic patterns like phrases and sentences pattern.

We first show two detailed examples of mid-level transformer factors. Figure 5 shows a transformer factor that detects the pattern of consecutive usage of adjectives. This pattern starts to emerge at layer 4, develops at layer 6, and becomes quite reliable at layer 8. Figure 6 shows a transformer factor, which corresponds to a pretty unexpected pattern: "unit exchange", e.g., 56 inches (140 cm). Although this exact pattern only starts to appear at layer 8, the sub-structures that make this pattern, e.g., parenthesis and numbers, appear to trigger this factor in layers 4 and 6. Thus this transformer factor is also



(a) layer 4 (b) layer 6 (c) layer 8

Figure 6: Another example of a mid-level transformer factor visualized at layer-4, 6, and 8. The pattern that corresponds to this transformer factor is "unit exchange". Such a pattern is somewhat unexpected based on linguistic prior knowledge.

	2 example words and their contexts with high activation	Patterns	L4 (%)	L6 (%)	L8 (%)	L10 (%)
$\Phi_{:,13}$	• the steel pipeline was about 20 ° f(- 7 ° c) degrees. • hand(56 to 64 inches(140 to 160 cm)) war horse is that it was a	Unit exchange with parentheses	0	0	64.5	95.5
$\Phi_{:,42}$	• he died at the hospice of lancaster county from heart • holly' s drummer carl bunch suffered frostbite to his toes(while aboard the ailments on 23 june 2007.	Something unfortunate happened	94.0	100	100	100
$\Phi_{:,50}$	• hurricane pack 1 was a revamped version of story mode; • in 1998, the categories were retitled best short form music video, and best	Doing something again, or making something new again	74.5	100	100	100
$\Phi_{:,86}$	• he finished the 2005 – 06 season with 21 appearances and seven goals. • of an offensive game, finishing off the 2001 – 02 season with 58 points in the 47 games	Consecutive years, used in foodball season naming	0	100	85.0	95.5
$\Phi_{:,102}$	• the most prominent of which was bishop abel muzorewa' s united african national council • ralambo' s father, andriamanelo, had established rules of succession by	African names	99.0	100	100	100
$\Phi_{:,125}$	• music writer jeff weiss of pitchfork describes the" enduring image" • club reviewer erik adams wrote that the episode was a perfect mix	Describing someone in a paraphrasing style. Name, Career	15.5	99.0	100	98.5
$\Phi_{:,184}$	• the world wide fund for nature(wwf) announced in 2010 that a biodiversity study from • fm) was halted by the federal communications commission(fcc) due to a complaint that the company buying	Institution with abbreviation	0	15.5	39.0	63.0
$\Phi_{:,193}$	• 74, 22@,@ 500 vietnamese during 1979 – 92, over 2@,@ 500 bosnian •, the russo@-@ turkish war of 1877 – 88 and the first balkan war in 1913.	Time span in years	97.0	95.5	96.5	95.5
$\Phi_{:,195}$	•s, hares, badgers, foxes, weasels, ground squirrels, mice, hamsters •-@ watching, boxing, chess, cycling, drama, languages, geography, jazz and other music	Consecutive of noun (Enumerating)	8.0	98.5	100	100
$\Phi_{:,225}$	• technologist at the united states marine hospital in key west, florida who developed a morbid obsession for • 00°,11", w, near smith valley, nevada.	Places in US, followings the convention "city, state"	51.5	91.5	91.0	77.5

Table 3: A list of typical mid-level transformer factors. The top-activation words and their context sequences for each transformer factor at layer-8 are shown in the second column. We summarize the patterns of each transformer factor in the third column. The last 4 columns are the percentage of the top 200 activated words and sequences that contain the summarized patterns in layer-4,6,8, and 10 respectively.

gradually developed through several layers.

While some mid-level transformer factors verify common semantic or syntactic patterns, there are also many surprising mid-level transformer factors. We list a few in Table 3 with quantitative analysis. For each listed transformer factor, we analyze the top 200 activating words and their contexts in each layer. We record the percentage of those words and contexts that correspond to the factors' semantic pattern in Table 3. From the table, we see that large

	Adversarial Text	Explaination	α_{35}
(o)	album as "full of exhilarating, ecstatic, thrilling, fun and sometimes downright silly songs"	The original top-activated word and its context sentence for transformer factor $\Phi_{:,35}$ (not an adversarial text)	9.5
(a)	album as "full of delightful, lively, exciting, interesting and sometimes downright silly songs"	Replace the adjectives in sentence (o) with different adjectives.	9.2
(b)	album as "full of unfortunate, heartbroken, annoying, boring and sometimes downright silly songs"	Replace the adjectives in sentence (o) with negative adjectives.	8.2
(c)	album as "full of [UNK], [UNK], thrilling, [UNK] and sometimes downright silly songs"	Mask the adjectives in sentence (o) with unknown tokens.	5.3
(d)	album as "full of thrilling and sometimes downright silly songs"	Remove the first three adjectives in sentence (o).	7.8
(e)	album as "full of natural, smooth, rock, electronic and sometimes downright silly songs"	Replace the adjectives in sentence (o) with neutral adjectives.	6.2
(f)	each participant starts the battle with one balloon. these can be re@-@ inflated up to four	Use a random sentence.	0.0
(g)	The book is described as "innovative, beautiful and brilliant". It receive the highest opinion from James Wood	We create this sentence that contain the pattern of consecutive adjective.	7.9

Table 4: We construct adversarial texts similar but different to the pattern "Consecutive adjective". The last column shows the activation of $\Phi_{:,35}$, or $\alpha_{35}^{(8)}$, w.r.t. the blue-marked word in layer 8.

percentages of top-activated words and contexts do corresponds to the pattern we describe. It also shows most of these mid-level patterns start to develop at layer 4 or 6. More detailed examples are provided in the appendix section F. Though it's still mysterious why the transformer network develops representations for these surprising patterns, we believe such a direct visualization can provide additional insights, which complements the "probing tasks".

To further confirm a transformer factor does correspond to a specific pattern, we can use constructed example words and context to probe their activation. In Table 4, we construct several text sequences that are similar to the patterns corresponding to a particular transformer factor but with subtle differences. The result confirms that the context that strictly follows the pattern represented by that transformer factor triggers a high activation. On the other hand, the closer the adversarial example to this pattern, the higher activation it receives at this transformer factor.

High-level: long-range dependency. High-level transformer factors correspond to those linguistic patterns that span an extended range in the text. Since the IS curves of mid-level and high-level transformer factors are similar, it is difficult to distinguish those transformer factors based on their IS cures. Thus, we have to manually examine the top-activation words and contexts for each transformer factor to differentiate between mid-level and high-level transformer factors. To ease the process, we choose to use the black-box interpretation algorithm *LIME* (Ribeiro et al., 2016) to identify the contribution of each token in a sequence. There also exist interpretation tools that specifically leverage the transformer architecture (Chefer et al., 2021, 2020). In the future, one could adapt those interpretation tools, which may potentially provide better visualization.

Given a sequence $s \in S$, we can treat $\alpha_{c,i}^{(l)}$, the activation of $\Phi_{:,c}$ in layer-l at location i, as a scalar function of s, $f_{c,i}^{(l)}(s)$. Assume a sequence s triggers a high activation $\alpha_{c,i}^{(l)}$, i.e. $f_{c,i}^{(l)}(s)$ is large. We want to know how much each token (or equivalently each position) in s contributes to $f_{c,i}^{(l)}(s)$. To do so, we generated a sequence set $\mathcal{S}(s)$, where each $s' \in \mathcal{S}(s)$ is the same as s except for that several random positions in s' are masked by ['UNK'] (the unknown token). Then we learns a linear model $g_w(s')$ with weights $w \in \mathbb{R}^T$ to approximate $f(s')$, where T is the length of sentence s. This can be solved as a ridge regression:

$$\min_{w \in \mathbb{R}^T} \mathcal{L}(f, w, \mathcal{S}(s)) + \sigma \|w\|_2^2.$$

The learned weights w can serve as a saliency map that reflects the "contribution" of each token in the sequence s. Like in Figure 7, the color reflects the weights w at each position. Red means the given position has positive weight and green means negative weight. The magnitude of weight is represented by the intensity. The redder a token is, the more it contributions to the activation of the transformer factor. We leave more implementation and mathematical formulation details of LIME algorithm in the appendix.

We provide detailed visualization for two different transformer factors that show long-range dependency in Figure 7, 8. Since visualization of high-level information requires more extended context, we only offer the top two activated words and their contexts for each such transformer factor. Many more will be provided in the appendix section G.

We name the pattern for transformer factor $\Phi_{:,297}$ in Figure 7 as "repetitive pattern detector". All top activated contexts for $\Phi_{:,297}$ contain an obvious repetitive structure. Specifically, the text snippet "can't get you out of my head" appears twice in the first example, and the text snippet "xxx class passenger, star alliance" appears three times in the second example. Compared to the patterns we found in the mid-level [6], the high-level patterns like "repetitive pattern detector" are much more abstract. In some sense, the transformer detects if there are two (or multiple) almost identical embedding vectors at layer-10 without caring what they are. Such behavior might be highly related to the concept proposed in the capsule networks (Sabour et al., 2017; Hinton, 2021). To further understand this behavior and study how the self-attention mechanism helps model the relationships between the features outlines an interesting future research direction.

Figure 8 shown another high-level factor, which detects text snippets related to "the beginning of a biography". The necessary components, day of birth as month and four-digit years, first name and last name, familial relation, and career, are all mid-level information. In Figure 8, we see that all the information relates to biography has a high weight in the saliency map. Thus, they are all together combined to detect the high-level pattern.

Figure 7: Two examples of the high activated words and their contexts for transformer factor $\Phi_{:,297}$. We also provide the saliency map of the tokens generated using LIME. This transformer factor corresponds to the concept: "repetitive pattern detector". In other words, repetitive text sequences will trigger high activation of $\Phi_{:,297}$.

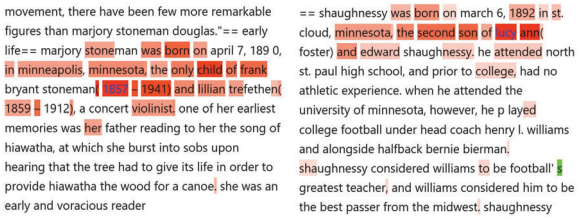


Figure 8: Visualization of $\Phi_{:,322}$. This transformer factor corresponds to the concept: "some born in some year" in biography. All of the high-activation contexts contain the beginning of a biography. As shown in the figure, the attributes of someone, name, age, career, and familial relation all have high saliency weights.

4 Discussion

Dictionary learning has been successfully used to visualize the classical word embeddings (Arora et al., 2018; Zhang et al., 2019). In this paper, we propose to use this simple method to visualize the representation learned in transformer networks to supplement the implicit "probing-tasks" methods. Our results show that the learned transformer factors are relatively reliable and can even provide many surprising insights into the linguistic structures. This simple tool can open up the transformer networks and show the hierarchical semantic or syntactic representation learned at different stages. In short, we find word-level disambiguation, sentence-level pattern formation, and long-range dependency. The idea of a neural network learns low-level features in early layers, and abstract concepts in the later stages are very similar to the visualization in CNN (Zeiler and Fergus, 2014). Dictionary learning can be a convenient tool to help visualize a broad category of neural networks with skip connections, like ResNet (He et al., 2016), ViT models (Dosovitskiy et al., 2020), etc. For more interested readers, we provide an interactive website[1] for the readers to gain some further insights.

Acknowledgements

We thank our reviewers for their detailed and insightful comments. We also thank Yuhao Zhang for his suggestions during the preparation of this paper.

References

Pretrained bert base model (12 layers). https://huggingface.co/bert-base-uncased,

[1] https://transformervis.github.io/transformervis/

last accessed on 03/11/2021.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

Amir Beck and Marc Teboulle. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202.

Hila Chefer, Shir Gur, and Lior Wolf. 2020. Transformer interpretability beyond attention visualization. *CoRR*, abs/2012.09838.

Hila Chefer, Shir Gur, and Lior Wolf. 2021. Generic attention-model explainability for interpreting bimodal and encoder-decoder transformers. *CoRR*, abs/2103.15679.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

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

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah A. Smith. 2015. Sparse overcomplete word vector representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Geoffrey Hinton. 2021. How to represent part-whole hierarchies in a neural network. *arXiv preprint arXiv:2102.12627*.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know. *Trans. Assoc. Comput. Linguistics*, 8:423–438.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B. Viégas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and measuring the geometry of BERT. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, (NeurIPS)*, pages 8592–8600.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, abs/1602.04938.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. *arXiv preprint arXiv:1710.09829*.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. 2019. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

Juexiao Zhang, Yubei Chen, Brian Cheung, and Bruno A Olshausen. 2019. Word embedding visualization via dictionary learning. *arXiv preprint arXiv:1910.03833*.