

TOPOLOGICAL COLLOCATION AND MULTIWORD EXPRESSION EXTRACTION

AMELIE SCHREIBER

ABSTRACT. In this note, we discuss computing the persistent homology of context vectors associated to a certain keyphrases in text corpora, such as collocations, multiword expressions (MWEs), and idioms. We discuss the behavior of several multilingual language models, and several monolingual language models, and we discuss how well they preserve the persistent homology across different contexts, across different languages, and for different individual attention heads. We perform the analysis of how well an attention head preserves the persistent homology in terms of the bottleneck and Wasserstein distance metrics on persistence diagrams and find that the Wasserstein distance is a better metric for the analysis. In our preliminary experiments we find clear trends in performance where certain models, and certain heads, tend to preserve the persistent homology of the collocations and MWEs better than others across contexts, and across languages for multilingual models, validating the need for further large scale study. We also present the idea of including persistent homology in the training objective to improve performance on tasks such as translation, as models known to perform better at translation also consistently perform better at preserving the persistent homology of keyphrases like collocations and idioms. We also discuss the potential of using the Fréchet mean of persistence diagrams as a kind of average topological structure which might serve this purpose, where minimizing the distance of the persistence diagram of a collocation or MWE to the Fréchet mean diagram could provide a meaningful topological inductive bias when learning contextual embeddings for individual attention heads.

CONTENTS

1. Introduction	1
2. Context Vectors	2
3. Application of Clustering Algorithms on Context Vectors	4
4. Semantic and Syntactic Analysis and Model Interpretability	5
5. Topological Structures that Persist in Different Contexts	6
6. Some Heuristics: Comparing Barcode Diagrams	7
7. Fréchet Mean of Persistence Diagrams	8
References	9

1. INTRODUCTION

1.1. Multiword Expressions, Collocations, Idioms, and Keyphrases. While we go beyond idioms, considering multiword expressions and collocations that have no hidden non-literal meanings, they serve as a good example of the kinds of keyphrases we wish to focus on in text corpora. They represent groups of words that appear together more often than would be expected by chance, and may have a meaning which is not easily understood from the meanings of the individual words that

Date: June 10, 2023.

2020 Mathematics Subject Classification. Primary 68T07, 68T45, Secondary 68T50 68T10,

make them up. They also serve as a difficulty in the context on machine understanding of text and machine translation. So, we would like to study how they behave from a new perspective. While we do not restrict ourselves to Chinese, and take a more multilingual approach, it serves as a good example:

"Idioms are different from ordinary Chinese characters in that they have semantic unity and structural persistence; that is, they are semantically indivisible as a whole, and their overall meaning cannot be speculated from the individual words that make up the idiom. In terms of structure, the order cannot be changed casually, let alone the grammatical structure. This leads to the low accuracy of the existing models for idiom machine reading."[DLYF]

In [DLYF], the authors present an architectural modification to the usual transformer that captures a somewhat more hierarchical representation of idiomatic expressions. However, rather than propose an architectural change to the transformer model, we propose a modification of the objective, possibly only for Low Rank Adaptation (LoRA) of a pretrained model [Hu et. al.], for downstream tasks where extracting and understanding collocations and multiword expressions such as idioms might be important, such as machine translation. In particular, we propose including the persistent homology of collocations and multiword expressions (MWEs) in the objective for LoRA modified attention heads.

2. CONTEXT VECTORS

The context vector is a critical part of the multihead attention mechanism in transformers. As indicated by the provided formula:

$$\mathbf{c}_i^{l,h} = \sum_{j=1}^n a_{i,j}^{l,h} \mathbf{v}_j^{l,h}$$

the context vector $\mathbf{c}_{i,l}^h$ for the h^{th} head in layer l is computed as a weighted sum of the value vectors $\mathbf{v}_j^{l,h}$, where the weights are the attention scores $a_{i,j}^{l,h}$. Here, n is the number of value vectors, corresponding to the number of input tokens for the attention mechanism.

The attention scores $a_{i,j}^{l,h}$ for attention head k in layer l , themselves are computed using the query and key vectors. In the scaled dot-product attention mechanism typically used in transformers, the attention score between the i^{th} query $\mathbf{q}_i^{l,h}$ and the j^{th} key $\mathbf{k}_j^{l,h}$ is computed as:

$$a_{i,j}^{l,h} = \frac{\exp(\mathbf{q}_i^{l,h} \cdot \mathbf{k}_j^{l,h} / \sqrt{d})}{\sum_{m=1}^n \exp(\mathbf{q}_i^{l,h} \cdot \mathbf{k}_m^{l,h} / \sqrt{d})}$$

where d is the dimensionality of the queries and keys, and \cdot denotes the dot product. The division by \sqrt{d} is a scaling factor that is used to prevent the dot product from growing too large in magnitude, which could lead to vanishing gradients during training. The softmax function is applied to the raw attention scores to ensure that they sum up to 1, allowing them to be interpreted as probabilities or relative importances.

The queries, keys, and values are themselves computed by applying learned linear transformations to the input embeddings. If \mathbf{x}_i denotes the input embedding for the i^{th} token, then we have:

$$\mathbf{q}_i^{l,h} = W_Q^{l,h} \mathbf{x}_i$$

$$\mathbf{k}_i^{l,h} = W_K^{l,h} \mathbf{x}_i$$

$$\mathbf{v}_i^{l,h} = W_V^{l,h} \mathbf{x}_i$$

where $W_Q^{l,h}$, $W_K^{l,h}$, and $W_V^{l,h}$ are the weight matrices for the queries, keys, and values, respectively, for the k^{th} head in the l^{th} layer.

The context vectors $\mathbf{c}_{i,l}^h$ provide a summary of the input tokens, weighted by their relevance to the query. They can be thought of as a form of "contextualized" embedding, where the context is determined by the other tokens in the input sequence and their interaction weight given by the attention matrix $a_{ij}^{l,h}$ for head $h \in \{1, 2, \dots, \mathcal{H}\}$. The multihead attention mechanism allows the model to capture different types of relevance or "attention" by using multiple heads, each with its own learned linear transformations.

In the multihead attention mechanism, the attention operation is not performed just once, but multiple times in parallel. The queries, keys, and values are transformed with different learned linear projections to h different sets of queries, keys, and values, where h is the number of heads. Then the attention mechanism is applied to each of these sets, yielding h output vectors, which are then concatenated and linearly transformed to result in the final output.

Let $W_Q^{l,h}$, $W_K^{l,h}$, and $W_V^{l,h}$ denote the weight matrices for the h^{th} head for the queries, keys, and values, respectively, in the l^{th} layer, and let W_O^l denote the output weight matrix for layer l . Then the output of the multihead attention mechanism for layer l is computed as:

$$\mathbf{c}_i^l = W_O^l [\mathbf{c}_i^{l,1}; \mathbf{c}_i^{l,2}; \dots; \mathbf{c}_i^{l,\mathcal{H}}]$$

where $\mathbf{c}_i^{l,h}$ is the output of the attention mechanism for the h^{th} head in layer l , for token x_i , computed as:

$$\mathbf{c}_i^{l,h} = \sum_{j=1}^n a_{ij}^{l,h} \mathbf{v}_j^{l,h}$$

Each transformer layer comprises a multihead self-attention mechanism, followed by layer normalization, a position-wise feed-forward network, and another layer normalization. Let's denote the l^{th} transformer layer in the model, where l ranges from 1 to L (with L being the total number of layers).

- (1) Multihead Self-Attention Mechanism: The multihead self-attention mechanism in the l^{th} layer operates on the input embeddings \mathbf{x}_i^l , transforming them into queries $\mathbf{q}_i^{l,h}$, keys $\mathbf{k}_i^{l,h}$, and values $\mathbf{v}_i^{l,h}$ for each head h as before. Then the context vectors are computed, and concatenated as before, and the weight matrix W_O^l is applied to get \mathbf{c}_i^l
- (2) Layer Normalization: Layer normalization stabilizes the learning process and reduces internal covariate shift by normalizing the multihead self-attention output across the hidden dimension. For the l^{th} layer, this is computed as:

$$\mathbf{c}_i'^l = \frac{\mathbf{c}_i^l - \mu^l}{\sigma^l}$$

where μ^l and σ^l are the mean and standard deviation of the layer outputs, computed as:

$$\mu^l = \frac{1}{H} \sum_{i=1}^H c_i^l$$

$$\sigma^l = \sqrt{\frac{1}{H} \sum_{i=1}^H (c_i^l - \mu^l)^2}$$

- (3) **Position-Wise Feed-Forward Network (FFN):** The output of the layer normalization is then passed through a position-wise feed-forward network (FFN). The

The output of the layer normalization is then passed through a position-wise feed-forward network (FFN). The FFN consists of two linear transformations with a ReLU activation in between. For each position i , the FFN is applied to \mathbf{c}_i^l independently. Let's denote the weight matrices and bias vectors of the two linear transformations as W_1^l , b_1^l , W_2^l , and b_2^l , respectively. Then the output of the FFN is computed as:

$$\mathbf{d}_i^l = W_2^l \max(0, W_1^l \mathbf{c}_i^l + b_1^l) + b_2^l$$

where $\max(0, x)$ denotes the ReLU activation function.

- (4) **Second Layer Normalization:** Finally, the output of the FFN goes through another layer normalization step to compute the final output of the transformer layer:

$$\mathbf{o}_i^l = \frac{\mathbf{d}_i^l - \mu^l}{\sigma^l}$$

where again, μ^l and σ^l are the mean and standard deviation of the layer outputs, computed similarly as before but now with \mathbf{d}_i^l instead of \mathbf{c}_i^l .

3. APPLICATION OF CLUSTERING ALGORITHMS ON CONTEXT VECTORS

In the field of computational linguistics, the concept of context vectors, as generated by models like Transformers, has proved to be critical in capturing the semantic and syntactic nuances of language. These context vectors, which provide a representation of each token in the input sequence in terms of its context, can be further processed to extract meaningful patterns and relationships. In this section, we will discuss the application of clustering algorithms, specifically Density-Based Spatial Clustering of Applications with Noise (DBSCAN), and its topological data analysis cousin, persistent homology, on these context vectors.

3.1. Context Vectors and Clustering. Given a Transformer model with L layers and \mathcal{H} attention heads, for any token x_i in the input sequence, we can obtain a set of $L \times \mathcal{H}$ context vectors $\mathbf{c}_i^{l,h}$, with each context vector corresponding to a different attention head in a different layer.

For a given layer l and head h , the context vector $\mathbf{c}_i^{l,h}$ is a d -dimensional vector (where d is the dimensionality of the embeddings), which can be thought of as a point in a d -dimensional space. Thus, for all tokens in the input sequence, we can obtain a set of points in this d -dimensional space.

Clustering can be applied to this set of points to group together tokens that have similar context vectors, and hence, are likely to have similar semantic and syntactic roles. This can be useful for various applications, such as identifying synonyms, identifying words that often occur in similar contexts, or even for interpretability, by identifying what types of tokens each head and layer of the transformer is paying attention to.

3.2. Density-Based Spatial Clustering of Applications with Noise (DBSCAN). DBSCAN is a density-based clustering algorithm that groups together points that are densely packed in the input space. The algorithm is defined by two parameters: ϵ , which defines the neighborhood around a point, and $minPts$, which defines the minimum number of points required to form a dense region.

Given a set of n context vectors $\mathbf{c}_i^{l,h}$, $i \in \{1, 2, \dots, n\}$, the DBSCAN algorithm can be applied as follows:

- (1) For each point $\mathbf{c}_i^{l,h}$, compute the ϵ -neighborhood $N_\epsilon(\mathbf{c}_i^{l,h})$, which is the set of points that are within a distance ϵ of $\mathbf{c}_i^{l,h}$.

- (2) For each point $\mathbf{c}_i^{l,h}$, if $|N_\epsilon(\mathbf{c}_i^{l,h})| \geq \text{minPts}$, mark $\mathbf{c}_i^{l,h}$ as a core point. Otherwise, if $\mathbf{c}_i^{l,h}$ is within the ϵ -neighborhood of some core point, mark $\mathbf{c}_i^{l,h}$ as a border point. All other points are marked as noise points.
- (3) Form a cluster for each core point and its associated border points. Two core points are in the same cluster if they are within a distance ϵ of each other.

The resulting clusters group together tokens with similar context vectors, which can be used to identify patterns and relationships in the input data.

3.3. Persistent Homology. Persistent homology is a method from the field of topological data analysis that can be used to identify and quantify the shape or structure of a dataset. It can be thought of as a multi-scale analysis that identifies "features" of the data that persist over a range of scales.

Given a set of n context vectors $\mathbf{c}_i^{l,h}$, $i \in \{1, 2, \dots, n\}$, the persistent homology analysis can be carried out as follows:

- (1) Construct a simplicial complex K from the context vectors, where each vector is a 0-simplex (a point), and a k -simplex (a k -dimensional triangle) is formed whenever $k + 1$ points are within a distance ϵ of each other.
- (2) For each $\epsilon \geq 0$, compute the homology groups $H_k(K_\epsilon)$ of the ϵ -sublevel set K_ϵ , which is the subcomplex of K consisting of all simplices that can be formed with a distance less than ϵ .
- (3) For each homology group $H_k(K_\epsilon)$, compute the Betti numbers $\beta_k = \text{rank}(H_k(K_\epsilon))$, which count the number of k -dimensional "holes" in the data.

The result of this analysis is a set of "persistence diagrams", one for each k , which plot the birth and death scales of each feature (where the birth scale is the smallest ϵ at which the feature appears, and the death scale is the smallest ϵ at which the feature disappears).

These persistence diagrams can be used to identify and quantify the "shape" or "structure" of the context vectors. For instance, a high β_0 indicates a large number of disconnected clusters, a high β_1 indicates a large number of loops or cycles, and so on.

3.4. Applications. Clustering of context vectors using methods like DBSCAN and persistent homology can have a variety of applications in computational linguistics and natural language understanding:

- (1) **Semantic and Syntactic Analysis:** Clusters of context vectors can reveal groups of tokens that are used in similar semantic or syntactic roles. For example, words that are synonyms or words that belong to the same part of speech might cluster together.
- (2) **Model Interpretability:** Clustering can help to understand what features each head and layer of the transformer model is paying attention to. For example, certain heads might focus more on syntactic features, while others might focus more on semantic features.
- (3) **Anomaly Detection:** Outliers in the context vectors, which would be identified as noise points by DBSCAN or as short-lived features by persistent homology, can indicate tokens that the model is having difficulty processing.
- (4) **Model Compression:** Clustering can be used to identify redundancies in the model, such as heads or layers that produce similar context vectors with similar persistent topological features, which can be removed to reduce the size of the model without significantly impacting performance.

4. SEMANTIC AND SYNTACTIC ANALYSIS AND MODEL INTERPRETABILITY

In preliminary studies, it has been shown that using the persistent homology of a single attention head's context vectors can extract collocations and keyphrases from the text, providing a connection between analyzing the linguistic properties of the text and interpreting the model's behavior. To be

more precise, it can be shown empirically that using a persistent homology barcode diagram, obtained from the persistent homology analysis of the context vectors computed by a single attention head, can be used to inform which distance threshold parameter to choose for a DBSCAN algorithm (also applied to the context vectors). Using this persistent homology informed BDSCAN of the context vectors, clusters of collocations and keyphrases we detected at a specific scale which correspond to low-lying H_1 persistence structures. The majority of these clusters contained few or no extraneous words and were made up only of collocations.

5. TOPOLOGICAL STRUCTURES THAT PERSIST IN DIFFERENT CONTEXTS

Suppose we are given a collocation or multiword expression (MWE) that appears in different contexts. The question arises, does the group of tokens representing the collocation or MWE have a topological structure that persists in different contexts? In other words, if we take the simplicial complex associated to the collocation or MWE, does the 1-skeleton with edges connecting nodes corresponding to tokens in our collocation or MWE, do the pairwise distances remain similar in multiple contexts? What about across languages before and after translation? As an example, take the multiword expression *"the tip of the iceberg"* in the following paragraph:

*"In the realm of linguistics, it is crucial to foster a deep understanding of language structures and their interconnections. This understanding is not simply about memorizing vocabulary or grammar rules, but delving into the intricate details of phonetics, semantics, and syntax. However, the common elements that we often focus on, such as morphemes, phonemes, and syntax rules, are merely **the tip of the iceberg**. Beneath the surface, there lies a vast field of cognitive, social, and historical factors that play significant roles in shaping a language. Grasping these underlying influences is what sets apart a competent linguist from a casual language learner."*

and in the paragraph:

*"In the realm of deep learning, having a deep understanding of neural networks and their architectures is of paramount importance. It's not just about knowing how to code an algorithm, but also about comprehending how different layers in a network interact and learning to interpret the output. Yet, the algorithms and architectures we often discuss, like Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), are just **the tip of the iceberg**. Hidden below the surface are complex concepts like backpropagation, weight initialization, and optimization techniques, which truly drive the functionality of these networks. Mastering these intricacies is what differentiates an expert in the field from an amateur coder."*

How does the topology and the pairwise distances between the nodes representing the tokens for *"the tip of the iceberg"* change when we study in the context of the first paragraph versus the second. How does do different language translation models compare to one another in preserving this topology? What happens to the topological structure as we change languages? Is the topology easier to preserve in one language than in another? How do different heads both within models and across models compare to each other? These are some of the questions we aim to answer in this preliminary study.

Understanding the persistent homology of the 1-skeleton associated to a collocation or MWE is a difficult task. With persistent homology we can compare the persistence diagrams (or equivalently the barcode diagrams), associated to the distance matrix of pairwise Euclidean distances between the context vectors associated to the tokens. Comparing barcode diagrams requires a notion of a distance metric between barcode diagrams. We use both the bottleneck distance, as well as the Wasserstein distance¹ to compare barcode diagrams in our preliminary analysis. We also compare

¹Wasserstein distance is also called the Kantorovich distance.

two multilingual models, **xlm-roberta-large**² and **bert-base-multilingual-cased**³. We also compare seven languages, some of which are considered morphologically complex, and some which are considered low resource, namely:

- (1) English
- (2) Yiddish
- (3) Hebrew
- (4) Russian
- (5) German
- (6) French
- (7) Bengali

In our preliminary experiments we find that *xlm-roberta-large* consistently outperforms *bert-base-multilingual-cased* at preserving the persistent homology across languages, and across contexts within a fixed language. For a fair comparison, context size in terms of tokens was to roughly 150 ± 30 tokens. The effects of drastically different context sizes is not studied here. For *bert-base-multilingual-cased*, the persistence diagrams tended to have significantly larger distances for Yiddish, Hebrew, and Bengali, compared to the other four languages. For *xlm-roberta-large* the difference was less pronounced and seemed significant only for Yiddish (relative to the other six languages).

While this is only a preliminary set of experiments and examples, with a limited number of languages, the findings were very consistent and compelling.⁴ We also note another study of the same phenomena for Hebrew for the following models is also being conducted:

- (1) bert-base-multilingual-cased
- (2) heBERT
- (3) xlm-roberta-large
- (4) AlephBERT
- (5) dicta-il/alephbertgimmel-base

All of these can be found on Huggingface. In this case, we find that *xlm-roberta-large*, *AlephBERT*, and *dicta-il/alephbertgimmel-base* consistently outperform *bert-base-multilingual-cased* and *heBERT* at preserving the persistent homology across contexts, but that within the two groups we find no clear winner in our examples.

6. SOME HEURISTICS: COMPARING BARCODE DIAGRAMS

It appears, based on the limited experiments run so far, that the Wasserstein distance exhibits trends that the bottleneck distance does not always exhibit. Recall, given two persistence diagrams D_1 and D_2 , the **bottleneck distance** d_B is defined as:

$$d_B(D_1, D_2) = \inf_{\phi: D_1 \rightarrow D_2} \sup_{x \in D_1} \|x - \phi(x)\|_\infty,$$

where ϕ ranges over all bijections between the diagrams D_1 and D_2 , and $\|\cdot\|_\infty$ represents the L_∞ norm. Essentially, the bottleneck distance measures the maximum difference between matched points in the two diagrams. It is notably robust to noise, as it focuses on the worst-case discrepancies between the diagrams. However, it may fail to capture finer distinctions between diagrams that exhibit numerous smaller differences.

²This model is available on HuggingFace at <https://huggingface.co/xlm-roberta-large>

³This model is available on HuggingFace at <https://huggingface.co/bert-base-multilingual-cased>

⁴Code in the form of some example notebooks can be found at https://github.com/Amelie-Schreiber/multilingual_persistent_homology

In contrast, the **Wasserstein distance** offers a more nuanced comparison between persistence diagrams. The p -th Wasserstein distance W_p between D_1 and D_2 is defined as:

$$W_p(D_1, D_2) = \left(\inf_{\gamma \in \Gamma(D_1, D_2)} \int_{\mathbb{R}^2 \times \mathbb{R}^2} \|x - y\|^p d\gamma(x, y) \right)^{1/p},$$

where $\Gamma(D_1, D_2)$ denotes the set of all couplings of D_1 and D_2 , and $\|\cdot\|$ represents the Euclidean distance. The p -Wasserstein distance measures the minimum "effort" required to transform one diagram into another, considering all possible bijections. It provides a holistic assessment of the diagrams, accounting for smaller but numerous variations that the bottleneck distance might overlook. Equivalently, we can define the p -Wasserstein distance $d_W^p(D_1, D_2)$ as

$$d_W^p(D_1, D_2) = \left(\inf_{\sigma: D_1 \rightarrow D_2} \sum_{x \in D_1} \|x - \sigma(x)\|_p^p \right)^{1/p}$$

where σ ranges over all bijections from D_1 to D_2 , and $\|\cdot\|_p$ denotes the p -norm. When $p = 1$, we get the 1-Wasserstein distance (also known as the Earth Mover's distance), which intuitively measures the amount of "work" needed to transform one diagram into another, where "work" is the "mass" of a feature times the distance it has to be moved. When $p = 2$, we get the 2-Wasserstein distance, which is more sensitive to outliers. The Wasserstein distance is less stable than the bottleneck distance, but it can capture more subtle differences between diagrams.

Importantly, the persistence diagram space equipped with the Wasserstein distance exhibits mathematical properties that make it amenable to statistical analyses. Specifically, the space is complete and separable under the Wasserstein distance, enabling the definition of statistical tools such as expectation, variance, and conditional probability in the Fréchet sense.

The selection between the Wasserstein and bottleneck distances is not a one-size-fits-all decision; rather, it depends on the data's characteristics and the specific research question:

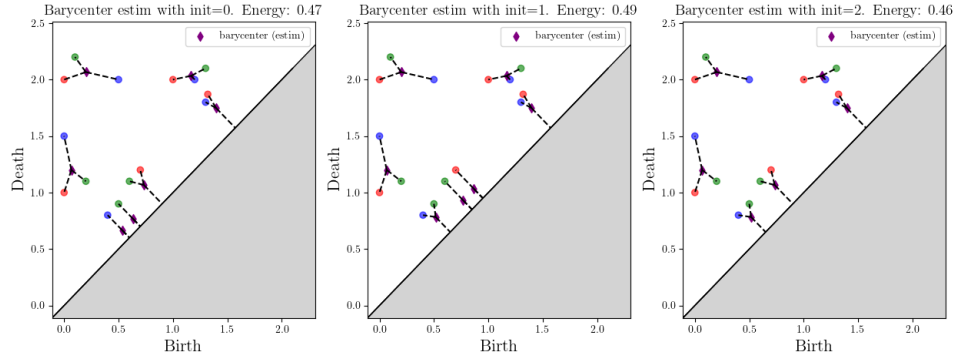
- **Bottleneck Distance:** Choose this metric when dealing with noisy data or when the interest is predominantly in the most significant topological features, as it is less sensitive to small, numerous variations.
- **Wasserstein Distance:** Opt for this metric when analyzing clean data, where capturing subtle differences across diagrams is essential. Moreover, if the aim is to perform statistical analysis on the diagrams, the Wasserstein distance is the preferred choice due to its favorable mathematical properties.

In essence, both the Wasserstein and bottleneck distances provide distinct yet complementary insights into the comparison of persistence diagrams. The choice between these two distances should be informed by the nature of the data and the research objectives at hand. Since we are using very high dimensional context vectors, and our tokens are a subset of collections of around 150 or so other tokens, it is not surprising that the Wasserstein distance proves more useful in capturing trends between languages, between models, and between individual attention heads.

7. FRÉCHET MEAN OF PERSISTENCE DIAGRAMS

In this section we discuss the possibility of averaging multiple barcodes or persistence diagrams for the context vectors associated to a collocation or multiword expression. We do this in order to find an average topological structure of a collocation or multiword expression, represented in many different contexts with each context containing roughly the same number of tokens. Finding such an average topological structure would then allow us to use its persistence diagram or barcode as a reference point in the space of all persistence diagrams when including persistent homology in the objective of a large language model during pretraining, effectively encouraging the model to learn

contextual representations of the collocation or multiword expression that have similar persistent topological features to this mean topological structure. To do this, we need the concept of Fréchet mean; since the space of persistence diagrams is not a linear space, we cannot simply take the arithmetic mean. [MMH, TMMH, LCO, DL] A generic example of the barycenter diagram using optimal transport and the topological data analysis library Gudhi can be seen below. We refer to the notebook for an example with *xlm-roberta-large*.⁵



Note, this does not restrict the individual context vectors for each token of the collocation or MWE, nor does it impose any direct restrictions on the other tokens' context vectors. It only imposes a restriction on the relative distances between the context vectors of the collocation or MWE, so that the persistent homology is preserved, relative to some average that can be computed for many different contexts. This will encourage the model to learn a topological representation of the context vectors that is more stable across different contexts, which as we have already seen, is a behavior that models better at many multilingual tasks exhibit.

Conjecturally, this may be important in multilingual tasks such as machine translation, as models that are known to perform better at preserving the persistent homology also perform better at translation tasks. This could provide a way of boosting performance of smaller models that have not learned topologically stable representations of collocations and multiword expressions. Note, this could be used to train or fine tune single attention heads that detect collocations, MWE, or idioms, and using Low Rank Adaptations we may be able to accomplish this even after the fine-tuning phase, especially if we find that some subset of attention heads already preserved the persistent homology of collocations, or MWEs better than others.

REFERENCES

- [DLYF] Yu Dai, Yuqiao Liu, Lei Yang, Yufan Fu, *An Idiom Reading Comprehension Model Based on Multi-Granularity Reasoning and Paraphrase Expansion*, Appl. Sci. 2023, 13, 5777. <https://doi.org/10.3390/app13095777>
 - [DL] Vincent Divol, Théo Lacombe *Understanding the Topology and the Geometry of the Space of Persistence Diagrams via Optimal Partial Transport*, <https://arxiv.org/abs/1901.03048>
 - [Hu et. al.] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, *LoRA: Low-Rank Adaptation of Large Language Models*, <https://arxiv.org/abs/2106.09685>
 - [LCO] Théo Lacombe, Marco Cuturi, Steve Oudot, *Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport*, <https://arxiv.org/abs/1805.08331>
 - [MMH] *Probability measures on the space of persistence diagrams*, Yuriy Mileyko, Sayan Mukherjee, John Harer, <https://math.hawaii.edu/yury/papers/probpers.pdf>
 - [TMMH] Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, John Harer, *Fréchet Means for Distributions of Persistence diagrams*, <https://arxiv.org/abs/1206.2790>
- Email address: amelie.schreiber.math@gmail.com

⁵See the following repo https://github.com/Amelie-Schreiber/multilingual_persistent_homology