

# Local Topology Measures of Contextual Language Model Latent Spaces With Applications to Dialogue Term Extraction

Benjamin Matthias Ruppik, Michael Heck, Carel van Niekerk, Renato Vukovic,  
Hsien-Chin Lin, Shutong Feng, Marcus Zibrowius, Milica Gašić

Heinrich Heine University Düsseldorf, Germany  
{ruppik, heckmi, niekerk, renato.vukovic,  
linh, shutong.feng, marcus.zibrowius, gasic}@hhu.de

## Abstract

A common approach for sequence tagging tasks based on contextual word representations is to train a machine learning classifier directly on these embedding vectors. This approach has two shortcomings. First, such methods consider single input sequences in isolation and are unable to put an individual embedding vector in relation to vectors outside the current local context of use. Second, the high performance of these models relies on fine-tuning the embedding model in conjunction with the classifier, which may not always be feasible due to the size or inaccessibility of the underlying feature-generation model.

It is thus desirable, given a collection of embedding vectors of a corpus, i.e. a datastore, to find features of each vector that describe its relation to other, similar vectors in the datastore. With this in mind, we introduce complexity measures of the local topology of the latent space of a contextual language model with respect to a given datastore.

The effectiveness of our features is demonstrated through their application to dialogue term extraction. Our work continues a line of research that explores the manifold hypothesis for word embeddings, demonstrating that local structure in the space carved out by word embeddings can be exploited to infer semantic properties.

## 1 Introduction

The prevailing approach to sequence tagging tasks such as named entity recognition or dialogue term extraction involves a two-step process: start with a general contextual vector representation for text sequences, for instance the embedding vectors created by a pretrained language model, then train a separate tagging model on top of the vector representations (Lample et al., 2016; Ramshaw and Marcus, 1995). Optionally, assuming differentiability of the model and target function, one can fine-tune

the representation model such that its embeddings are more suitable for the tagging task (Panchendrarajan and Amaesan, 2018). While highly effective, the representations may be expensive to compute, and fine-tuning a language model is not always feasible, for instance if the underlying model is hidden behind an application programming interface (API). Thus, it is desirable to develop tagging methods which achieve the best performance on the given representations. In fact, the performance of prompting-based approaches with large language models (LLMs) on named entity recognition tasks has lagged behind that of supervised sequence tagging approaches (Wang et al., 2023). Additionally, this leads to problems such as hallucinations and potential dataset contamination, which prevent a fair evaluation.

A more fundamental limitation of the prevailing paradigm is that the relation of a single input sequence to other sequences in the dataset cannot be taken into account. Both the representation module and the tagging module commonly have a limited maximum context length. They cannot process the entire dataset at once, but rather need to be provided with single sentences or paragraphs at a time. The limited context can lead to suboptimal performance (Amalvy et al., 2023). For example, consider named entity recognition for an isolated sentence such as *Prince was prominently featured at the event*. The word *Prince* is ambiguous. In a corpus containing news articles, *Prince* or *Prince Harry* likely appear in many articles related to the British royal family. In a different corpus, the term *Curry Prince* might appear frequently in the context of restaurant reviews. So only with regard to the entire corpus under consideration, an informed choice on how to tag *Prince* in the example sentence could be made.

In this work, we show that the relation between the representation of a single token and its containing corpus can be captured by studying the *latent*

*space* – the collection of the language model’s hidden states – surrounding the corresponding embedding vector. The geometry of these hidden states is known to capture both syntactic and semantic properties of the underlying text. For instance, [Cohen et al. \(2019\)](#) find that distances between the contextual vectors of bidirectional encoder representations from transformers (BERT) ([Devlin et al., 2019](#)) correspond to parse tree embeddings based on the grammatical structure of the input phrases. Here, we study neighborhoods of embedding vectors from a *topological* viewpoint, and introduce descriptors of the shapes of these neighborhoods that are stable under symmetries such as permutations, translations, and rotations. In particular, we define descriptors based on *persistent homology*, a well-established tool of topological data analysis ([Carlsson and Vejdemo-Johansson, 2021](#)).

## 1.1 Contribution

Consider the latent space of a language model in the neighborhood of a given contextual embedding vector. For instance, the neighborhood of an embedding of the word *cheap* in the context *I am looking for options for cheap dinner* contains other occurrences of the word *cheap* in different contexts, but also different words expressing a similar meaning (*inexpensive, good-value*) and words connected to the center word, such as *restaurant*. In this work, we show that:

- (a) this neighborhood contains information that is not present in the language model next-token prediction distribution, and that cannot be ‘distilled’ into the language model via naive fine-tuning,
- (b) this additional information can be used to improve the performance of sequence tagging tasks, and
- (c) this information can be efficiently summarized using low-dimensional topological feature descriptors.

Our topological descriptors are codensity at multiple scales ([Carlsson et al., 2008](#); [Carlsson, 2014](#)), topological singularity measures based on Wasserstein norms ([Cohen-Steiner et al., 2010](#)), and vectorized persistence modules. Towards (a), we show that several of our one-dimensional numerical measures show minimal correlation with language model perplexity, indicating that they contain independent information. Towards (b) and

(c), we empirically demonstrate improvements on the natural language processing task of variants of term extraction. In each case, we build the latent space through a masked language model from a dialogue corpus. As a baseline, we employ a tagging model trained directly on the original language model vectors, and compare with models that take as input a combination of the language model vectors with our topological descriptors of the neighborhood within the latent space of a contextual language model. Furthermore, we compare with models trained on features from [Vukovic et al. \(2022\)](#), which are based on neighborhoods in a *static* word embedding space. We show that utilizing the *contextually* augmented vectors yields statistically significant improvements.

Observation (a) is not completely new. For example, it is present in the idea of  $k$ -nearest neighbor language models ([Khandelwal et al., 2020](#); [Xu et al., 2023](#)), where the current hidden state is augmented by the nearest neighbors from a datastore. Our low-dimensional descriptors, on the other hand, have not been deployed before, and our experiments for (b) provide the first application of contextual topological features to token level sequence tagging tasks. Note with reference to point (c) that summarizing a collection of vectors in a permutation-invariant way is a challenging problem in representation learning ([Zaheer et al., 2017](#)), which we tackle in this work via tools from persistent homology.

Our work is complementary to other recent applications of topological methods to the study of contextual embedding spaces. [Tulchinskii et al. \(2023\)](#) demonstrate that the topology of a point cloud derived from a text paragraph can be utilized in a sequence classification task, namely to differentiate human-written from artificially generated paragraphs. Their approach takes into account solely the given paragraph’s embedding vectors, and does not explore how these reside within the larger latent space. Another approach involves constructing filtered graphs from the attention scores in a transformer model, followed by sequence-level classification based on persistent homology ([Kushnareva et al., 2021](#); [Perez and Reinauer, 2022](#)). However, this approach only applies to supervised sequence classification tasks, and does not yield local features required for tagging. In a more qualitative direction, [Valeriani et al. \(2023\)](#) investigate the intrinsic dimension of the latent space through the different layers of a transformer, and [Ethayarajh](#)

(2019) and Cai et al. (2021) identify isolated clusters and low dimensional manifolds in the latent spaces of various language models. However, they do not apply their quantitative local analysis to a practical task.

## 2 Background and Methods

### 2.1 Latent Spaces of Contextual Language Models

We consider the encoder part of a contextual language model, which can be thought of as a map

$$e: (\mathbb{R}^d)^{\times N} \rightarrow (\mathbb{R}^h)^{\times N}.$$

Here,  $d$  is the dimension of the input layer,  $h$  is the hidden dimension ( $h \ll d$ ), and  $N$  is the maximum sequence length, after which sequences will be truncated. This maximum length is usually fixed and finite. The input of the encoder is a sequence of vectors  $\mathbf{X} \in (\mathbb{R}^d)^{\times N}$  representing a tokenized context. *Tokenization* describes the process in which an input string is decomposed into a sequence of vectors. In our setting, tokenization can be thought of as a lookup layer converting short text segments to vectors (together with position information). Typically, longer words are split into several token vectors in this process.

The output of the encoder is a sequence of so-called hidden states. Commonly, these hidden states are inputs to the “prediction head” of the language model, which produces a probability distribution over the token space for the corresponding token location.

We think of a language corpus  $C$  as a collection of tokenized portions of text. From the point of view of a language model, each instance  $i$  of a particular token appears in a specific context  $\mathbf{X}(i)$ . These contexts are filled with padding tokens so that they always have length  $N$ , permitting construction of the embedding sets  $e(\mathbf{X}(i))$ .

**Definition 2.1** *Given an encoder  $e$  derived from a pretrained language model, the ambient corpus datastore with respect to a corpus  $C$  is the multi-set<sup>1</sup>/point cloud of all the embeddings  $e(\mathbf{X}(i))$  of all instances  $i$  of all tokens in  $C$ .*

Note that we cannot explore the entire latent space of the language model, but only the subspace “carved out” by the datastore under consideration,

<sup>1</sup>We write *multi-set* to allow for repetitions/multiplicities. This is relevant in our setting, because strings might appear multiple times in the corpus.

as in Definition 2.1. In other words, by selecting the task-dependent ambient corpus for sampling the language model hidden states, we are making a choice of how we explore the hidden state space. This choice of ambient corpus may have a big impact on the derived features.

### 2.2 Local Topological Measures

All our topological measures are based on neighborhoods of a given contextual embedding vector  $v$  with respect to a collection of contextual embedding vectors coming from an ambient corpus datastore. Given an integer  $n \geq 1$ , we define the *neighborhood*  $\mathcal{N}_n(v)$  as the multi-set consisting of  $v$  and its  $(n - 1)$  nearest neighbors. To avoid adding another copy of the query center vector  $v$  when building the neighborhood from the datastore, we first check for similarity to existing vectors in the datastore with a Euclidean distance threshold of  $10^{-4}$ , and take a possible match as center vector if applicable. For a schematic illustration of the neighborhood extraction process and the feature computation, see Figure 1. We consider the following local features:

**Persistence Images** For a positive integer  $d$ , *persistent homology of degree  $d$*  associates with a point cloud a multi-set that encodes “ $d$ -dimensional topological features” of the cloud. We refer to Edelsbrunner and Harer (2010) or Otter et al. (2017) for introductions. Various vectorizations of this multi-set have been developed for subsequent use in machine learning. Persistence images are introduced in Adams et al. (2017) as a refined, higher-dimensional vectorization of persistent homology. We define  $\text{PI}^d(v) \in \mathbb{R}^{100}$  as a persistent image vector of the degree  $d$  persistent homology of  $\mathcal{N}_n(v)$ , scaled by a factor of  $\frac{1}{n \cdot 100}$ . The parameter  $n$  is not included in this notation, as it will be fixed to 128 throughout all experiments. For detailed definitions, see Appendix A.1. The factor  $\frac{1}{n \cdot 100}$  appearing in our definition of the persistence image is not important at this point. It is included to avoid instabilities in the training of the BIO-tagger discussed in Section 3, which may otherwise arise from the vastly different scales of the coordinates of the language model embedding vectors and these additional coordinates.

**Wasserstein Measure** A simple one-dimensional vectorization is the Wasserstein norm. We

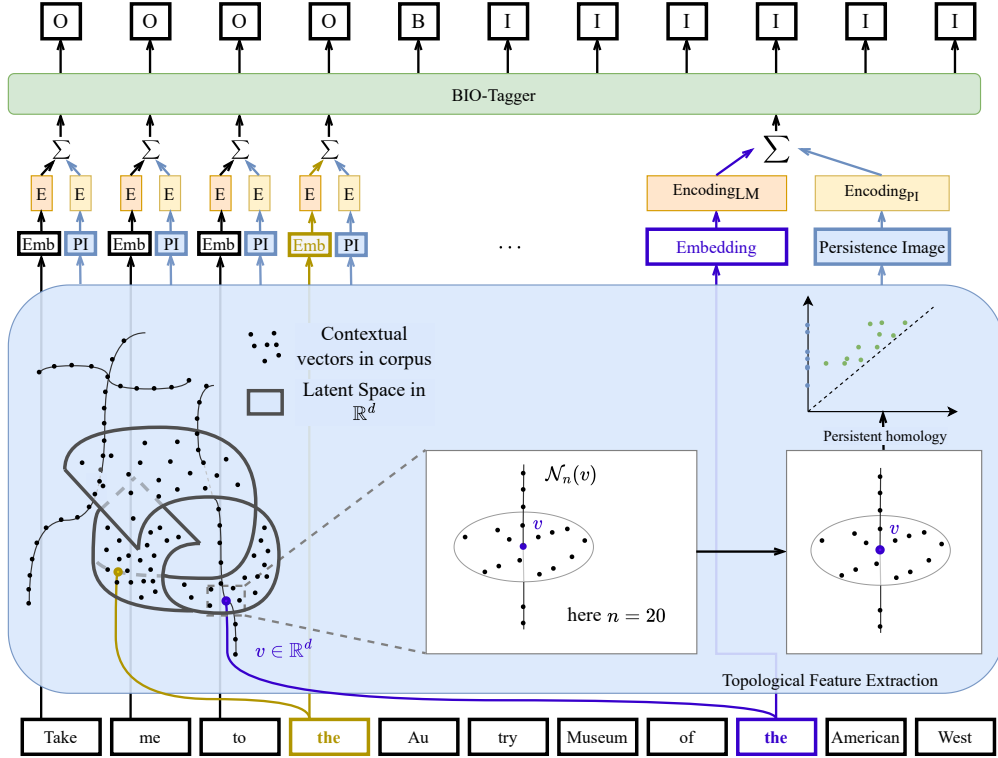


Figure 1: Schematic illustration of the local topological feature extraction and of our topological deep learning pipeline: The blue box illustrates the extraction of neighborhoods  $\mathcal{N}_n(v)$  in the contextualized embedding space, followed by the computation of each neighborhood’s topological features, resulting in a contextualized persistence image vector. Note the color coding of the different occurrences of the token ‘the’; contextuality leads to different language model embedding vectors and persistence images depending on whether it is part of the term ‘Auntry Museum of the American West’ or used as a non-content word. For each token, the language model embedding (Emb) and persistence image vectors (PI) are encoded (E), combined ( $\Sigma$ ), and then serve as input to our BIO-tagging transformer (green), which is trained on the token-level term labels (B-TERM (begin), I-TERM (inside), O (outside)).

define  $W_n^d(v) \in \mathbb{R}$  as the Wasserstein norm of the degree  $d$  persistent homology of  $\mathcal{N}_n(v)$ .

**Codensity** We define the  $n$ -th codensity  $\text{coden}_n(v) \in \mathbb{R}$  as the radius of  $\mathcal{N}_{n+1}(v)$ . Higher codensity corresponds to regions where the vectors are farther spread apart.

There are several reasons why we fix the cardinality  $n$  of the neighborhoods rather than, say, their radius. Firstly, fixing the cardinality takes into account sample density of the ambient corpus from the latent space of the language model. If we took a fixed radius, sparse regions of the ambient corpus space would be underrepresented. Secondly, some of the topological features we consider are more readily comparable when computed on fixed cardinalities. Indeed, a reasonable comparison of Wasserstein norms of neighborhoods of different cardinalities seems difficult, and our multiscale definition of (co)density could also not

easily be emulated for neighborhoods of fixed radius. Finally, computational complexity limits the feasibility of approaches that allow for unlimited cardinalities of neighborhoods. For instance, in [Von Rohrscheidt and Rieck \(2023\)](#), where neighborhoods of fixed radii are employed, an additional sampling step is necessary. More on this is discussed in [Appendix A.1](#).

### 3 Application of Local Topology Measures to Token Level Tagging Task

We perform a correlation analysis of local features and conduct a case study to explore the efficacy of local topology measures. Specifically, we apply our proposed topological feature augmentations to the task of dialogue term extraction.

#### 3.1 Set-Up

**Data** For the term extraction case study, we resort to the MultiWOZ2.1 ([Budzianowski et al.](#),



2018; Eric et al., 2020) and schema-guided dialogue (SGD) (Rastogi et al., 2020) task-oriented dialogue datasets. Here, the ambient reference corpus  $C$  is built solely from the language model hidden states for the training corpus of MultiWOZ2.1, so that the local measurements are comparable across both datasets.

**BIO-Tagging** For the sequence tagging tasks, we employ a beginning (B), inside (I), outside (O) labeling schema, as in Qiu et al. (2022). To keep the comparison between our different models fair and to obtain statements about the quality of the underlying features, we choose the architectures so that the trainable BIO-tagging components have a similar number of adjustable parameters. In this way, we can safely attribute any increase in performance to our topological augmentation of the input features rather than a stronger tagging component.

In all cases, the BIO-tagging transformer follows the RoBERTa architecture (Liu et al., 2019) and uses 8 attention heads, 2 hidden layers, and 512 maximum position embeddings. The language model vectors and augmenting feature vectors are fed into the BIO-tagging component through separate two-layer fully connected encoding networks with subsequent individual layer normalization, whose purpose is down- or up-scaling the feature dimension (768 for the language model vectors, 100 for persistence images) to the hidden size (512) of the tagging transformer. For a schematic of our BIO-tagging setup, see Figure 1.

**Features** For creating the language model embeddings, we use the second to last hidden states (at layer 11) of the pretrained RoBERTa base model (Liu et al., 2019), which returns 768-dimensional vectors, with  $L_2$ -normalization. Note that on unit vectors, the cosine distance is proportional to the square of the Euclidean distances, thus for the relative order in which the nearest neighbors occur, it does not matter whether we search with respect to the cosine or the Euclidean distance.

We decide on the *second-to-last* hidden state of the language model, as opposed to another intermediate layer, for two reasons: Cai et al. (2021) show that the local intrinsic dimension tends to increase with the depths in the transformer, thus the resulting neighborhoods should be more expressive. Moreover, Peters et al. (2018) and Tenney et al. (2019) find that deeper layers in language models tend to capture more of the semantic properties, while earlier layers tend to capture the syntax. For

feature based learning, Devlin et al. (2019) report that among single-layer features, the second-to-last layer leads to the highest performance. Note that our setup is not specific to the RoBERTa model or tokenizer. Our contextualized topological features can be computed for any (masked or causal) embedding model, extraction layer, datastore produced by the model, and query dataset.

As a baseline in our term extraction experiments, we use the language model hidden state vectors described in Section 3 as input for the BIO-tagging model. We test these against augmentation of the hidden states with our local persistence image descriptors introduced in Section 2.2.

### 3.2 Correlation Analysis of Local Features

We begin by collecting statistical observations on the correlation between the different local topological feature types, as well as their correlation with *pseudo-perplexity*, on the example of the MultiWOZ2.1 and SGD datasets. The *perplexity* of a causal language model is a model intrinsic measurement of the surprise of seeing the next token, defined as the exponentiation of the cross-entropy between the model prediction and the corpus data. While causal perplexity is not available here, in our the masked language model setting, we apply a pseudo-likelihood score by masking tokens one by one, and computing the prediction loss of the masked out token following Salazar et al. (2020).

In addition to the non fine-tuned version of the language model, here we also include the perplexity of the fine-tuned version for comparison, which uses the MultiWOZ2.1 training portion (fine-tuning for 5 epochs, 0.15 masking proportion, selecting the best model on MultiWOZ2.1 validation loss). All local features are based a non fine-tuned version of the language model.

Note that we cannot directly compute correlations between the vector-valued persistence images and the perplexity measures. For this reason, we are relying on selected codensities  $\text{coden}_n(v)$  with values  $n \in [1; 127; 511]$  and Wasserstein norms as numerical proxy estimates of the neighborhoods' topological complexity. Our term extraction models will take only the persistence images as input, as they provide the most powerful and comprehensive representation of the local topology. The neural network feature extraction model can learn directly from the persistence images to estimate complexity measures approximating the Wasserstein norms, which avoids manual feature engineering.

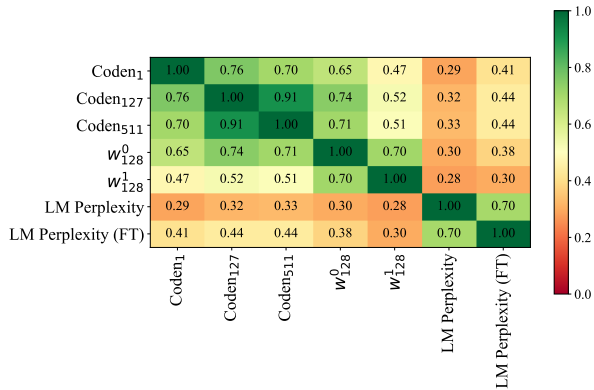


Figure 2: Kendall’s rank correlation coefficients between various local estimates and language model (LM) perplexity for the SGD test dataset. FT stands for LM fine-tuned on the MultiWOZ2.1 training split. All correlations have  $p < 10^{-6}$ .

The results of Kendall’s rank correlation are given in Figure 2. The Wasserstein measures are strongly positively correlated with the codensities. More importantly, the codensities and Wasserstein norms are only weakly correlated with the perplexity, indicating that these topological measures (and thus persistence images) capture information that is not present in the language model masked-token prediction distribution, and that cannot be ‘distilled’ into the language model via naive fine-tuning.

### 3.3 Case Study: Term Extraction in Task-Oriented Dialogue Data

#### 3.3.1 Task Definition

We approach dialogue term extraction as a transfer learning problem. Here, MultiWOZ2.1 serves as our source dataset used for training a term extractor. The trained model is then applied to the SGD dataset, necessitating sufficient transfer learning capabilities to properly handle the distributional shift in the data. We label all phrases in all utterances that match an entry in the respective dataset’s ontology, i.e., that match a value in a non-categorical slot of the current turn’s dialogue state or a value in the current turn’s dialogue act. The dataset ontology entities are normalized and matched to the occurrences in the respective utterances by applying the TripPy-R label map (Heck et al., 2022) and the SGD canonical value mapping. The ontology comprises names of entities, their domains, properties (slots), and values of these slots. We refer to these labelled phrases as *dialogue terms*.

These tagged spans for the dialogue datasets are encoded for the BIO-tagger, resulting in the three

labels: O (outside), B-TERM, I-TERM (begin and inside a term). Since our BIO-tagging model operates on the token-level of the underlying language model, we re-align the tags with the tokenization using the IOB2 schema: for a word with B-tag, the first subtoken is tagged with B, its remaining subtokens with I. For a word with I-tag, all its subtokens are tagged with I; and analogously for the O-tag.

While we employ token-level cross-entropy loss as the differentiable target function in the model training, the objective of term extraction within the context of this work is to retrieve each unique target dialogue term at least once. That is, we do not require the tagger to find all occurrences: We normalize the predicted phrases and ground truth by lower-casing, and deduplicate the resulting collections. A term is considered a true positive if it is identical to exactly one ground truth term. If a term cannot be assigned to any ground truth term, e.g., comprises several ground truth terms or is an incomplete substring of a ground truth term, it is considered a false positive. The left-over ground truth terms without a matched prediction are the false negatives. We call the resulting prediction, recall and F1-scores the *phrasal results*.

We train the BIO-tagger for 10 epochs, using the AdamW optimizer (Loshchilov and Hutter, 2019) with learning rate  $5 \cdot 10^{-5}$ , linear warm-up for 10% of training steps and batch size of 48. The model predictions on the held-out MultiWOZ2.1 validation set are evaluated every 100 batches for the first 3 000 global steps, and the model checkpoint with the best phrasal results on the validation set is selected as the final model.

Our goal is to show that injecting our local topological features into the model yields statistically significant improvements over the original language model embeddings. We run statistical tests over multiple different random seeds for initialization and check for significant changes in evaluation scores on the transfer set, which we take as the full collection of 463 284 utterances from the SGD dataset comprising 20 domains.

**Training Data** In the full data setting, we train on all 113 556 utterances of the MultiWOZ2.1 training split, the results are included in Table 1.

To demonstrate that the contextual topological features are useful in settings with reduced data and might help in mitigating overfitting, we create a variation of the transfer task by restricting training to subsets of the MultiWOZ2.1 dataset. This is

a more realistic transfer setting, since a good model checkpoint needs to demonstrate that it can generalize to the unseen left-out domain which it encounters in the MultiWOZ2.1 validation split for the first time. Given one of the five major domains  $\mathcal{D} \in [\text{attraction}; \text{hotel}; \text{restaurant}; \text{taxi}; \text{train}]$  in the training split, we exclude those utterances contained in dialogues from  $\mathcal{D}$  in the tagger training, which leaves [71 768; 59 222; 58 156; 86 568; 66 736] utterances respectively. Model selection is performed based on phrasal F1-score on the 14 748 validation utterances, which span over all five domains. We report results of this cross-validation setup by macro averaging the phrasal scores on the SGD dataset over 10 seeds for each of the five left out data folds in Table 2.

**Static Topological Features Baseline** We evaluate term extraction performance on the level of phrase predictions. The phrase-level evaluation allows a comparison with Vukovic et al. (2022), who present a method that employs *static* topological descriptors in sequence tagging tasks. The main differences to our approach with contextual topological features are as follows:

- Our local persistent homology descriptors are defined on token level with respect to the tokenization of the language model. This is essential for combining our new features with the language model embeddings to create fusion models which can provide predictions on token-level. The *static* topological features of Vukovic et al. (2022) operate on word level, and they only gained contextuality in the BIO-tagging component of the model. Note that in this and our work, the context of an input sequence is a single dialogue utterance.
- Vukovic et al. (2022)’s features were based on neighborhoods in an ambient static word space composed of the 100 000 most common words in the English language. Thus, their method depends both on having a word-level separation of the input data, and on a given dictionary. Our contextualized features on the other hand can be defined without any additional external data.

For a comparison between static and contextual features, we align the static topological features with the roberta-base tokenization in our BIO-tagging setup, and train BIO-taggers with the

Input features	Precision $\uparrow$	Recall $\uparrow$	F1 $\uparrow$
LM roberta-base	48.89	56.61	52.39
LM roberta-base $\oplus$ static PI <sup>0</sup>	49.33	<b>58.82</b>	53.62
LM roberta-base $\oplus$ contextual PI <sup>0</sup>	<b>50.26*</b>	58.44	<b>53.97*</b>

Table 1: Phrasal-level performance comparison for term extractors trained on the MultiWOZ2.1 training split and evaluated on the full SGD dataset. Results are averages over 15 seeds. \* indicates statistically significant differences (one-sided independent *t*-test) w.r.t. the baseline LM roberta-base with  $p < 0.05$ .

Input features	Precision $\uparrow$	Recall $\uparrow$	F1 $\uparrow$
LM roberta-base	47.97	56.07	51.67
LM roberta-base $\oplus$ static PI <sup>0</sup>	47.92	<b>56.85</b>	51.94
LM roberta-base $\oplus$ contextual PI <sup>0</sup>	<b>48.92*</b> <sup>LM</sup> <sub>*stat</sub>	56.23	<b>52.24</b>

Table 2: Cross-validated phrasal-level performance on SGD for term extractors trained on MultiWOZ2.1 training split without selected domain in [attraction; hotel; restaurant; taxi; train] averaged over 10 seeds for each of the five folds. \* indicates statistically significant difference with  $p < 0.05$ , w.r.t. the baseline LM roberta-base (LM) and augmentation with static persistence images (stat).

same architecture and data as in the contextual topological feature setting. To that end, the first constituent subtoken of each word is augmented with the word’s corresponding 100-dimensional  $H_0$ -persistence image feature vector of Vukovic et al. (2022).

### 3.3.2 Results

**Quantitative Analysis** Table 1 lists the term extraction performance for the pure language model baseline, our proposed method of augmenting with contextual topological features, and the alternative approach by augmenting with the static topological features from Vukovic et al. (2022). The main objective of said work was the maximization of recall, and to that end they proposed separate language model and topological feature taggers, with a subsequent union of predictions. In contrast, we show that our unified model augmented with contextual topological features can increase precision, recall, and F1 over the language model baseline.

Table 2 presents averaged results for the models trained on a reduced dataset constructed by omit-

Ground Truth Terms	LM roberta-base (our baseline)	LM roberta-base $\oplus$ static PI <sup>0</sup>	LM roberta-base $\oplus$ contextual PI <sup>0</sup>
“cafe jolie”	“cafe jolie”	“cafe jolie”	“cafe jolie”
“angelina jolie”	–	“angelina jolie”	“angelina jolie”
“yellow chilli” “the yellow chilli by chef sanjeev Kapoor”	“yellow chilli” “the yellow chilli” “the yellow chilli by” “sanjeev Kapoor”	“yellow chilli” “the yellow chilli” “the yellow chilli by chef” “sanjeev Kapoor”	“yellow chilli” “the yellow chilli” “the yellow chilli by chef sanjeev Kapoor”
“water seed” “water seed concert”	“water seed” “the water seed” “water seed event”	“water seed”	“water seed” “water seed concert”
“be alright”	–	“alright” “be alright”	“alright economy”

Table 3: Representative examples of predictions where the baseline model fails to retrieve the correct term, while a local topology feature augmented model succeeds. We indicate **true positives** and **false positives** by color.

ting a given domain in the MultiWOZ2.1 training set. Here, on average, the augmentation with the contextual persistence images is again better than the language model vector baseline.

**Qualitative Analysis** To obtain explicit examples, we select a model checkpoint for each feature type after 1 100 global steps, and inspect the differences between predicted normalized phrase sets. In Table 3 we see examples where the topologically augmented model succeeds in finding complete multi-word terms, whereas the baseline model cuts off before the end of a term or misses proper names that should follow a preposition. Such information is highly dependent on the context of a term, and the contextual topological model is able to find long terms more consistently. All models identify the restaurant name “Cafe Jolie” correctly, but only the topological models recognize the actress “Angelina Jolie”. Similarly, the song title “Be alright” containing the frequent word “alright” is not recognized by the language model alone, but can be detected by a topological model.

### 3.4 Relation to the Manifold Hypothesis

At first glance, our results may appear to be at odds with the *manifold hypothesis*, a common assumption underlying many representation learning paradigms. While this hypothesis has been questioned for *static* word embeddings (Jakubowski et al., 2020), it remains uncontested for contextual embeddings. It posits that the latent space of a trained machine learning model is clustered around a disjoint union of lower-dimensional manifolds (Bengio et al., 2013; Brown et al., 2023). This implies that, from a purely topological perspective, the local structure of the latent space is constant,

at least along connected components – every point should have a neighborhood topologically identical to an open ball in some Euclidean space. How then is it possible that we can extract meaningful information from variations of the local topology?

There are at least two answers to this. First, all our measures depend on the way data is *sampled*. There is no reason to assume that the embeddings drawn from a given corpus provide a uniform sample of the latent space. On the contrary, the distribution of these samples will depend heavily on the corpus. And within a given corpus, we might expect the neighborhoods of latent vectors of content words to be “more spread-out” and of higher dimension than those of non-content words, since there are more plausible possibilities for filling in content words in a text than for non-content words. Second, our measures are based on persistent homology, which is known to detect not only topological properties but also differentiable structure such as curvature (Bubenik et al., 2020). Thus, even on a uniformly sampled manifold, these measures are expected to vary.

### 3.5 Computational Complexity

In this section, we address the computational overhead coming from our proposed method of augmenting a sequence tagger with contextual topological information of a given corpus. The one-off computational costs for the datastore, in our study derived from the MultiWOZ2.1 training dataset, and the query datasets (MultiWOZ2.1 training & validation dataset, and SGD dataset) involve a single embedding model forward pass for each input sequence.

For each query dataset relative to the datastore, assuming a constant and small neighborhood



size  $n$ , the asymptotic complexity of the neighborhood search depends on the tokenized cardinality of the query dataset  $|Q|$ , the tokenized cardinality of the datastore  $|C|$ , and the embedding dimension  $d$ . The runtime complexity using the exact search implementation from (Johnson et al., 2021) is  $\mathcal{O}(|Q||C|d)$ , and the storage complexity for neighborhood indices is  $\mathcal{O}(|Q|n)$ .

The persistent homology computation in dimension 0 for each query vector depends on the neighborhood size  $n$  as well. For degree 0, the number of simplices in the Vietoris-Rips complex can be upper-bounded by  $n^2$ . Thus, the persistence diagram for each neighborhood can be computed in  $\mathcal{O}(n^{2\omega})$ , where  $\omega < 2.4$  is the matrix multiplication exponent (Milosavljević et al., 2011). There are at most  $n$  generators in the 0-dimensional persistence diagrams, so the computation of the Wasserstein norms can be achieved in  $\mathcal{O}(n^3)$  (Lacombe et al., 2018). Empirically, the computation of the persistence images is observed to be very quick compared to the computation of the persistence diagrams.

Once computed and cached, these topological features can be reused for different training objectives on the given query dataset. The only overhead in transitioning from the baseline tagger (approximately 35.65 million parameters) to the tagger with input LM roberta-base $\oplus$ contextual PT<sup>0</sup> involves a few additional parameters (roughly 60 000) for the encoding module of the 100-dimensional contextual persistence image. Consequently, once the topological features have been cached, the training and inference of the topologically augmented BIO-tagger are only negligibly slower than the baseline BIO-tagger. Appendix A.1 discusses the software packages used in the implementation and how we handle caching of the precomputed neighborhoods and resulting contextual topological features.

## 4 Conclusion

In this work, we introduce a topological deep learning approach to enrich feature learning-based sequence tagging methods with contextual topological data. Our methods do not depend on access to the underlying feature creation method, nor on external knowledge bases. Once these local topological descriptors are computed, they offer the potential for reuse across different tasks, thereby mitigating the initial computational investment. One limitation lies in our method still requiring labels

on the seed dataset. Though our results in the case study hint at a correlation between dialogue terms and higher Wasserstein norms, we have yet to establish a clear-cut purely feature based criterion for distinguishing terms from non-terms in dialogue data.

Looking ahead, we conjecture that the utility of our approach extends beyond the term extraction task investigated in this study. Given its generic design and challenge, it is plausible that it is applicable to other language models and modalities. Although our empirical evaluations have been confined to masked language models, the difficulty of the term extraction task provides optimism that our method could be advantageous in other scenarios where understanding the relation between individual data points and a datastore is critical.

**Reproducibility Statement** The MultiWOZ2.1 and SGD datasets are publicly available through the ConvLab-3 unified data format (Zhu et al., 2023), and we release our preprocessing, local topological feature computation and tagging model training code.<sup>2</sup>

## 5 Limitations

The experiments have been confined to a small masked language model (RoBERTa base model). Our proposed method can be applied to embedding spaces derived from causal LLMs (BehnamGhader et al., 2024), but current state-of-the-art models typically produce latent spaces with significantly larger embedding dimension. This has great influence on the computational complexity required to generate the contextual topological features. While our BIO-tagger can be trained on a single V100 GPU with 16 GB of VRAM in 2 hours, one should note that for efficiently precomputing the nearest neighbors in the contextual topological feature extraction, the embedded datastore needs to fit into the graphic card memory. This one-off neighborhood computation is not an issue for the MultiWOZ2.1 training set datastore, but might limit applications to larger corpus sizes. One possible remedy could be given by applying embedding space dimension reduction techniques such as (Kusupati et al., 2024) to the datastore before computing our topological features.

<sup>2</sup><https://gitlab.cs.uni-duesseldorf.de/general/dsml/tda4contextualembeddings-public>

## Acknowledgments

BMR and RV are supported by funds from the European Research Council (ERC) provided under the Horizon 2020 research and innovation programme (Grant agreement No. STG2018 804636) as part of the DYMO project. CVN and HL are supported by the Ministry of Culture and Science of North Rhine-Westphalia within the framework of the Lamarr Fellow Network. MH and SF are supported by funding provided by the Alexander von Humboldt Foundation in the framework of the Sofja Kovalevskaja Award endowed by the Federal Ministry of Education and Research. Computational infrastructure and support were provided by Google Cloud. We want to thank the anonymous reviewers whose comments improved the quality of our paper.

## 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.
- Arthur Amalvy, Vincent Labatut, and Richard Dufour. 2023. [The Role of Global and Local Context in Named Entity Recognition](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 714–722, Toronto, Canada. Association for Computational Linguistics.
- Ulrich Bauer. 2021. [Ripser: efficient computation of Vietoris-Rips persistence barcodes](#). *J. Appl. Comput. Topol.*, 5(3):391–423.
- Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. [LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders](#). *Preprint*, arXiv:2404.05961.
- Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. [Representation Learning: A Review and New Perspectives](#). *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.
- Bradley CA Brown, Anthony L. Caterini, Brendan Leigh Ross, Jesse C Cresswell, and Gabriel Loaiza-Ganem. 2023. [Verifying the Union of Manifolds Hypothesis for Image Data](#). In *The Eleventh International Conference on Learning Representations*.
- Peter Bubenik, Michael Hull, Dhruv Patel, and Benjamin Whittle. 2020. [Persistent homology detects curvature](#). *Inverse Problems*, 36(2):025008, 23.
- Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. [MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.
- Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. [Isotropy in the Contextual Embedding Space: Clusters and Manifolds](#). In *International Conference on Learning Representations*.
- Gunnar Carlsson. 2014. [Topological pattern recognition for point cloud data](#). *Acta Numerica*, 23:289–368.
- Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. 2008. [On the Local Behavior of Spaces of Natural Images](#). *International Journal of Computer Vision*, 76:1–12.
- Gunnar Carlsson and Mikael Vejdemo-Johansson. 2021. [Topological Data Analysis With Applications](#). Cambridge University Press.
- Andy Coenen, Emily Reif, Ann Yuan, Been Kim, Adam Pearce, Fernanda Viégas, and Martin Wattenberg. 2019. [Visualizing and measuring the geometry of BERT](#). In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.
- David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Yuriy Mileyko. 2010. [Lipschitz functions have  \$L\_p\$ -stable persistence](#). *Foundations of Computational Mathematics. The Journal of the Society for the Foundations of Computational Mathematics*, 10(2):127–139.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Paweł Dlotko. 2017. [Persistence representations](#). In *GUDHI User and Reference Manual*. GUDHI Editorial Board.
- Herbert Edelsbrunner and John Harer. 2010. [Computational Topology: An Introduction](#). American Mathematical Society.
- Mihail Eric, Rahul Goel, Shachi Paul, Abhishek Sethi, Sanchit Agarwal, Shuyang Gao, Adarsh Kumar, Anuj Goyal, Peter Ku, and Dilek Hakkani-Tur. 2020. [MultiWOZ 2.1: A Consolidated Multi-Domain Dialogue Dataset with State Corrections and State Tracking Baselines](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages

- 422–428, Marseille, France. European Language Resources Association.
- Kawin Ethayarajh. 2019. [How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.
- Michael Heck, Nurul Lubis, Carel van Niekerk, Shutong Feng, Christian Geishausser, Hsien-Chin Lin, and Milica Gašić. 2022. [Robust Dialogue State Tracking with Weak Supervision and Sparse Data](#). *Transactions of the Association for Computational Linguistics*, 10:1175–1192.
- Alexander Jakubowski, Milica Gašić, and Marcus Zibrowius. 2020. [Topology of Word Embeddings: Singularities Reflect Polysemy](#). In *Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics*, pages 103–113, Barcelona, Spain (Online). Association for Computational Linguistics.
- Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. [Billion-Scale Similarity Search with GPUs](#). *IEEE Transactions on Big Data*, 7(3):535–547.
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. [Generalization through Memorization: Nearest Neighbor Language Models](#). In *International Conference on Learning Representations*.
- Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, Ekaterina Artemova, Serguei Barannikov, Alexander Bernstein, Irina Piontkovskaya, Dmitri Piontkovski, and Evgeny Burnaev. 2021. [Artificial Text Detection via Examining the Topology of Attention Maps](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 635–649, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. 2024. [Matryoshka representation learning](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Théo Lacombe, Marco Cuturi, and Steve Oudot. 2018. [Large Scale computation of Means and Clusters for Persistence Diagrams using Optimal Transport](#). In *Proceedings of the 32nd International Conference on Neural Information Processing Systems, NIPS'18*, page 9792–9802, Red Hook, NY, USA. Curran Associates Inc.
- Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. [Neural Architectures for Named Entity Recognition](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [RoBERTa: A Robustly Optimized BERT Pretraining Approach](#). *Preprint*, arXiv:1907.11692.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled Weight Decay Regularization](#). In *International Conference on Learning Representations*.
- Nikola Milosavljević, Dmitriy Morozov, and Primož Škraba. 2011. [Zigzag persistent homology in matrix multiplication time](#). In *Computational geometry (SCG'11)*, pages 216–225. ACM, New York.
- Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. 2017. [A roadmap for the computation of persistent homology](#). *EPJ Data Science*, 6(1).
- Rrubaa Panchendrarajan and Aravindh Amaresan. 2018. [Bidirectional LSTM-CRF for named entity recognition](#). In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.
- Ilan Perez and Raphael Reinauer. 2022. [The Topological BERT: Transforming Attention into Topology for Natural Language Processing](#). *Preprint*, arXiv:2206.15195.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep Contextualized Word Representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Liang Qiu, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. [Structure Extraction in Task-Oriented Dialogues with Slot Clustering](#). *Preprint*, arXiv:2203.00073.
- Lance Ramshaw and Mitch Marcus. 1995. [Text Chunking using Transformation-Based Learning](#). In *Third Workshop on Very Large Corpora*.
- Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. [Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset](#). In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*,



- AAAI 2020, *The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8689–8696. AAAI Press.
- Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. [Masked Language Model Scoring](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.
- Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. [BERT rediscovers the classical NLP pipeline](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.
- The GUDHI Project. 2015. *GUDHI User and Reference Manual*. GUDHI Editorial Board.
- Eduard Tulchinskii, Kristian Kuznetsov, Laida Kushnareva, Daniil Cherniavskii, Sergey I. Nikolenko, Evgeny Burnaev, Serguei Barannikov, and Irina Piontkovskaya. 2023. [Intrinsic Dimension Estimation for Robust Detection of AI-Generated Texts](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Lucrezia Valeriani, Diego Doimo, Francesca Cuturello, Alessandro Laio, Alessio Ansuini, and Alberto Cazzaniga. 2023. [The geometry of hidden representations of large transformer models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Julius Von Rohrscheidt and Bastian Rieck. 2023. [Topological Singularity Detection at Multiple Scales](#). In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Renato Vukovic, Michael Heck, Benjamin Ruppik, Carel van Niekerk, Marcus Zibrowius, and Milica Gašić. 2022. [Dialogue Term Extraction using Transfer Learning and Topological Data Analysis](#). In *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 564–581, Edinburgh, UK. Association for Computational Linguistics.
- Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. [GPT-NER: Named Entity Recognition via Large Language Models](#). *Preprint*, arXiv:2304.10428.
- Frank F. Xu, Uri Alon, and Graham Neubig. 2023. [Why do Nearest Neighbor Language Models Work?](#) In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.
- Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J Smola. 2017. [Deep sets](#). In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 3394–3404, Red Hook, NY, USA. Curran Associates Inc.
- Qi Zhu, Christian Geischauser, Hsien-chin Lin, Carel van Niekerk, Baolin Peng, Zheng Zhang, Shutong Feng, Michael Heck, Nurul Lubis, Dazhen Wan, Xiaochen Zhu, Jianfeng Gao, Milica Gašić, and Minlie Huang. 2023. [ConvLab-3: A flexible dialogue system toolkit based on a unified data format](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 106–123, Singapore. Association for Computational Linguistics.

## A Appendix

### A.1 Implementation Details

In our term extraction applications, the ambient vector datastore comprises a collection of vectors with cardinality in the millions, making the computation of neighborhoods a major computational bottleneck. To alleviate this issue, we employ the Facebook AI Similarity Search (faiss) module (Johnson et al., 2021) to precompute neighborhood indices using GPU acceleration. These indices can be reused in subsequent computations of our local measurements at varying scales. We obtain the neighborhood indices and distances for 1 024 neighbors using the faiss.IndexFlatL2 build from the MultiWOZ2.1 training datastore. This neighborhood cache allows extraction of the codensity measurements and the vectors required to subsequently compute persistence images for neighborhood size  $n = 128$ . Loading the 2 739 744 many 768-dimensional vectors from the roberta-base MultiWOZ2.1 training datastore into the faiss index requires approximately 8 GB of GPU memory.

The faiss library does not currently offer GPU support for range-based nearest neighbor search. This makes it infeasible to compute range-based neighborhoods at the scale of our dataset for the methods described in Von Rohrscheidt and Rieck (2023). This limitation is especially critical because our BIO-tagger requires topological features for each input token in each context.

Another computational challenge lies in the local persistent homology computations, which become a bottleneck when the goal is training a BIO-tagger based on the resulting features. To address this, we precompute and store the topological features and their vectorizations, including both persistence



images and Wasserstein norms. We use the Ripser library (Bauer, 2021) for computing the persistence modules for  $H_0$  and  $H_1$  with  $\mathbb{F}_2$ -coefficients w.r.t. cosine distance from the precomputed neighborhoods, and GUDHI (The GUDHI Project, 2015; Dlotko, 2017) for the vectorization and persistence representation. The Wasserstein norms, i.e., the order-1 Wasserstein distances between the neighborhood persistence diagrams and the empty diagram with Euclidean ground metric, are computed separately for the  $H_0$  and  $H_1$  persistence diagrams using the GUDHI library.

For the persistence image vectorization of the  $H_0$ -persistence module, we decide on a bandwidth of 0.01, image range on the  $y$ -axis of  $[0.0, 1.0]$ , resolution of  $1 \times 100$  and weight each persistence homology generator by its  $y$ -value. We restrict our computations to 0-dimensional and 1-dimensional persistent homology. This is not only due to Ripser's optimizations, which result in a faster runtime, but also to circumvent the potential exponential increase in the number of simplices in the filtered complex when considering higher dimensional Vietoris-Rips complexes on a point cloud.