FINDING ONE MISSING PUZZLE OF CONTEXTUAL WORD EMBEDDING: REPRESENTING CONTEXTS AS MANIFOLD

Anonymous authors
Paper under double-blind review

ABSTRACT

The current understanding of contextual word embedding interprets the representation by associating each token to a vector that is dynamically modulated by the context. However, this "token-centric" understanding does not explain how a model represents context itself, leading to a lack of characterization from such a perspective. In this work, to establish a rigorous definition of "context representation", we formalize this intuition using a category theory framework, which indicates the necessity of including the information from both tokens and how transitions happen among different tokens in a given context. As a practical instantiation of our theoretical understanding, we also show how to leverage a manifold learning method to characterize how a representation model (i.e., BERT) encodes different contexts and how a representation of context changes when going through different components such as attention and FFN. We hope this novel theoretic perspective sheds light on the further improvements in Transformer-based language representation models.

1 INTRODUCTION

In modern natural language processing, using vector representation of words in a low dimension space is a common practice. In early days, word embeddings are static, i.e., the representation is solely determined by a word's identity, such as the case of word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014). In contrast, contextualized word embeddings, e.g., ELMO (Peters et al., 2018), BERT (Devlin et al., 2019) and GPT (Brown et al., 2020), revolutionize this technique by introducing information from the contexts to the central word.

When compared to static embedding, mechanistic studies of the contextual embedding generally lag behind. This is particularly unfavoured as the superior performance of pre-trained contextualized representations is gaining more and more attention. To tackle this problem, previous studies have proposed several directions to explore. For example, researchers in Hewitt & Manning (2019); Coenen et al. (2019); Wu et al. (2020) develop several probing techniques to relate contextualized word embedding to syntactic features. On the other hand, several works try to uncover how contexts influence the geometry of embedding space. Ethayarajh (2019) measures the context-specificity of the word embeddings from different layers of several deep representation models and confirms the majority variance is provided by the context but not the token. They also point out the embedding space is generally anisotropic. Then, results from Cai et al. (2021) indicate that isotropy actually exists in certain isolated clusters of words in the representation space. Mamou et al. (2020) proposes to analyze the manifolds defined by word identity and other linguistic features. Under both "predictive" (i.e., with [MASK] tokens) and "contextualized" (i.e., the real sentence) settings, the authors analyze how well the manifolds can be separated from each other.

While these pioneering works offer some useful insight, their starting points are generally "token-centric", i.e., the analysis focuses on how the context affects the representation of tokens. Therefore, very few studies touch another critical problem, i.e., for contextualized word embedding methods, how exactly does the model represent the context itself. In this study, to fill in this gap, we introduce a novel perspective to analyze the representation of context. To give a rigorous discussion, we first provide a set of mathematical definitions by formulating the language representation problem

using category theory. Then, inspired by manifold learning, we provide two analysis scenarios and also methods to probe the representation of contexts. As a realization of these theoretical analyses, we further provide empirical results and cases studies, which give more insights into the current contextual representation models. A particularly interesting example can be the re-thinking of the functions of Multi-head attention and Feed Forward Network (FFN) in a Transform-based model.

This paper is organized in the following order. In Section 2, we highlight our motivation and the contributions of this research. In Section 3, we provide a preliminary mathematical introduction of contextualized word embedding and our extension. In Section 4, we provide formal definitions and notations in the language of category theory. In Section 5, we explain in detail how to unify the current "token-centric" understanding with our new perspective and introduce two meaningful scenarios where the representation of contexts should play a role. In Section 6, we provide two methods to evaluate our proposed context representation. In Section 7, empirical results are listed. The remaining sections state the related works and the conclusions. The explicit usage of category theory is restricted to Section 4 and 5, as this formulation aims to further clarify our motivation.

2 MOTIVATION AND CONTRIBUTIONS

This work tackles an untouched fundamental problem in deep language learning, i.e., how a representation model actually learns to represent contexts. Therefore, it provides a novel perspective to re-think and characterize these models.

In a contextualized word embedding model, the representation at a specific sequence position is determined by both the token at that place and the surrounding context. However, most of the attention has been paid to one direction, i.e., understanding how the vector represent the token, leaving the context part largely unexplored. Intuitively, we think it may be a good idea to fix the context, mutate the central token, and measure how much is changed about the representation vector of the central token. We hope this disturbance can generate good information that reflects the nature of the contexts.

However, characterizing this kind of information is challenging. Obviously, taking the substituted sequences (and the associated vectors) as a set would miss a lot of information among them. Therefore, we propose a systematic and rigorous mathematical formulation based on category theory which states clearly the representation of context should reflect both the set of representations of the substituted central tokens as well as how these representations can transit to each other (i.e., a reflection of topological features). Both pieces of information are actually relevant to important aspects of language modeling. Furthermore, with of help of manifold learning, we also develop a tool set to analyze the aforementioned information experimentally, from either a functional or a topological perspective.

Besides the above contributions, our empirical results, enabled by these theoretical foundations, also generate several interesting findings. Our analysis indicates that during pre-training, very limited topological information is stored among the contexts surrounding the special token [CLS]. Therefore, its function in specific downstream tasks heavily relies on other task-specific components (e.g., a linear classifier). The fine-turning process, however, injects some of the topological information into the representation model, offering a novel signal to monitor such a process. Last but not least, our dissection of the attention and FFN layers identifies a Game between them when information gets through a deep model, and their final agreement actually supports our hypothesis that they both of them try to fit an identity morphism that reflects *bona fide* linguistic associations.

3 PRELIMINARY: CONTEXTUAL LANGUAGE REPRESENTATION

A contextual language representation (denoted by G in this work) is featured by the integration of information from the whole input sequence to each token (converted from words by tokenization). During a typical masked language model training (Devlin et al., 2019), a token of interest (i.e., the central token, denoted by w) of an input sentence is masked and left to be recovered by this context. In fact, the model takes two pieces of information as input. One is the uncontextualized word embedding of layer 0 (indexed by w), and the other is the observed unmasked tokens (denoted by the observed condition C). Formally, a contextual language representation models the conditional

probability density of the context given the two inputs, i.e.,

$$G(C, w) \sim p(context|C, w). \tag{1}$$

In this work, we propose to represent each observed condition C by inspecting how the model represents various central tokens given this condition, i.e.,

$$P(C) \sim \{G(C, w)|w \in Vac\} \tag{2}$$

where Vac is the set of distinct tokens used by the tokenizer. However, the definition of set only provides a rough description of the nature of concepts, which motivates as to introduce category for a rigorous definition, as described below.

4 Category Theory Formulation and Notations

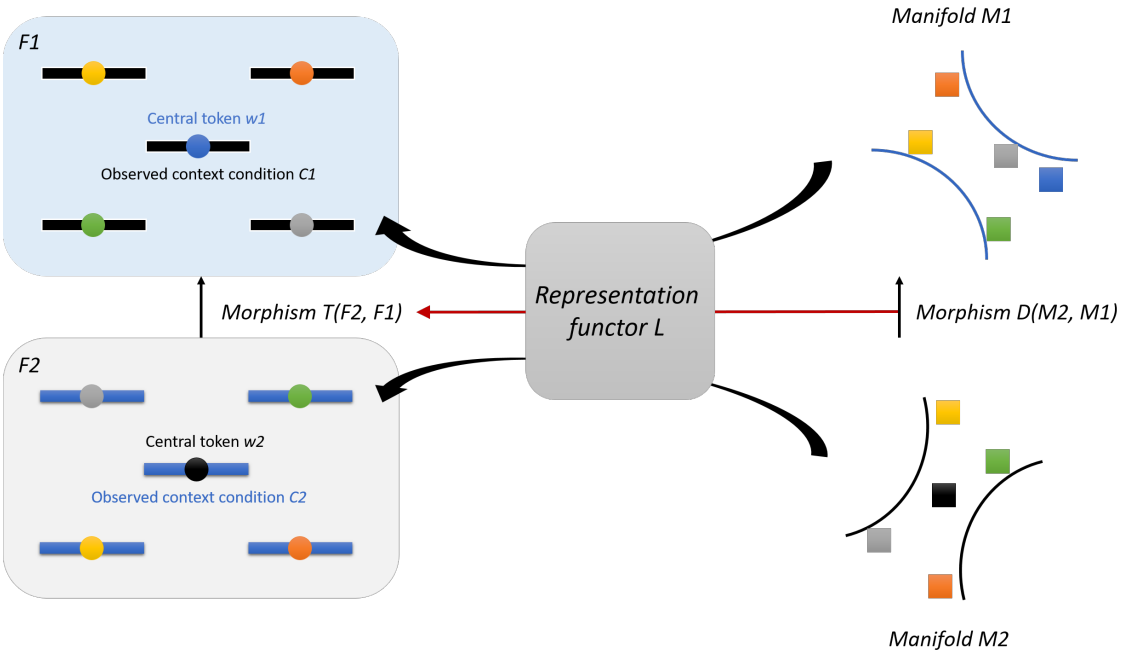


Figure 1: Illustration of the proposed theoretic framework. Please see the main text for more details.

Here, we first explain how to formulate contexts using principles from category theory. Then we further formulate the representation of contexts as a *functor* from the input category to a category of representation space. To begin with, we give the following notations:

Definition 4.1. To encode each context, we first define a finite discrete set K whose each element $k_i : \Omega \to \mathbb{E}$ is a projection from the discrete sample space of tokens $\Omega \in \mathbb{N}^N$ to its σ-field E, where N is set to three, corresponding to token species, token type and token position, respectively. The cardinal number of K is marked as n.

Therefore, each element k_i represents the existence of a certain token (with defined species, type and position) in the context. Following the concepts in language modeling, we introduce the probability density of a context given the observed condition C (e.g., the unmasked tokens in a given sentence), which gives:

Definition 4.2 A context is encoded as a set $F(C)$ each element of which can be expressed as a probability density function f_i conditioning on the observed condition C, i.e., $F(C) = \{f_i(X = e|C)|e \in E)\}$.

Ultimately, one expects the modeling of the context to reveal a comprehensive characterization that transfers well to a general scope of downstream application. Among all the aspects, it is

straightforward to mention the features of each context as well as the relationship between them. This intuition is reminiscent of category theory that explicitly study the *objects* and their mappings (i.e.,*morphisms*).

Definition 4.3 (Category of context probabilities) To define the objects in a category consisting of contexts, we have

$$ob(\mathbb{F}) = \{F(C)|C \in E\} \tag{3}$$

which contains the probability density of all possible context events. Naturally, we can define an abstract collection of morphisms that map from one object to another, i.e.,

$$hom(\mathbb{F}) = \{hom(F_i, F_j)|F_i, F_j \in ob(\mathbb{F})\}, \tag{4}$$

where

$$hom(F_i, F_j) \triangleq \{T_{ij} : F_i \to F_j | T_{ij}(f_k(X = e|C_i, e \in E)) = f_k(X = e|C_j, e \in E)\}\}. \tag{5}$$

Through the above definition, we introduce several constrains that reflect the nature of language representation. First, for each set of morphism between two objects, there only exists one morphism, assuming the model should learn to capture a genuine relationship between contexts. Second, this morphism is a bijection defined on F, so each probability density of context should be projected a unique partner and vice versa (i.e., this morphism would not cause semantic collision). We will discuss how our experimental results support these constrains later.

After the definition of categories in the sample space, we move on to the representation space. In a typical language representation learning setting, a model G is trained to perform injection from the sample space to the representation space, i.e., $G(C) : F \to M$, where M is a d-dimensional Euclidean Space. Therefore, we also extend the aforementioned definition of category of contexts to their representations, which gives the following definition:

Definition 4.4 (Category of vector representations) Suppose we have a set of G models:$G_{set} = G_m$, let

$$ob(\mathbb{M}) = \{M_{C_iG_m}\}, \tag{6}$$

$$hom(\mathbb{M}) = hom(M_{C_iG_m}, M_{C_jG_n})|M_{C_iG_m}, M_{C_jG_n} \in ob(\mathbb{M})\} \tag{7}$$

where

$$M_{C_iG_m} = \{V_k|k \in K, G_m \in G_{set}, V_k \triangleq G_m(f_k(X = e|C_i, e \in E))\}, \tag{8}$$

$$hom(M_{C_iG_m}, M_{C_jG_n}) = \begin{cases} \phi & i \neq j \wedge m \neq n \\ D_{C_iG_m \to C_jG_n} : M_{C_iG_m} \to M_{C_jG_n} & i = j \vee m = n \end{cases}, \tag{9}$$

where

$$D_{C_iG_m \to C_jG_n}(G_m(F(C_i))) = G_n(F(C_j)),$$

and C_i and C_j stand for two different contexts. In this sense, we can define the representation of context as a functor L from this category $\mathbb{M}$ to category $\mathbb{F}$, i.e.,

$$L(G_m(C_i)) = F(C_i), \tag{10}$$

$$L(D_{C_iG_m \to C_jG_n}) = T_{ij}, \forall D_{C_iG_m \to C_jG_n} \in hom(M_{C_iG_m}, M_{C_jG_n}) \tag{11}$$

where $G_m, G_n \in G_{set}$.

5 Understanding contextual language representation under the categorical framework

In this section, we will explain the function of a contextual representation model of natural language G with respect to both objects and morphisms.

5.1 Representing objects of category

Given a certain context, Eq.(10) defines a *class-to-class* mapping of the vector representation given by G to probability density function f. In fact, this definition is consistent with the common understanding of how a representation model works. Actually, the characterization of representations that only consider the mapping of tokens and vectors tend to reflect the nature of object representation. Note that the definition of $\mathbb{M}$ does not require G to be determined, so this definition includes a collection of representations potentially given by different G.

5.2 REPRESENTING MORPHISMS OF CATEGORY

Apart from the "token-centric" understanding of contextual language representation, this work emphasizes on the concept that the learned functor L should also represent the mapping of morphisms from category $\mathbb{M}$ to category $\mathbb{F}$, i.e., the mapping of *intra-class* structure between the two categories. More specifically, the definition given by Eq.(5) and Eq.(7) indicates both the learned model G and observed condition C play a role in the representation of morphisms. Here, we propose to analyze the two factors side-by-side in the following two scenarios. **Scenario A.** The effect of different observed conditions when the representation model G is fixed. This scenario characterizes how the morphism between any two objects in category $\mathbb{M}$ (i.e., two classes of representation vectors) represent their counterparts in category $\mathbb{F}$, with the help of the learned representation model G. This scenario reflects how the model distinguishes different observed conditions around a token of interest, by taking into account that the semantic difference between the input sentences should be represented by some relationship of representation vectors (but not any single vector alone). This understanding offers a novel perspective to evaluate a pre-trained language model's function. **Scenario B.** The representation of a certain observed condition C given different representation models. This scenario extends the understanding of the representation model from a static view (i.e., the model parameters are already learned), to a dynamic view (i.e., the model can change either in configuration or parameter). This gives more implication of our theoretical perspective in dissecting the model components or the learning process. In this work, we select one of these angles. By treating the output given by each layer of a Transformers model as outputs from different G models, we can actually characterize and gain more insights into how the representation of context evolves in a pre-trained language model. Considering the constraints introduced in **Definition 4.3**, different models should all attempt to mimic the same identity morphism, i.e., $id_{F:F\rightarrow F}$, i.e., to preserve most of the original structure of category $\mathbb{F}$ intact. Keep this mind, and we can measure how such a layer of a model conducts this "structural preservation" as a probe for the model's function.

6 EVALUATING CHANGES IN REPRESENTATION OF CONTEXT: A MANIFOLD PERSPECTIVE

The aforementioned reasoning indicates the necessity of characterizing the representation of morphisms as a complement to the characterization of representation vectors alone. Therefore, our characterization focuses how morphisms in category $\mathbb{M}$ reflect the properties of the nature of category $\mathbb{F}$. To facilitate the characterization of structural information, following the manifold assumption (Jolliffe, 2011), we hypothesize that the representation of all the probability distributions of context given by model G lie on a low-dimensional manifold in the d-dimensional Euclidean space. More specifically, the elements of the manifolds are obtained following the protocol below, in the spirit of Eq.2: Given a central word w appearing with a real-world context C, we substitute w with all possible tokens in the vocabulary of the language model (Note that for the sake of computation efficiency, this sampling strategy fixes the dimension of position embedding as in the real context). The collection of sampled elements is denoted as a token substitution set below. With this approximation, while the exact solution of morphisms D is still unsolved, we can leverage comparison of manifold properties to study the behavior of D empirically, i.e.,

$$D^* : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^{n\times n} = Comparison(Q(D(G(\circ))), Q(G(\circ))) \tag{12}$$

where D^* is the observed change in manifold property that is induced by D, Q is certain characterization of the manifold. Then we can analyze the representation of the real context through 1) a task-related projection of the manifold, and 2) the topology output by a manifold learning algorithm.

Functional characterization When comparing two contexts, a straightforward way is to project the representation to task-related features. In practice, one can choose to a linear head that is either pre-trained along with the representation model, or fine-tuned for a specific downstream task. In fact, this projection is an Isomorphism that preserves the topological structure of the manifold, thus providing a task-related evaluation for the manifold consisting of context probability.

Topological characterization To further characterize the topology of the proposed manifold, we can also leverage the toolset from manifold learning community to reveal the local features. In practice, we perform an inspection of the existence of topological associations in the local region.

Particularly, here we borrow the method for topology determination from a prominent manifold learning algorithm, i.e., UMAP (McInnes & Healy, 2018). The local topology is determined among k-nearest neighbors, and represented by a fuzzy set to indicate whether an edge between nodes exists. Please refer to the original UMAP paper for a more detailed description. Note that as UMAP does not offer an estimation in global geodesic distances for arbitrary two elements (it focuses on local topology), our analysis also does not include this part. Nevertheless, we show below that the local information alone can provide some interesting insights for language modeling.

7 EXPERIMENTS

7.1 MODELS AND SETTINGS

In this study, to provide insights originating from our new understanding, we analyze the word representation of a widely used contextualized embedding model, i.e., BERT (Devlin et al., 2019). The implementation, feature extraction and pre-trained model ("Bert-base-uncased") are all based on the Transformers package [1].

Our Analysis involves two datasets. Wikitext-2 (Merity et al., 2017) is used as a representative of pre-training corpus, while SST-2 (Socher et al., 2013), one of the GLUE dataset (Wang et al., 2018), is used as a typical downstream task (i.e., sentiment analysis of movie reviews) dataset. Detailed statistics of data used in the analysis will be provided in the corresponding sections. To fine-tune the model for the SST-2 task, we use a learning rate of $2e^{-5}$, batch size of 32, and epoch number of 3. For the topology analysis using UMAP, we keep the original hyperparameter of kNN as 15.

In the two subsections below, we provide some empirical results corresponding to the two scenarios defined in Section 5.

7.2 UNDERSTANDING DIFFERENT REPRESENTATION OF CONTEXTS FROM A PRE-TRAINED MODEL

Star graph as a unit of analysis. To give a straightforward impression of the topology of a context manifold, here we first list an example (Figure 2(a)). The example context comes from Wikitext-2. Here, we visualize the connections of the real context to its neighbors, as determined by the local fuzzy set as in UMAP. Note that each context here is associated with a set of substitutions of an original token to other tokens. By treating the original sentence as the center, the topological evaluation can be conducted by examining the star-graph centered by the original context.

Vanishing of local topology: a case study of [CLS]. To provide a more task-related perspective, we also study how contexts are represented for the special token [CLS], whose representation is often used as input for natural language classification. Note that in correspondence to our formulation, the token substitution set's representations are projected, together with the [CLS] token's representation itself, by the linear classification layer that is learned during fine-tuning. Note that while projection preserves the impact of the original topology, its results are scalar values to indicate the classification results. In this sense, the mean score of positive and negative samples can indicate the power of the classifier, while the potential influence of the context can be expressed by the variance of the token substitution set when projected to the logits of binary classification.

In light of this understanding, we plot the classification score (for the negative class) of each sample (denoted by the dots), together with the variance (denoted by the radius of dots) of scores when using the substitution contexts as input (Figure 2(b)). Given the good classification performance (accuracy of 92.66), the scores of positive and negative sample almost separate perfectly well. Interestingly, when compared with the margins of positive and negative scores with respect to the boundary (i.e., 0.5), the variance values are with very limited scale. This result indicates that, when replacing the [CLS] tokens with other tokens, the representation of the whole sentence does not change a lot. From our understanding, this observation can be attributed to the pre-training process. More specifically, considering that the input is actually a triplet of token identity, token position and toke type, all contexts around [CLS] only learn to mark the most repetitive pattern, i.e., there is something at the first position. Therefore, our results clearly indicate the local topology among token substitution set

[1] https://huggingface.co/transformers/

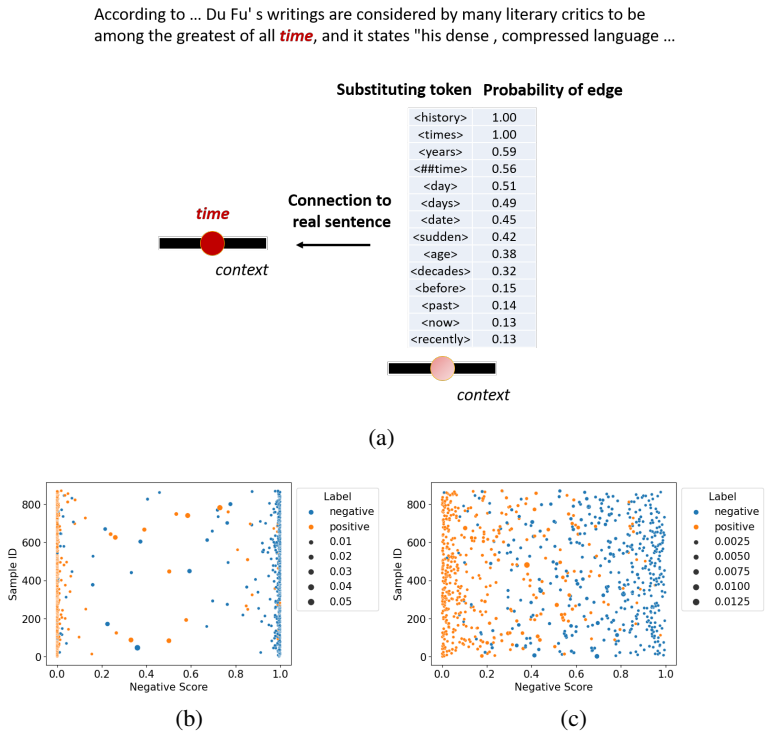


Figure 2: (a) An example of star graph where each element in the token substitution set has a probability of connecting to the real context, as determined by the local fuzzy set as in UMAP. (b) The effect of local topology of [CLS] token after fine-tuning, as evaluated by functional characterization. (c) The effect of local topology of [CLS] token before fine-tuning, as evaluated by functional characterization. For (b) and (c), the radius of a dot denotes the variance caused by local topology. The larger the radius is, the larger the variance will be.

around [CLS] does not play a significant role in language modeling. This is also consistent with the general understanding of [CLS] as a "pooling operator" for arbitrary sentences.

On the other hand, we also include the corresponding results obtained by a linear classifier on the top of a frozen BERT model for comparison, thus reflecting the situation of the pre-training stage. As shown in (Figure 2(c)), we do observe an expansion of variance of the token substitution set after fine-tuning. Since the scale of variance with respect to the classification margin reflects how similar contexts affect the classification results, this expansion actually indicates the gaining of task-oriented topological information and the degeneration of pre-training patterns. Note that on the y-axis, we plot the sample id, which shows no association with the mean or variance, further verifying the effect is a characteristic of [CLS] token but not any specific context.

7.3 Evolution of context representation through a Transformer model

In this analysis, our focus is to characterize how the context manifold from each layer of a pre-trained Transformers model differs from each other. As this does not consider any specific downstream tasks, we use the test set of Wikitext-2 dataset, which mimics a pre-training corpus of BERT. For the visualization purpose, we sample the 10 most frequent tokens from each of the three parts of speech (i.e., noun, verb and adjective) respectively. For each token as the token of interest, we further sample 10 samples and calculate the topology around each context.

Upper layers overwrite the context manifold topology given by lower layers. Considering the information flow in a Transfomer-based model, each token is gradually exposed to more "integrated" information when layers move forward. When considering the representation vector alone, this can be understood as the addition (with attention weights) of other token's vector to the vector of the

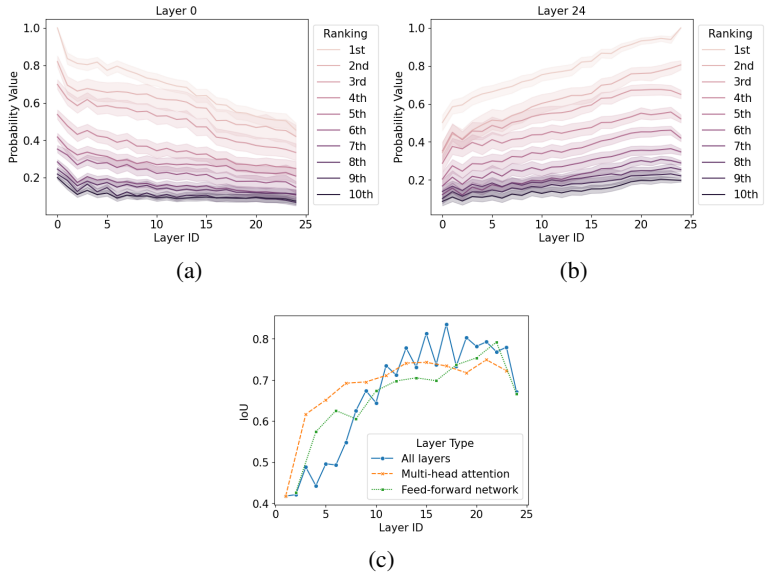


Figure 3: (a) The disappearance of neighbors in the manifold given by the first layer. (b) The emergence of neighbors in the manifold given by the last layer. The y-axis is the average probability of connection and the band width is the variance. Note that we treat the embedding layer as the layer 0 and separately count the attention and FFN layers in one Transformer layer. Therefore, layer 24 denotes the last FFN layer. (c) The IoU trend between different layers. Each IoU value is calculated with last layer or last layer of the same type. The results of tokens with different POS can be found in the Appendix.

central token. Here, we propose to analyze this change on a context-level, i.e., how the association among the context neighborhoods reflects morphisms between semantics in the discrete language space. Firstly, we examine the trajectories of evolution for the nearest neighbors appearing in the first layer (i.e., the static embedding layer) and the last layer (i.e., the output layer) (Figure 3(a) and 3(b)). This is done by calculating the average probability for the existence of an edge that connects to the real context based on how the tokens rank in the first or last layer. Clearly, we observe the gradual disappearance (loss of connection) of the neighbors in the embedding layer, as well as the emergence of new, sample-aware neighbors in the last layer. This result further validate the relevance of "context manifold" in language modeling. The corresponding results for other layers can be found in Appendix.

Representation Learning converges when the game between Multi-head attention and Feed-Forward Networks ends. When inspecting the manifold change for specific samples, we realize an interesting balance between the attention layers and FFN layers. By statistics, we also plot the average results here (Figure 3(c)). In this experiment, we try to reveal how neighbors given by different layers are similar with each other, measured by intersection-over-union (IoU). First we have a line possessing the sequential comparison of all layers, i.e., each layer has an IoU calculated with the last layer. Interestingly, while the consistency of layers generally goes up as layers get higher, the IoUs show a trend of "zigzag" especially during the first two and upper half of layers. This result means the attention and FFN layers tend to give different results and the model tries to get an agreement between these two layer types. Inspired by this understanding, we also calculate the IoUs solely between attention or FFN outputs (the first embedding layer serves as the starting point for both series). These results further reveal some detailed insights among the Transformer layers. At the first several layers, the results between attention and FFN layers possess a higher similarity than results between them, verifying the existence of debate. Also, as the attention layers generally have a higher consistency, a natural understanding is they tend to inject some uniform information from the whole input sequence, while the FFN layers tend to eliminate the information they don't agree (as they contain ReLU non-linearity). At the upper layers, in contrast, the overall consistency between attention and FFN layers surpasses the consistency between the layers with the same type,

suggesting the discrepancy in the lower layers actually converges to a better agreement. The above observations actually support our hypothesis that different layers in a model (treated as different G here) learns to achieve an identity morphism in $\mathbb{F}$, which reflects the genuine linguistic association between the probability distributions of different contexts (This result holds true for tokens from all the three parts of speech we samples). In other words, topological consistency between the attention layers and the FFN layers is a necessary condition of model convergence. In addition, we also notice that the last FFN layer seems to decrease the overall consistency a lot, which we speculate is due to the direct connection to a linear head (Ma et al., 2019).

8 Related Work

The understanding of contextual word embedding is a field gaining more and more attention (Rogers et al., 2020). However, it is generally regarded as a challenging task considering the complex interplay between different words. One line of researches proposes to train a probing network on the top of pre-trained word embedding, so as to find some relationship between the learned representation and the linguistic features such as syntactic tree (Hewitt & Manning, 2019; Coenen et al., 2019; Miaschi & Dell'Orletta, 2020). The process can also be realized in parameter-free fashion (Wu et al., 2020).

Another line of research in this field follows the basic idea of information geometry. As each token is associated with a vector, analysis of the clustering of these vectors can generate insightful findings, such as the geometry of polysemous words (Coenen et al., 2019), isotropy in embedding space (Ethayarajh, 2019; Cai et al., 2021; Rajaee & Pilehvar, 2021). Moreover, there are works that utilize the geometric information for better model performance, possibly through refinement (Chu et al., 2019; Hasan & Curry, 2017). While our method borrows some practice from information geometry, this work is fundamentally different from the above in the way that we try to characterize the nature of contexts, while these studies try to understand how a representation vector reflects the nature of a token.

9 Conclusions

In this work, we propose a novel perspective to understand the contextualized word embedding, with an emphasis on the context rather than the central token. With rigorous mathematic formulation, analysis method development, and empirical results, we hope our study can offer the community some new insights into the popular pre-trained language models.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. Isotropy in the contextual embedding space: Clusters and manifolds. In *ICLR*, 2021.

Yonghe Chu, Hongfei Lin, Liang Yang, Yufeng Diao, Shaowu Zhang, and Xiaochao Fan. Refining word representations by manifold learning. In *IJCAI*, 2019.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

Kawin Ethayarajh. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. *ArXiv*, abs/1909.00512, 2019.

Souleiman Hasan and Edward Curry. Word re-embedding via manifold dimensionality retention. In *EMNLP*, 2017.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *NAACL*, 2019.

Ian T. Jolliffe. Principal component analysis. In *International Encyclopedia of Statistical Science*, 2011.

Xiaofei Ma, Zhiguo Wang, Patrick Ng, Ramesh Nallapati, and Bing Xiang. Universal text representation from bert: An empirical study. *ArXiv*, abs/1910.07973, 2019.

Jonathan Mamou, Hang Le, Miguel Angel del Rio, Cory Stephenson, Hanlin Tang, Yoon Kim, and SueYeon Chung. Emergence of separable manifolds in deep language representations. *ArXiv*, abs/2006.01095, 2020.

Leland McInnes and John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. *ArXiv*, abs/1802.03426, 2018.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *ArXiv*, abs/1609.07843, 2017.

Alessio Miaschi and F. Dell'Orletta. Contextual and non-contextual word embeddings: an in-depth linguistic investigation. In *REPL4NLP*, 2020.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

S. Rajaee and Mohammad Taher Pilehvar. A cluster-based approach for improving isotropy in contextual embedding space. In *ACL/IJCNLP*, 2021.

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

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *ArXiv*, abs/1804.07461, 2018.

Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. Perturbed masking: Parameter-free probing for analyzing and interpreting bert. In *ACL*, 2020.

A APPENDIX

A.1 THE OVERWRITING OF LOCAL TOPOLOGY IN DIFFERENT TRANSFORMER LAYERS

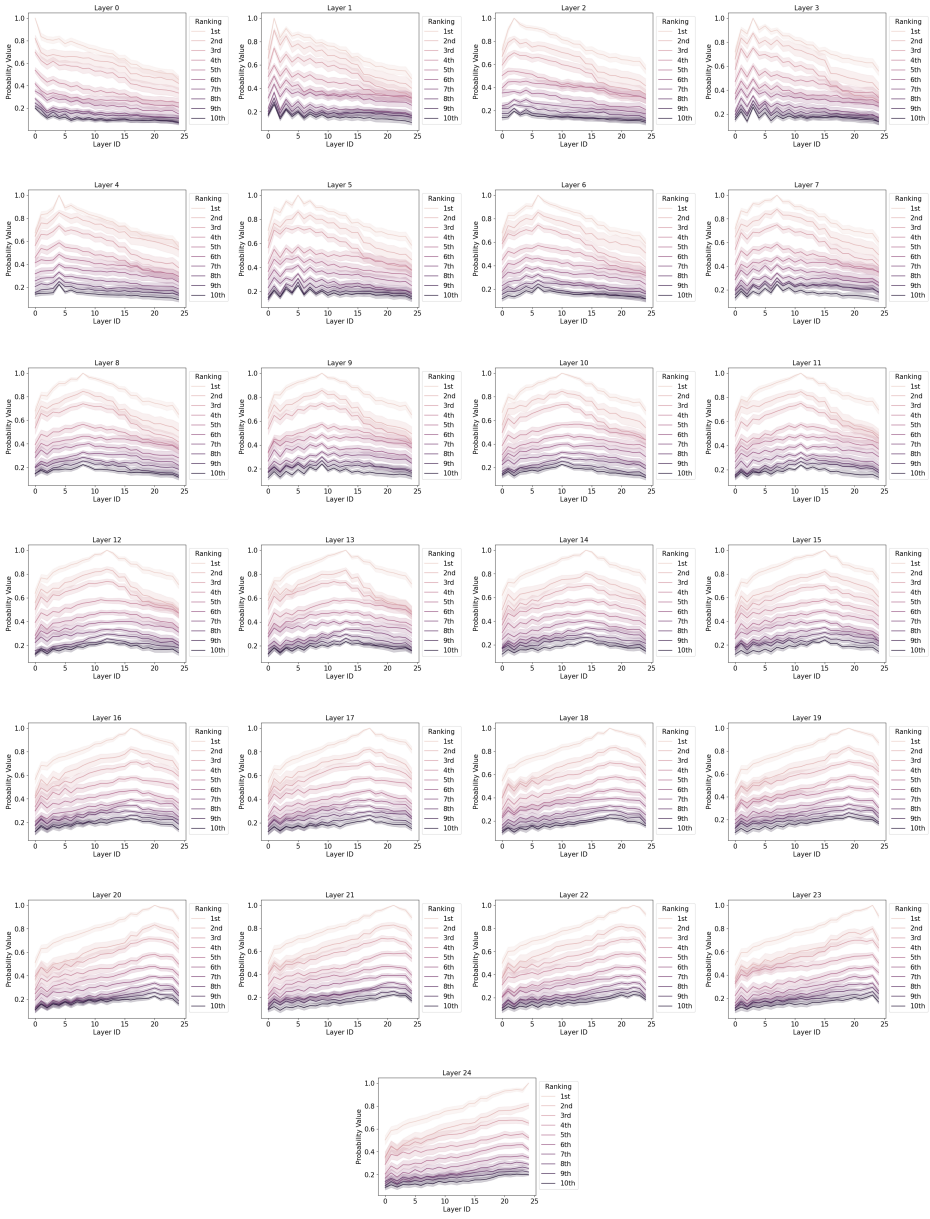
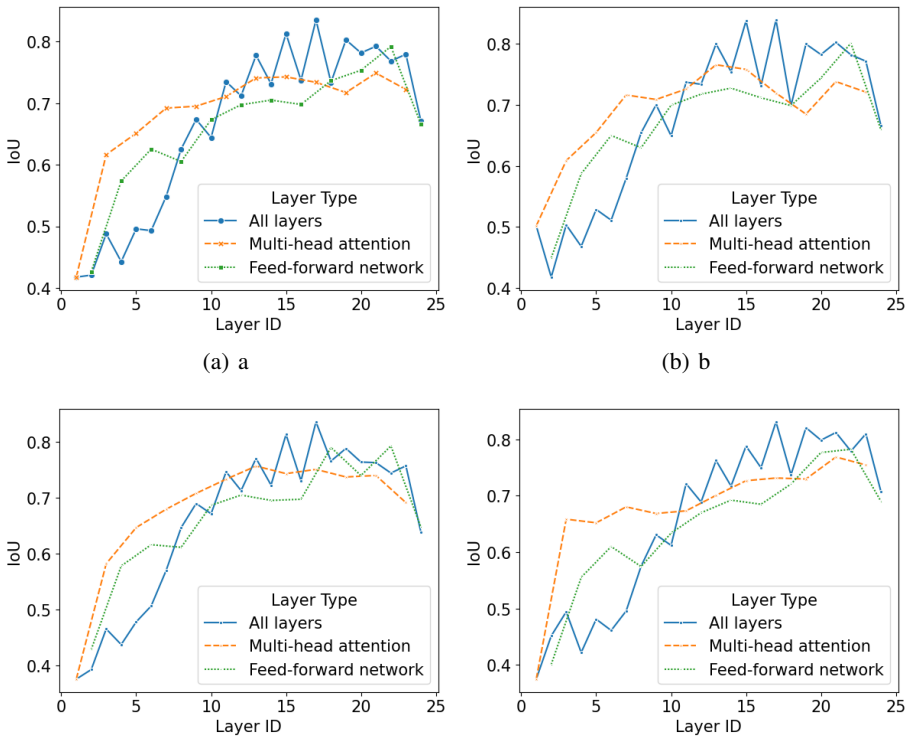


Figure 4: The overwriting of local topology in different Transformer layers. Note that we treat the embedding layer as the layer 0 and separately count the attention and FFN layers in one Transformer layer. Therefore, layer 24 denotes the last FFN layer. We record changing trajectory of the top 10 neighbors of the real context at each layer.

A.2 THE GAME BETWEEN THE MULTIHEAD ATTENTION LAYERS AND FFN LAYERS



(a) a (b) b (c) c (d) d

Figure 5: The IoU trend results of different parts of speech. (a) All. (b) Adjective. (C) Noun. (d) Verb. We sample the top 10 most frequent tokens of each POS and get the average score.