Exploring Dowker Homology for Sentence Similarity

Marius Huber **Juri Opitz**
Department of Computational Linguistics
University of Zürich
{marius.huber,jurialexander.opitz}@uzh.ch

Abstract

Dowker homology is a topological tool that may be used to analyze the relative position of two point clouds living in a common space. We investigate whether Dowker homology captures sentence similarity information by treating the embeddings of the tokens that constitute a sentence pair as a pair of point clouds in the latent space of a transformer model, using both models that have and have not been fine-tuned for sentence similarity. We find that Dowker homology captures sentence similarity information, as measured by regressing Dowker homology features onto ground-truth similarity scores, and that it can be used for visual inspection of similarity data and models. In an attempt to make Dowker homology readily applicable, we derive from it single-number summaries that we expect to capture sentence similarity directly. These turn out to work reasonably well, but without outperforming standard sentence similarity measures based on established pooling methods.[1]

1 Introduction

Dowker homology (DH; Dowker, 1952; Chowdhury and Mémoli, 2018) is a tool from topological data analysis, an emerging data analysis framework drawing on ideas from topology, metric geometry, and related areas of mathematics. Among other things, DH may be used to capture the relative position of two point clouds living in a common space that is endowed with a distance function (typically $\mathbb{R}^n$ with Euclidean or cosine distance).

In this paper, we study a novel application of DH to NLP by applying it to Siamese sentence transformer models (Reimers and Gurevych, 2019). Specifically, we ask the following.

(1) Can DH distinguish similar and dissimilar sentence pairs?

(2) Can DH be used to extract sentence similarity from token embeddings?

On the positive side, we find that DH can elucidate sentence similarity data and models, both quantitatively and visually. Findings regarding the second question are mixed: we define single-number summaries extracted from DH that capture ground truth sentence similarity reasonably well, but not more so than established methods.

2 Background on Dowker Homology

This section provides a high-level review of DH; for details, we refer the reader to Appendix A.1. We limit ourselves to reviewing zero-dimensional DH, as that is the version of DH used in our experiments.

We consider a pair of point clouds X_1 and X_2 living in a common space that is endowed with a distance function d. We say that a point in X_1 or X_2 is *r-close* to another if the distance between the two points is at most r (as measured by d). Given a scale $r \geq 0$, one constructs a graph whose vertices are those points in X_1 that have a neighbor in X_2 that is r-close, and where two points in X_1 are connected by an edge whenever there exists a point in X_2 that is r-close to *both* of them. In particular, the graph is constructed on elements of X_1 only, while X_2 merely dictates the connectivity of the graph. To obtain DH of X_1 relative to X_2, one initially sets the scale to $r = 0$, and gradually increases it to theoretical infinity. Along the way, one keeps track of the number of connected components constituting the aforementioned graph: any such component "is born" at a certain scale, and "dies" at some later scale as it is absorbed by—that is, connected via an edge to—an existing component. This yields a sequence $(b_1, d_1), \ldots, (b_k, d_k)$, where each pair contains the scales at which the birth and death, respectively, of a component happen. This sequence is recorded in a *persistence*

[1] All code to run experiments is available at https://github.com/m-a-huber/dowker-for-sent-sim and archived at https://doi.org/10.5281/zenodo.21810253.

Proceedings of the Seventh Workshop on Insights from Negative Results in NLP (Insights 2026), pages 7–18
October 29, 2026 ©2026 Association for Computational Linguistics

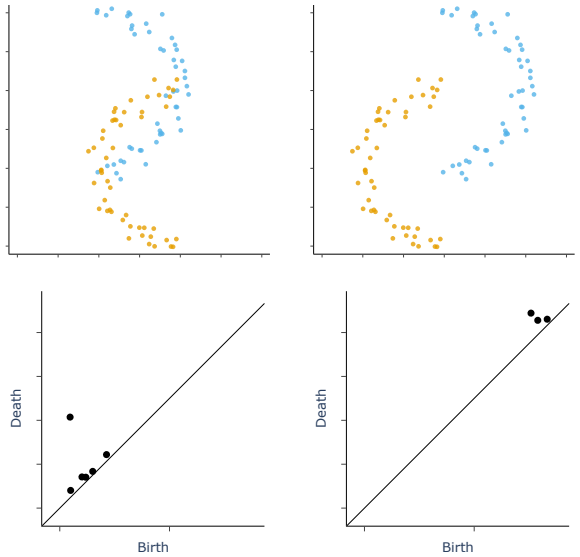


Figure 1: Two pairs of point clouds of varying degree of spatial separation (top) and their corresponding Dowker persistence diagrams (bottom). The more the point clouds are separated, the larger the birth values in the persistence diagrams.

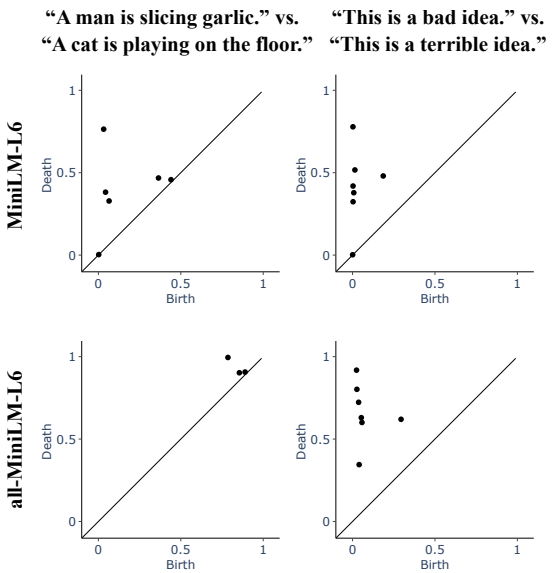


Figure 2: Dowker persistence diagrams capturing sentence similarity of two randomly selected sentence pairs from the STS-B dataset with low (left column) and high (right column) similarity, computed from embeddings produced using MiniLM-L6 (top row) and all-MiniLM-L6 (bottom row). Higher sentence similarity is reflected in smaller birth values in the diagrams, and the fact that MiniLM-L6 is not fine-tuned for sentence similarity is reflected in the top two diagrams being more similar to each other than the bottom two diagrams. Moreover, fine-tuning MiniLM-L6 for sentence similarity substantially alters the persistence diagram corresponding to the dissimilar sentence pair, while leaving the other diagram largely unaffected.

diagram, which is obtained by placing a point at coordinates (b_i, d_i) in $\mathbb{R}^2$ for all $i = 1, \ldots, k$.[2] A key property of DH is that it is unaffected when swapping the roles of X_1 and X_2 in the above process, making the resulting persistence diagram an "undirected" quantity (Dowker, 1952; Chowdhury and Mémoli, 2018).

If the point clouds X_1 and X_2 are spatially well separated, vertices and edges appear only at large scales. If X_1 and X_2 nearly coincide, those appear at small scales already; indeed, if $X_1 = X_2$, all birth values equal zero. The spatial proximity (resp. separation) of X_1 and X_2 is thus reflected in the points of the Dowker persistence diagram having small (resp. large) birth values; see Figure 1.

3 Dowker Homology for Similarity

To apply DH to sentence similarity, we treat the token embeddings stemming from a pair of sentences as a pair of point clouds in the latent space of a transformer model, and compute the corresponding Dowker persistence diagram. If the two sentences are similar (resp. dissimilar), we expect the corresponding point clouds to be colocalized (resp. separated) and, as discussed in Section 2, we expect the resulting Dowker persistence diagram to have predominantly small (resp. large) birth values.

We point out that our method is similar in spirit to those used in Barannikov et al. (2022) and Michel et al. (2017). Indeed, both methods also draw on ideas from topological data analysis to assess the (dis-)similarity of two point clouds. The former method, however, requires the two point clouds to be of equal sizes, while the latter takes into account only the topological features of the two point clouds individually and does not capture, for instance, their spatial separation.

Example. Figure 2 illustrates that our expectation holds for two sentence pairs randomly selected from the STS Benchmark (STS-B; Cer et al., 2017) when using the fine-tuned all-MiniLM-L6 model, but not when using the non-fine-tuned MiniLM-L6 model. Fine-tuning a transformer model for sentence similarity seems to strongly affect how dissimilar sentences are represented (as seen through the lens of DH), but less so for similar sentences,

[2] A persistence diagram is usually endowed with the diagonal line where birth values equal death values. The vertical distance of the i-th point to the diagonal equals $d_i - b_i$, and can thus be thought of as the "lifetime" of the corresponding connected component, $i = 1, \ldots, k$.

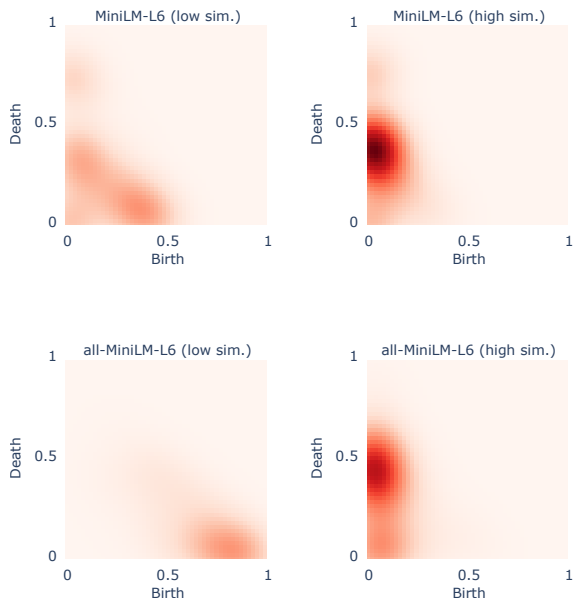


Figure 3: Mean persistence images capturing sentence similarity of all sentence pairs from the STS-B dataset with low (left column) and high (right column) similarity, computed from embeddings produced using MiniLM-L6 (top row) and all-MiniLM-L6 (bottom row). These persistence images can be thought of as "heat maps" that capture the distribution of points across multiple persistence diagrams (see Appendix A.2 for details). The images indicate that fine-tuning MiniLM-L6 for sentence similarity leads to larger birth values in DH of dissimilar sentence pairs, thus generalizing the findings from Figure 2 across the entire STS-B dataset. Here, the sentence pairs with low (resp. high) similarity are those whose ground truth similarity in STS-B differs by at most 0.1 from the minimum (resp. maximum) possible similarity value. For the analogous plots for the other models used, see Appendix A.5.1.

where the persistence diagrams stemming from the non-fine-tuned and the fine-tuned model are basically indistinguishable. Indeed, the finding of Figure 2 generalizes across all similar and dissimilar pairs from the STS-B dataset, as illustrated in Figure 3.

Experiment outline. Motivated by this example, we conduct the following two experiments for each layer of every model from a diverse set of transformer models fine-tuned for sentence similarity as well as their underlying, non-fine-tuned base models (see Table A.1 in Appendix A.3 for a list of models used). We use sentence pairs from the STS-B dataset, which contains sentence pairs annotated by human raters with semantic similarity scores ranging from 0 to 5.

1. **Layer regression experiment:** We investigate to what extent standard token-level measures as well as Dowker persistence diagrams contain information on sentence similarity. We do so by fitting a linear regression model that predicts ground truth sentence similarities from these measures.

2. **Layer correlation experiment:** In order to circumvent the need for a regression trained on Dowker persistence diagrams, and to provide measures of sentence similarity that are more readily applicable than the raw persistence diagrams, we define single-number summaries extracted from DH. These summaries are defined so that we expect them to capture sentence similarity directly, and we assess how well they correlate with ground truth sentence similarities.

We now describe the sentence similarity methods used in our experiments.

3.1 Baseline Similarity Methods

We use BOS-, EOS-, max- and mean-pooling as single-vector sentence representations. The similarity of a sentence pair is then computed as the cosine similarity between the pooled representations of the sentences.

3.2 Dowker Similarity

We define four versions of *Dowker similarity* (DS), one for each of four different kinds of aggregation used along the way. To compute DS for a sentence pair (s_1, s_2), we consider the token embeddings of s_1 and s_2 as point clouds X_1 and X_2, respectively, and compute DH of X_1 relative to X_2, as explained in Section 2. We extract single-number summaries from the resulting persistence diagram by computing the minimum, maximum and average of all birth values of points in the diagram, as well as the weighted average birth value, where birth values are weighted by the lifetime of the corresponding point. We regard each of these as a notion of distance between X_1 and X_2, since larger birth values indicate a larger degree of spatial separation of X_1 and X_2 (see Section 2), and we denote these distances by $d_{\mathrm{Dow}}^{\mathrm{min}}$, $d_{\mathrm{Dow}}^{\mathrm{max}}$, $d_{\mathrm{Dow}}^{\mathrm{mean}}$ and $d_{\mathrm{Dow}}^{\mathrm{wght}}$, respectively. Finally, we define the DSs by setting $s_{\mathrm{Dow}}^{\bullet}(s_1, s_2) = e^{-d_{\mathrm{Dow}}^{\bullet}(X_1, X_2)}$ for $\bullet \in \{\mathrm{min}, \mathrm{max}, \mathrm{mean}, \mathrm{wght}\}$. All versions of DS thus attain values between 0 and 1. Moreover, if $s_1 = s_2$, then $X_1 = X_2$, and all birth values in DH equal zero, so that all four versions of Dow-

ker distance vanish, and, correspondingly, all four versions of DS yield a similarity value of 1 for identical sentences.

4 Experiments and Results

We now provide detailed descriptions of the two experiments conducted, and then briefly discuss some limitations and design choices.

4.1 Layer Regression Experiment

Setup. We sample 512 and 256 sentence pairs from the train and test splits, respectively, of STS-B. As baseline features, we use the vectors $(u_1, u_2, |u_1 - u_2|)$, where u_1 and u_2 are the pooled representations of a sentence pair (s_1, s_2), obtained via the four pooling methods described in Section 3.1 (this is motivated by Reimers and Gurevych (2019, Section 6); here, $|\cdot|$ denotes the entry-wise absolute value). For DH, we compute the persistence diagram of each sentence pair from the corresponding token embeddings (as described in Section 3.2) using cosine distance, and vectorize it into a persistence image of resolution 25×25, resulting in a vector of length 625 (see Appendix A.2 for details on persistence images). We then train a ridge regression on these features and evaluate it using Spearman's rank correlation between the predicted and the ground-truth similarities; the choice of ridge regression is motivated by the fact that persistence images produce high-dimensional, sparse vectors with correlated features, making ordinary least squares estimates potentially unstable (see Appendix A.4 for selection of hyperparameters). This procedure is run for each hidden layer of each model.

Results. We find that across all models (fine-tuned or not), the regression trained on the Dowker persistence diagrams never outperforms the one trained on the best performing baseline similarity methods. The regression trained on the Dowker persistence diagrams always wins at least third place (as measured by Spearman correlation averaged across all layers of a model), consistently beating those trained on BOS- and EOS-pooling, but coming short of outperforming those trained on max- and mean-pooling. Interestingly, max-pooling yields the best features for $9/10$ models despite the fact that the sentence transformers used were fine-tuned using mean-pooling. Finally, we observe that in the fine-tuned models—in contrast to the non-fine tuned ones—all similarity methods tend to yield higher Spearman correlations on later layers, which is somewhat intuitive as the final layer is where fine-tuning is directly applied. As an example, Figure 4 shows the Spearman correlations of all methods at each layer of MiniLM-L6 (not fine-tuned) and all-MiniLM-L6 (fine-tuned); the former is one of the two cases in which the features derived from DH rank second, outperforming those derived from mean-pooling. Appendix A.5.2 shows analogous plots for the other model pairs.

4.2 Layer Correlation Experiment

Setup. We sample 256 sentence pairs from the test split of STS-B, and compute similarities for each hidden layer of each model, using the four baseline similarity methods and the four versions of DS from Sections 3.1 and 3.2, respectively. For each method, model and layer, we evaluate the performance via Spearman correlation between the predicted and the ground-truth similarities.

Results. As before, we assess the goodness of the baseline methods and the four versions of DS by the Spearman correlations averaged across all layers of a model. We find that (at least one version of) DS comes in third behind the max- and mean-pooling methods for $5/10$ models, and wins first and second place for $2/10$ models each. Again, we find that max-pooling outperforms mean-pooling in terms of average Spearman correlation. Mean-pooling, however, unsurprisingly yields the highest maximal Spearman correlation for the sentence transformer models, and this is always the case at the final model layer, sometimes after a sheer increase of performance in the last few layers. Finally, we find that $s_{\mathrm{Dow}}^{\mathrm{mean}}$ outperforms the other versions of DS for $9/10$ models. As an example, Figure 5 shows the Spearman correlations of all methods at each layer of MiniLM-L6 and all-MiniLM-L6; the former is one of two cases where $s_{\mathrm{Dow}}^{\mathrm{mean}}$ outperforms all other methods (the other such instance being MiniLM-L12). We refer to Appendix A.5.3 for plots of the remaining results.

4.3 Experiment Limitations

Dataset and Sample Sizes. In both experiments, we confined ourselves to working with just the STS-B dataset and the relatively modest respective sample sizes due to computational constraints.

Computation of DH Feature Vectors. While DH may be computed using any metric, we restricted ourselves to using cosine distance in our

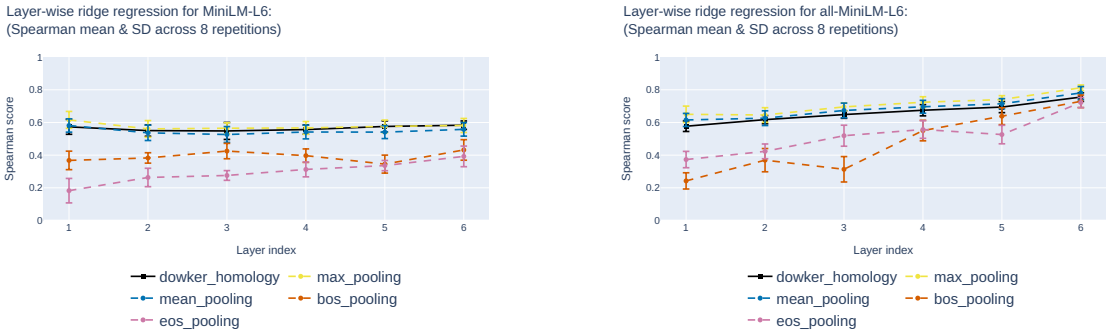


Figure 4: Two results of the layer regression experiment. Bars indicate standard deviation over eight runs.

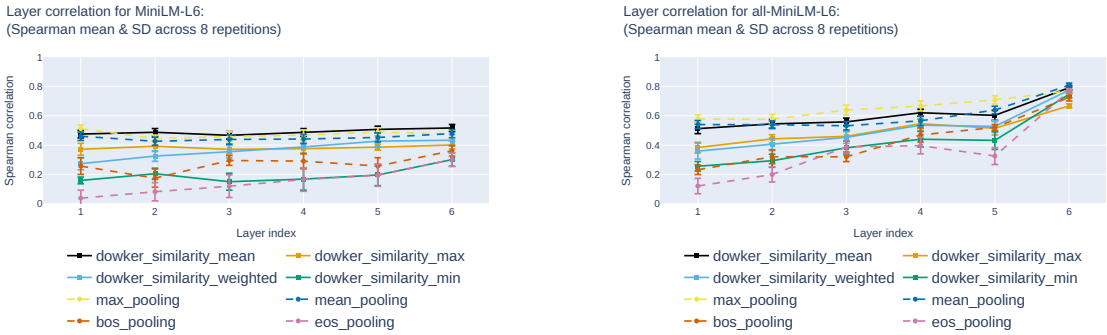


Figure 5: Two results of the layer correlation experiment. Versions of Dowker similarity are indicated by the suffixes.

experiments. We experimented using Euclidean distance, but cosine distance consistently yielded better results. For the layer regression experiment, the resolution of 25×25 used for vectorization of persistence diagrams was chosen as an empirical compromise between preserving performance and limiting the dimensionality of the resulting feature vectors. Finally, we restricted ourselves to working with zero-dimensional DH because we suspected higher-dimensional DH to capture information that goes beyond spatial separation of point clouds. Indeed, preliminary experiments showed that the point clouds consisting of token embeddings rarely exhibited any higher-dimensional DH at all, and including higher-dimensional DH as features in the layer regression experiment had virtually no effect on the results.

Evaluation metric. In both experiments, we confined ourselves to evaluating the agreement of DSs with ground truth similarities via Spearman correlation since DSs are obtained by applying the monotonic transformation $d \mapsto e^{-d}$ to the uncalibrated distance measures $d_{\mathrm{Dow}}^{\mathrm{min}}$, $d_{\mathrm{Dow}}^{\mathrm{max}}$, $d_{\mathrm{Dow}}^{\mathrm{mean}}$ and $d_{\mathrm{Dow}}^{\mathrm{wght}}$. We hence do not expect any linear relationship between DSs and ground truth similarities as captured, for instance, by Pearson correlation.

5 Discussion

Our experiments show that Dowker persistence diagrams reflect sentence similarity (cf. Question (1)) and, moreover, that they can provide visual insights on NLP models, such as the effect of fine-tuning on the representation of similar and dissimilar sentence pairs (cf. the example in Section 3). Moreover, while all versions of DS capture sentence similarity to some extent, the version that averages birth values is by far the best-performing one, as witnessed by the fact that it correlates with ground truth similarities essentially as well as a trained linear regression. Even this version of DS, however, does not consistently outperform established methods, answering Question (2) with a "yes, but". In summary, our findings provide insights on the usefulness and interpretability of DH, and justify further research on connections between topology and sentence similarity, such as exploration of further single-number summaries extracted from DH.

References

Henry Adams, Tegan Emerson, Michael Kirby, Rachel Neville, Chris Peterson, Patrick Shipman, Sofya Chepushtanova, Eric Hanson, Francis Motta, and Lori Ziegelmeier. 2017. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35.

Serguei Barannikov, Ilya Trofimov, Nikita Balabin, and Evgeny Burnaev. 2022. Representation topology divergence: A method for comparing neural network representations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 1607–1626. PMLR.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Samir Chowdhury and Facundo Mémoli. 2018. A functorial Dowker theorem and persistent homology of asymmetric networks. *J. Appl. Comput. Topol.*, 2(1-2):115–175.

Pawel Dlotko. 2026. Persistence representations. In *GUDHI User and Reference Manual*, 3.12.0 edition. GUDHI Editorial Board.

C. H. Dowker. 1952. Homology groups of relations. *Ann. of Math. (2)*, 56:84–95.

Marius Huber, David Robert Reich, and Lena Ann Jäger. 2026. Fixation sequences as time series: A topological approach to dyslexia detection. In *Proceedings of the 2026 Symposium on Eye Tracking Research and Applications*, ETRA '26, New York, NY, USA. Association for Computing Machinery.

Marius Huber and Patrick Schnider. 2025. Flagifying the dowker complex. *Preprint*, arXiv:2508.08025. ArXiv:2508.08025 [math.AT], https://arxiv.org/abs/2508.08025.

Paul Michel, Abhilasha Ravichander, and Shruti Rijhwani. 2017. Does the geometry of word embeddings help document classification? A case study on persistent homology-based representations. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 235–240, Vancouver, Canada. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

A Appendix

A.1 Dowker Homology

DH, introduced by Dowker (1952) and promoted to the setting of persistent homology by Chowdhury and Mémoli (2018), is a tool from topological data analysis that captures the topology of two sets X and Y and a relation $R \subseteq X \times Y$ between them. Here, we review DH as used in the main text of the article. We point out that the implementation of DH used in our experiments is that of Dowker–Rips homology provided by Huber and Schnider (2025). While Dowker–Rips homology is, in general, merely an approximate, computationally cheaper variant of DH, we use that implementation since all our experiments rely only on zero-dimensional DH, in which case Dowker and Dowker–Rips homology coincide. Correspondingly, the following exposition is restricted to zero-dimensional DH.

In the case where X and Y are subsets of some metric space (Z, d), and given some scale $r \geq 0$, one may define a relation R_r by declaring that $(x, y) \in R_r$ iff $d(x, y) \leq r$, for $x \in X$ and $y \in Y$. The relation R_r may be translated into the graph $\Gamma_r = (V_r, E_r)$, defined as having vertex set

$$V_r = \{x \in X \mid \exists y \in Y \colon d(x, y) \leq r\} \subseteq X,$$

and where two vertices $x_1, x_2 \in X$ are connected via an edge whenever there exists $y \in Y$ such that $d(x_1, y) \leq r$ and $d(x_2, y) \leq r$.

As is the case for any graph, the graph Γ_r consists of some—possibly zero—connected components. Indeed, if $r = 0$, $\Gamma_r = \varnothing$ (provided that X and Y have no common elements). As one increases the scale r, vertices and edges appear in Γ_r. Correspondingly, the number of connected components of Γ_r can increase or decrease as r increases: a new vertex may appear when r becomes large enough for that vertex to "see" an element of Y, or two existing components of Γ_r may merge into one when r becomes large enough for two vertices initially belonging to different connected components to be connected via an edge. In other

words, increasing the scale r from zero to (theoretical) infinity yields a sequence "births" and "deaths" of connected components. DH is the information contained in this process, and may be recorded by a sequence $(b_1, d_1), \ldots, (b_k, d_k)$ of pairs of non-negative real numbers, where the i-th pair contains the value of the scale r at which the "birth" (b_i) and "death" (d_i), respectively, of the i-th connected component occur.

This sequence is usually recorded in a *persistence diagram*, which, by definition, is the subset of $\mathbb{R}^2$ consisting of the pairs $(b_1, d_1), \ldots, (b_k, d_k)$ (see Figure 1 in the main text for examples of persistence diagrams). Persistence diagrams are usually endowed with the diagonal line characterized by those points where birth values equal death values; the vertical distance of the point (b_i, d_i) in the persistence diagram indicates the lifetime (that is, $d_i - b_i$) of the i-th connected component. As such, a persistence diagram provides an at-a-glance summary of the evolution of the graph Γ_r as r increases from zero to infinity, and—as explained in Section 2—captures the spatial separation of X and Y as measured by the metric d defined on the metric space (Z, d). We close this section by remarking that any Dowker persistence diagram contains precisely one point whose lifetime is infinite, that is, which dies at only scale $r = \infty$. Since this infinite value typically breaks many formulas and pipelines, one usually either discards this point from the diagram, or replaces the scale at which it dies by some finite number that is chosen according to some heuristic. In our experiments, we experimented with both options, and found that simply discarding that point yielded better and more stable results.

A.2 Vectorizing Persistence Diagrams

To make persistence diagrams usable in machine learning pipelines, one must transform them into vectors of a fixed length. In our experiments, we use a method known as *persistence image* (Adams et al., 2017). A persistence image is obtained from a persistence diagram by first applying a shearing that maps the diagonal of the persistence diagram to a horizontal line. The resulting diagram is then converted into a "heat map" by placing a bivariate Gaussian distribution with a given standard deviation (also referred to as *bandwidth* in the present context) at each point of the diagram. This yields a single-channel image commonly referred to as persistence surface; see Figure A.1 for an illustration of the process of obtaining a persistence surface from a persistence diagram. Finally, the persistence image is obtained from the persistence surface by discretizing it into a pixel grid of a chosen *resolution*. The bandwidth and resolution used in the process are treated as hyperparameters in applications.

A.3 Models used

See Table A.1 for a list of models used in our experiments.

A.4 Hyperparameter Selection

To train the ridge regressions used in the layer regression experiment described in Section 4.1, we used the implementation provided by scikit-learn (Pedregosa et al., 2011). In doing so, we used the SVD-solver, tuned the regularization parameter α via grid search over the values $10^{-3}, 10^{-2}, \ldots, 10^{2}$, and left other parameters at their default values. Persistence diagrams in these experiments were transformed into persistence images using the GUDHI library (Dlotko, 2026). We used default values for persistence image creation, apart from the bandwidth, which we tuned via grid search over the values $10^{-3}, 10^{-2}, \ldots, 10^{2}$.

A.5 Results

Here we provide plots containing the results from the two experiments conducted, analogous to the plots in the main text.

A.5.1 Mean Persistence Images for Dissimilar and Similar Sentence Pairs

See Figure 3 in the main text and Figures A.2a to A.2d.

A.5.2 Layer Regression Experiment

See Figure 4 in the main text and Figures A.3 to A.6.

A.5.3 Layer Correlation Experiment

See Figure 5 in the main text and Figures A.7 to A.10.

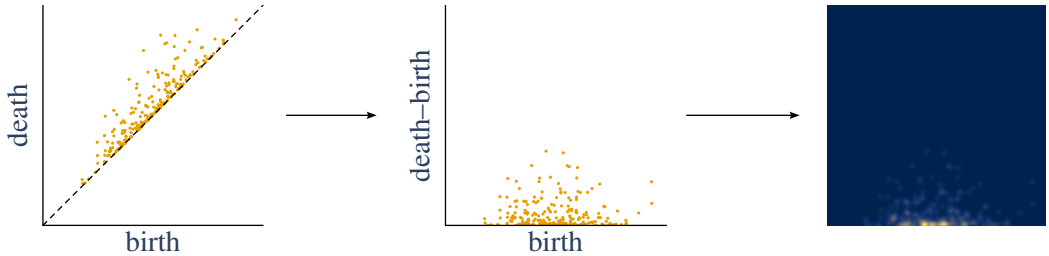
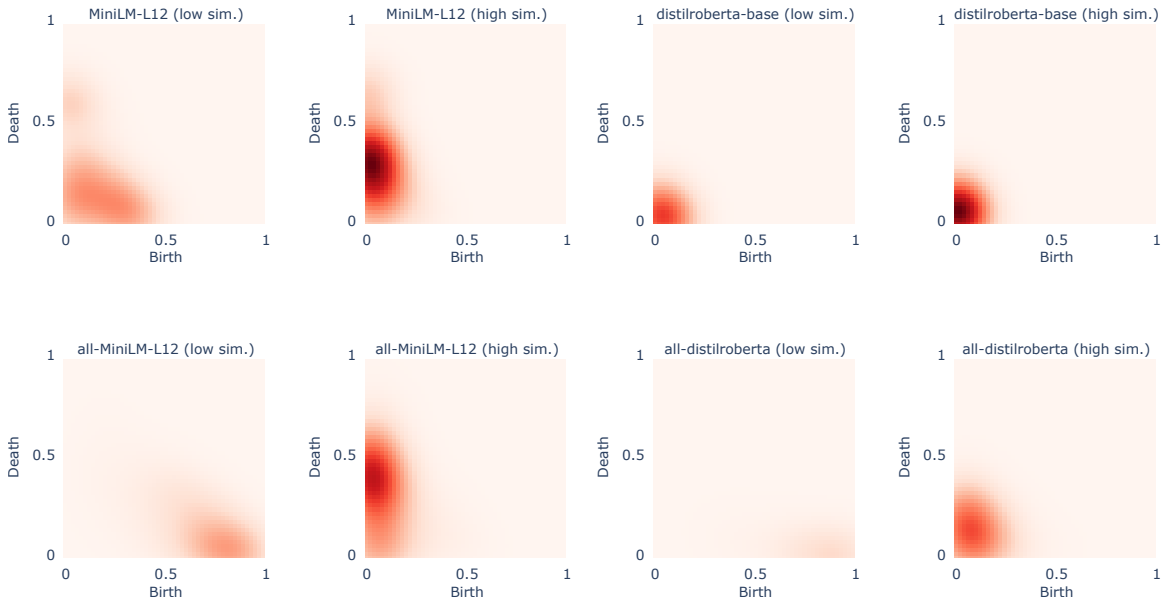


Figure A.1: Schematic showing the transformation of a persistence diagram in (birth, death)-coordinates (left) to (birth, death-birth)-coordinates (middle) by means of a shearing map, and the resulting persistence surface (right). Figure adapted from Huber et al. (2026).

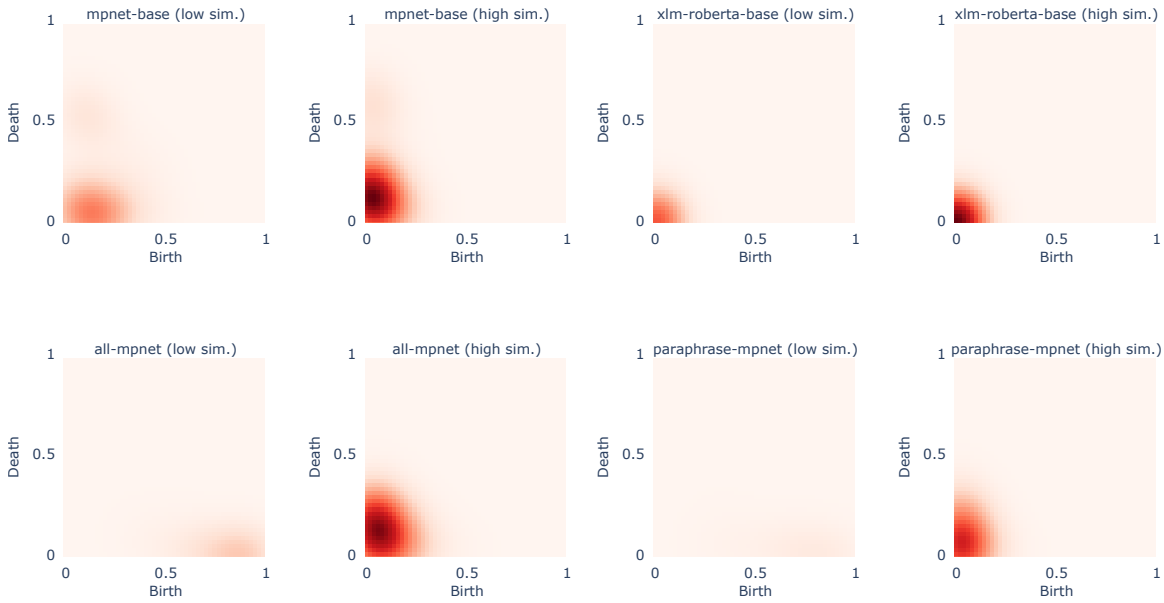
Base model	**Sentence transformer**
MiniLM-L6-H384-uncased	all-MiniLM-L6-v2
MiniLM-L12-H384-uncased	all-MiniLM-L12-v2
distilroberta-base	all-distilroberta-v1
mpnet-base	all-mpnet-base-v2
xlm-roberta-base	paraphrase-multilingual-mpnet-base-v2

Table A.1: Sentence transformers and respective base models used in our experiments.



(a) Mean persistence images for the models MiniLM-L12-H384-uncased and all-MiniLM-L12-v2.

(b) Mean persistence images for the models distilroberta-base and all-distilroberta-v1.



(c) Mean persistence images for the models mpnet-base and all-mpnet-base-v2.

(d) Mean persistence images for the models xlm-roberta-base and paraphrase-multilingual-mpnet-base-v2.

Figure A.2: Mean persistence images for the model pairs from Table A.1. See Figure 3 in the main text for a description of what these images capture and how they are obtained.

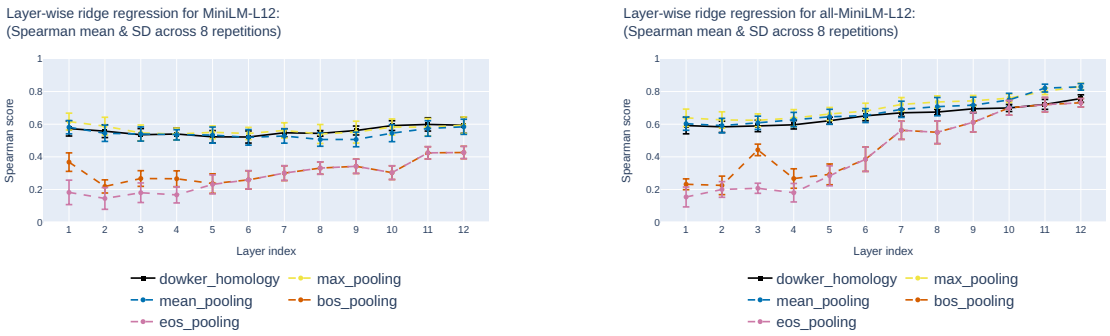


Figure A.3: Results of the layer regression experiment for MiniLM-L12-H384-uncased (left) and all-MiniLM-L12-v2 (right).

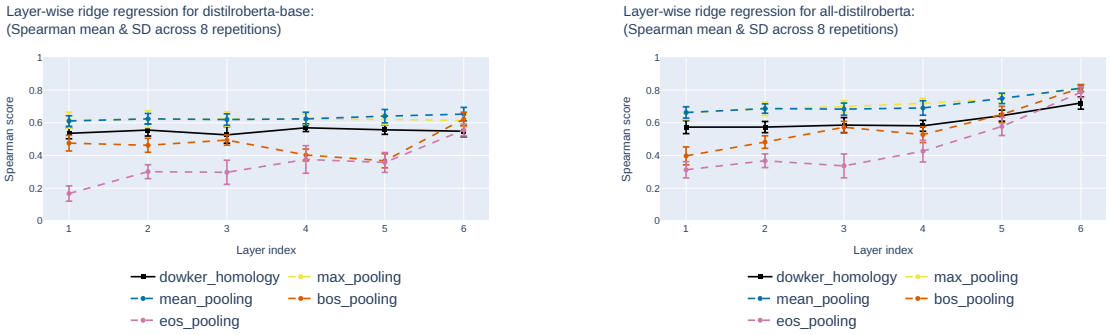


Figure A.4: Results of the layer regression experiment for distilroberta-base (left) and all-distilroberta-v1 (right).

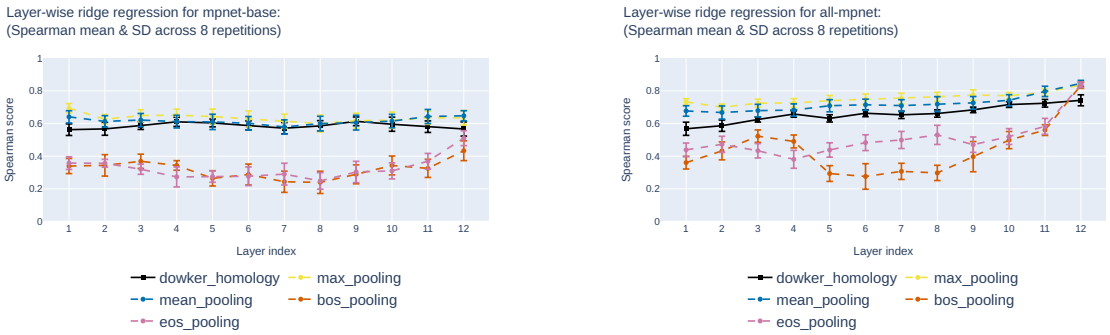


Figure A.5: Results of the layer regression experiment for mpnet-base (left) and all-mpnet-base-v2 (right).

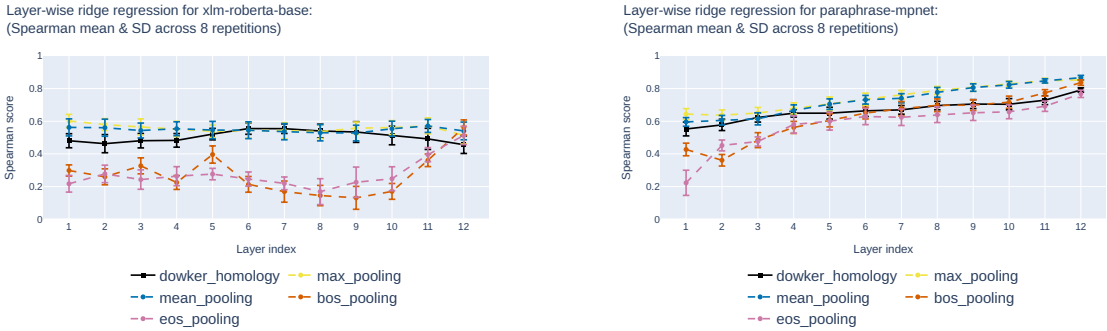


Figure A.6: Results of the layer regression experiment for xlm-roberta-base (left) and paraphrase-multilingual-mpnet-base-v2 (right).

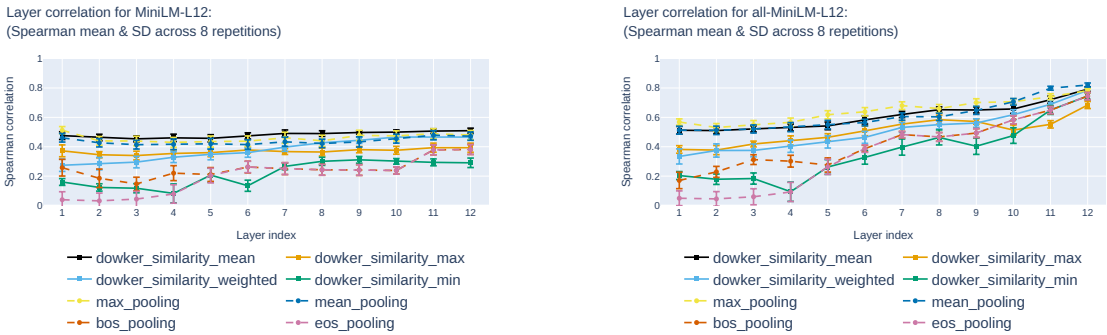


Figure A.7: Results of the layer correlation experiment for MiniLM-L12-H384-uncased (left) and all-MiniLM-L12-v2 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

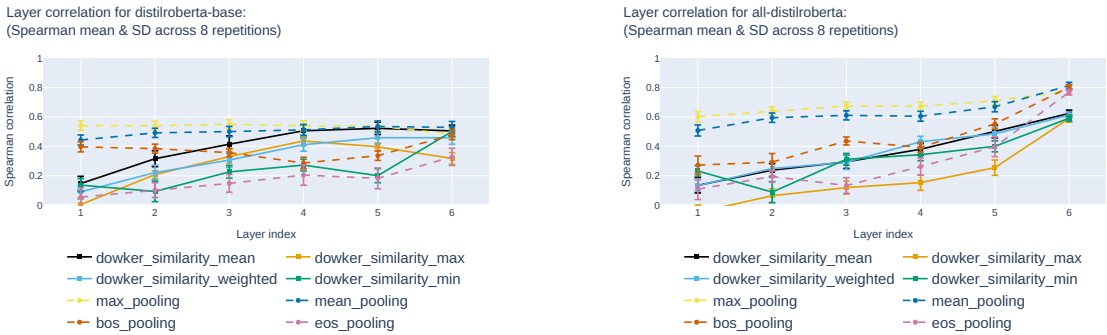


Figure A.8: Results of the layer correlation experiment for distilroberta-base (left) and all-distilroberta-v1 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

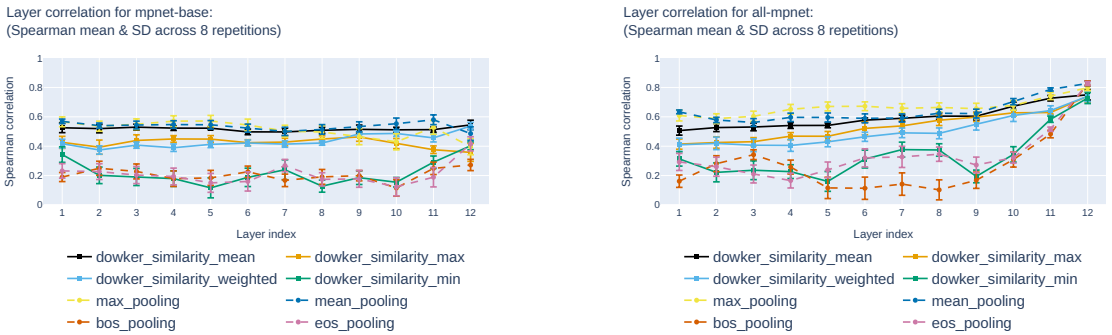


Figure A.9: Results of the layer correlation experiment for mpnet-base (left) and all-mpnet-base-v2 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.

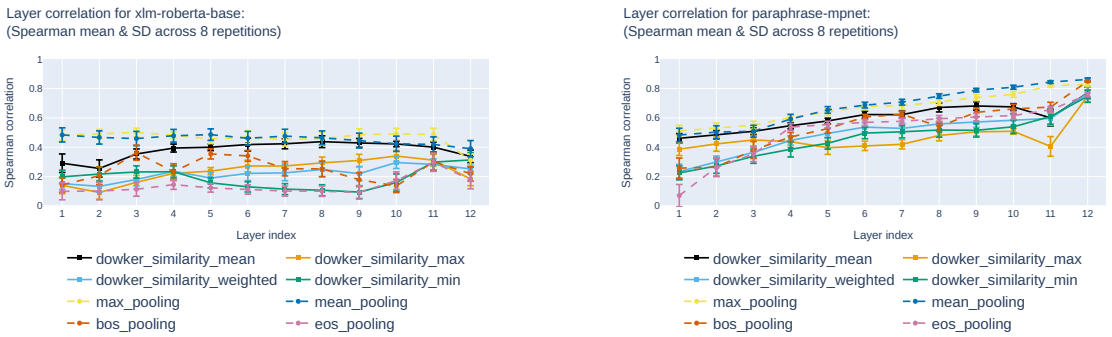


Figure A.10: Results of the layer correlation experiment for xlm-roberta-base (left) and paraphrase-multilingual-mpnet-base-v2 (right). Method names in the legend are sorted according to decreasing Spearman correlation averaged across model layers.