Topology of Word Embeddings: Singularities Reflect Polysemy

Alexander Jakubowski
Heinrich Heine University
Düsseldorf
jakubowskialexander
@gmail.com

Milica Gašić
Heinrich Heine University
Düsseldorf
gasic
@hhu.de

Marcus Zibrowius
Heinrich Heine University
Düsseldorf
marcus.zibrowius
@cantab.net

Abstract

The manifold hypothesis suggests that word vectors live on a submanifold within their ambient vector space. We argue that we should, more accurately, expect them to live on a *pinched* manifold: a singular quotient of a manifold obtained by identifying some of its points. The identified, singular points correspond to polysemous words, i.e. words with multiple meanings. Our point of view suggests that monosemous and polysemous words can be distinguished based on the topology of their neighbourhoods. We present two kinds of empirical evidence to support this point of view: (1) We introduce a topological measure of polysemy based on persistent homology that correlates well with the actual number of meanings of a word. (2) We propose a simple, topologically motivated solution to the SemEval-2010 task on *Word Sense Induction & Disambiguation* that produces competitive results.

1 Introduction

Static word embeddings attempt to represent words by vectors in a high-dimensional vector space $\mathbb{R}^n$ in such a way that words of similar meaning are represented by (cosine) similar vectors, and vice versa. According to the manifold hypothesis, we should expect these vectors to lie within a lower-dimensional **word space** $\mathcal{W}$, a subspace of $\mathbb{R}^n$ that resembles a manifold. To what extent this hypothesis is true in this and other contexts is the subject of ongoing research (Fefferman et al., 2016). In this paper, we argue and demonstrate that for the word space $\mathcal{W}$, polysemy is a principal obstruction to any strict interpretation of the manifold hypothesis.

That polysemy presents a serious obstacle to the creation of adequate word vector representations is clear from the outset. Take, for example, a polysemous word like "mole". We would want the vectors representing "birthmark" and "counterspy" to be similar to the vector of "mole", but *not* similar to each other. This is impossible. In order for similarity of vectors to accurately encode similarity in meaning, we need vectors representing meanings, not words.

Let us therefore hypothesize a **space of meanings** $\mathcal{M}$ that accurately represents all possible meanings and their similarities. Our argument is a simple topological observation based on the relationship between this space $\mathcal{M}$ and the word space $\mathcal{W}$. For an idealised language, where there is a bijection between meanings and words, these two spaces would agree. For a natural language, however, multiple points of $\mathcal{M}$ get identified with a single point of $\mathcal{W}$. This process corresponds to a topological construction that we refer to as **pinching** (see Figure 2). It is easy to see that a space resulting from pinching cannot be a manifold. Thus, even if the space of meanings $\mathcal{M}$ satisfies the manifold hypothesis perfectly, the pinched space $\mathcal{W}$ cannot satisfy the hypothesis near polysemous words.[1]

Based on this intuition, and using tools from Topological Data Analysis, we introduce a measure for the polysemy of a word based on its vector embedding. Our experiments show that this **topological polysemy** (TPS) correlates well with the actual number of meanings that a word has. In addition, we present an approach to the SemEval-2010 task on *Word Sense Induction & Disambiguation* (task 14) (Manandhar et al., 2010). This approach is independent of TPS, but based on the same ideas. Despite its simplicity, it is almost on par with the best performing algorithm within the

[1] It may appear that a similar complication arises from synonyms, multiple words with a single meaning. However, synonyms are irrelevant for our analysis; see the discussion at the end of Section 3.1.

*Proceedings of the Ninth Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 103–113
Barcelona, Spain (Online), December 12–13, 2020

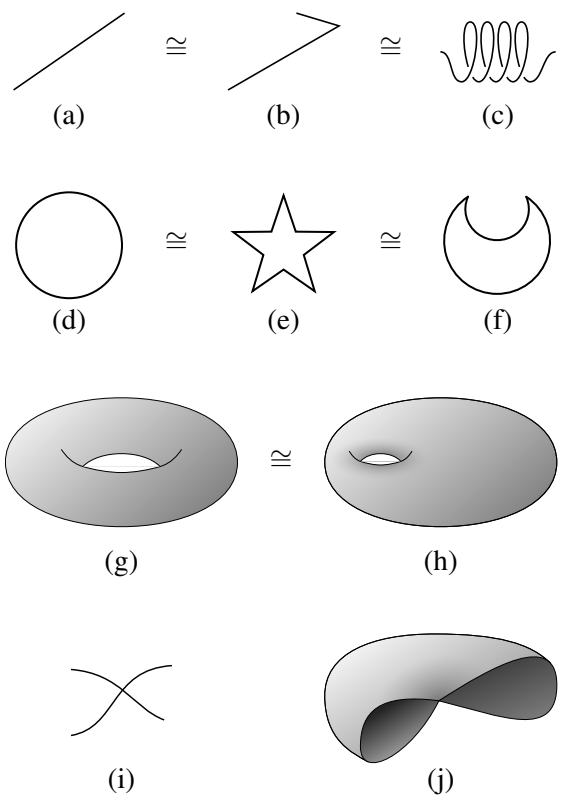

Figure 1: Some subspaces of $\mathbb{R}^3$: various deformations of an open line segment (a, b, c), deformations of a circle (d, e, f), a torus (g) and a deformation of the torus (h), two intersecting line segments (i), and a surface with a figure eight as boundary (j)

2010 challenge, and outperforms far more complicated approaches.

We see these experimental results as strong evidence that our interpretation of the word space $\mathcal{W}$ as a pinched manifold is more adequate than a more naïve view of $\mathcal{W}$ as an actual manifold.

2 Background

2.1 Topology

A space, for us, is a topological space. Readers unfamiliar with the notion may simply think of metric spaces, or indeed of subspaces of euclidean space $\mathbb{R}^n$. Two such spaces are considered equivalent, or **homeomorphic**, if they can be deformed into each other. We will not make this precise here, but we hope that Figure 1, in which homeomorphic spaces are connected by the symbol "≅", gives a clear intuition. As flexible as this notion may appear, the deformations considered do keep certain properties of a space invariant. Crucially, homeomorphic spaces always have the same number of connected components and the same number of holes. Topologists have developed a myriad of more subtle invariants that allow us to decide whether two spaces are homeomorphic. We refer to Hatcher (2002) for an introduction into this vast field.

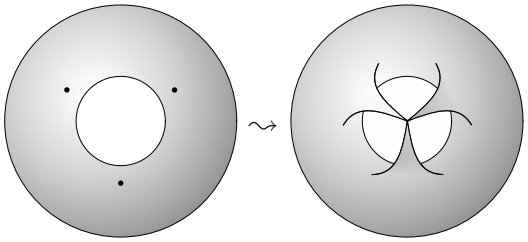

Figure 2: The effect of pinching on the torus (example (g) from Figure 1): before *(left)* and after the identification of three marked points to a singular point *(right)*

Two kinds of spaces will be important for us: manifolds and pinched manifolds. A (topological) **manifold** is a space in which each point has a neighbourhood homeomorphic to an open ball of $\mathbb{R}^d$ for some d (cf. Hatcher, 2002, § 3.3).[2] We call d the local dimension of the manifold at that point. The spaces (a), (b) and (c) in Figure 1 are manifolds since each point has a neighbourhood homeomorphic to an open interval in $\mathbb{R}^1$, and so are the spaces (d), (e), (f). The spaces (g) and (h) are manifolds because each point has a neighbourhood homeomorphic to an open disk in $\mathbb{R}^2$. Space (i), on the other hand, is not a manifold, because the point of intersection has no neighbourhood homeomorphic to an open ball of any dimension, and neither is space (j), because the manifold condition is violated at all boundary points. The spaces "without corners", i.e. examples (a), (c), (d), (g) and (h) in Figure 1 are not only topological manifolds but even *differentiable* manifolds, but this distinction will be of no importance to us.

By a **pinched manifold**, we will mean a space obtained from a manifold by marking a finite number of points in different colours, and identifying ("glueing together") all points of the same colour, as illustrated in Figure 2. In a pinched manifold, the neighbourhoods of most points still look like open balls, but the neighbourhoods of the identified points look like several balls glued together at their centres. We will call these identified points **singular points**.

Singular points can thus easily be distinguished from non-singular points by the topology of their neighbourhoods. More precisely, we can distin-

[2] Manifolds are moreover required to be *Hausdorff*, a technical condition that all metric spaces satisfy.

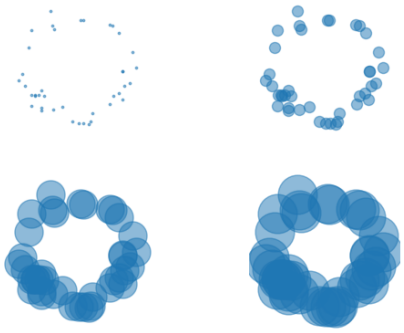

Figure 3: Points noisily sampled from the unit circle *(top left)* and the corresponding spaces $\mathcal{W}_r$ for different radii r

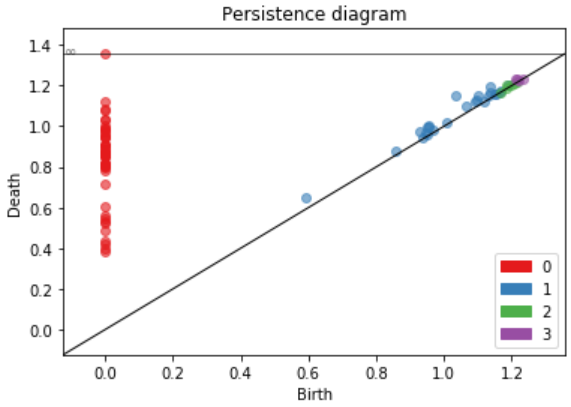

Figure 4: An example of a persistence diagram, summarizing the persistent homology of some point cloud $\mathcal{W}_0$ in degrees $i = 0, 1, 2$ and 3. Each dot encodes the life span of a distinct feature. Features of different degrees are displayed in different colours, as indicated in the lower right corner. For the computations in this paper, we will focus on the degree zero features, i.e. on connected components, indicated in red. As all of these are already present in the point cloud $\mathcal{W}_0$, they all have horizontal coordinate equal to zero. Their vertical coordinates are the radii at which different components merge

guish them by counting the number of connected components of their **punctured neighbourhoods**: neighbourhoods of a point from which the point itself has been removed. The punctured neighbourhood of a non-singular point is a single punctured ball, and thus is connected, at least in dimensions $d \geq 2$. The punctured neighbourhood of a singular point obtained by identifying $k > 1$ points, on the other hand, is a disjoint union of k punctured balls, and thus has several connected components. Thus, in dimensions $d \geq 2$, a point is singular if and only if small punctured neighbourhoods of it have more than one connected component.

Puncturing the neighbourhood, i.e. removing the centre, is crucial for this distinction. The unpunctured neighbourhoods of singular and non-singular points are not distinguishable by the usual topological invariants. (In technical terms, the neighbourhoods of both types of points are *contractible*.)

2.2 Topological data analysis

Topological data analysis (TDA) is an instrument for extracting topological information from a point cloud, that is a finite set of vectors $\mathcal{W}_0 = \{p_1, p_2, \ldots, p_N\} \subset \mathbb{R}^n$. The point cloud itself is trivial from a topological point of view. The fundamental assumption of TDA is that the vectors of $\mathcal{W}_0$ are not randomly distributed but instead are sampled from some underlying space $\mathcal{W} \subset \mathbb{R}^n$ which – unlike the point cloud itself – is topologically interesting. A human immediately recognises that the points in Figure 3 have been sampled from a circle. TDA provides algorithms that encode this intuition, and extend it to higher dimensions.

One such algorithm is **persistent homology**. The basic idea is to replace $\mathcal{W}_0$ with the union $\mathcal{W}_r$ of all open balls of a certain radius r centred at the points of $\mathcal{W}_0$. As we vary this radius, we obtain a sequence of spaces, starting for $r = 0$ with the point cloud itself and ending at some high value of r with a space in which all balls are merged into a single big blob. We compute certain topological invariants, the so-called Betti numbers b_i, for each space $\mathcal{W}_r$. The Betti number b_i counts certain i-dimensional features of the space. For example, b_0 is the number of connected components and b_1 is the "number of holes"; both are equal to 1 for the two spaces in the lower half of Figure 3.

The radii at which different i-dimensional features appear and disappear can be summarized into a multiset and visualized as a two-dimensional **persistence diagram** D as in Figure 4. Each dot in this diagram encodes the life span of a distinct feature: its horizontal coordinate is the *smallest* radius r at which the feature is present in $\mathcal{W}_r$, its vertical coordinate is the *largest* radius r at which it is present. Points that lie far off the diagonal correspond to features that *persist* across a wide range of values of r, and are hence likely to reflect features of the underlying space $\mathcal{W}$. For technical reasons, every point on the diagonal is also included in the persistence diagram D with infinite multiplicity.

The **Wasserstein distance** provides a notion of distance between two such persistence diagrams, and hence a measure of similarity between different point clouds and their underlying spaces. For two diagrams D and $\tilde{D}$ it is defined as:

$$W(D, \tilde{D}) := \inf_{\eta: D \to \tilde{D}} \left(\sum_{x \in D} \|x - \eta(x)\|_\infty \right)$$

where η runs over all bijections between the two diagrams. As all points on the diagonal are included in both diagrams, such bijections always exist.

The computation of persistent homology can be restricted to a range of degrees i. In this paper, we will concentrate on persistent homology in degree $i = 0$, which is essentially a systematic application of single-linkage clustering. Computations in higher dimensions quickly become very expensive. For an in-depth discussion of the concepts mentioned in this section we recommend (Edelsbrunner and Harer, 2010).

2.3 Word vector embeddings

The distributional hypothesis states that "the meaning of words lies in their use" (Wittgenstein, 1953). This provides the basis for distributional semantics, a data driven study of word meanings. Words are modelled as vectors in such a way that (cosine) similarity of vectors corresponds to similarity in the distributions of the corresponding words in natural language, and hence to semantic similarity. In the most naïve approaches, the dimension of these vectors corresponds to the number of distinct words in the language. More sophisticated implementations in which word vectors are real-valued but of significantly smaller dimension are popularly known as **word vector embeddings**. They have proven important for various tasks of natural language processing (Collobert et al., 2011; Lubis et al., 2020).

Early word vector embeddings were constructed in latent semantic analysis using singular value decomposition. Neural methods were introduced by Bengio et al. (2003), and popularised by the algorithms word2vec (Mikolov et al., 2013a,b) and GloVe (Pennington et al., 2014). Our method of choice in this paper is fastText (Bojanowski et al., 2017), which can produce high-quality embeddings from relatively small corpora. All of these methods produce **static** embeddings: they assign to each word a single, context-independent vector.

There is, of course, a lot of existing and ongoing research to overcome the difficulties inherent in adequately representing polysemous words. One way to address polysemy is to produce multiple, context-dependent embeddings for the same word. The deep learning approaches mentioned above are amenable to this by incorporating heuristics (Huang et al., 2012) or non-parametric clustering (Neelakantan et al., 2014). More recently, transformer based models that exploit massive datasets have been used to produce contextualised word embeddings. Examples of these are CoVe (McCann et al., 2017), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019) and its variants ERNIE (Sun et al., 2019) and RoBERTa (Liu et al., 2019). Alternative approaches address polysemy by training multi-lingual word embeddings on multi-lingual corpora (Dufter et al., 2018; Heyman et al., 2019).

As the problem of polysemy is, at least partially, resolved in all of these more advanced approaches, we would expect the phenomenon studied in this paper to be less pronounced in the embeddings they produce. We therefore concentrate exclusively on mono-lingual static embeddings. Our analysis will not require any data beyond such an embedding.

3 The topology of the word space

3.1 The word space as a pinched manifold

In order to explain the apparent efficiency of machine learning, the **manifold hypothesis** postulates that, in general, real world data tends to live on a small-dimensional submanifold of the vector space in which it is represented (Bengio et al., 2013; Fefferman et al., 2016). For word vector embeddings, the ambient space $\mathbb{R}^n$ typically has dimension n in the range $50 \leq n \leq 300$. The hypothesis states that word vectors in fact lie on, or are densely distributed around, a submanifold $\mathcal{W} \subset \mathbb{R}^n$ of much smaller dimension. What this hypothetical **word space** $\mathcal{W}$ might look like is an intriguing question. Work of Arora et al. (2018) suggests a dimension of $\mathcal{W}$ as low as five. It is easy to imagine even smaller subspaces of $\mathcal{W}$, like a line segment connecting "cold", "cool", "lukewarm", "warm" and "hot", or a circle connecting "north", "east", "south", "west". But the global structure seems mysterious.

The manifold hypothesis has two parts: (1) that $\mathcal{W}$ is of small dimension, and (2) that $\mathcal{W}$ is a manifold.[3] It is the second statement that we would

[3] It may not be evident what the correct notion of "dimension" should be for arbitrary subspaces. However, there are much larger classes of spaces than manifolds to which the notion

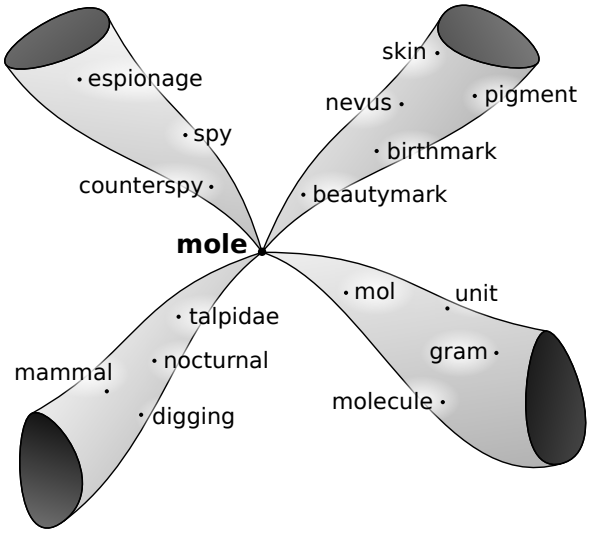

Figure 5: An idealized picture of the word space $\mathcal{W}$ near "mole": four regions of the meaning manifold are glued together to a single word

like to challenge. We argue that, in the vicinity of polysemous words, $\mathcal{W}$ cannot possibly have the structure of a manifold, i.e. it cannot resemble an open ball of any dimension. The best we can expect is that this might be true for some **space of meanings** $\mathcal{M}$ – a space that parametrizes all possible meanings that words of a given language may assume – from which $\mathcal{W}$ is obtained by identifying multiple meanings to a single word. This identification process is precisely the pinching construction discussed in Section 2.1. For example, we should expect the neighbourhood of the polysemous word "mole" in $\mathcal{W}$ to be obtained from the neighbourhoods of its different meanings in $\mathcal{M}$, all glued together as in Figure 5. Thus, even if we optimistically hypothesize the space of meanings $\mathcal{M}$ to be a manifold, the word space $\mathcal{W}$ cannot be: it is at best a *pinched* manifold. It is this hypothesis that we will pursue in the following. (If $\mathcal{M}$ has more complicated local structure, then *a fortiori* so does $\mathcal{W}$.)

The presence of synonyms in a language has no bearing on this analysis. To explain this, we need to temporarily distinguish carefully between a word w and its associated word vector $\mathbf{v}_w$. The word space $\mathcal{W}$ should more precisely be called *space of word vectors*, since this is the space in which the vectors $\mathbf{v}_w$ live, not the words themselves. Under a given word vector embedding, synonyms w and w' may get mapped to the same word vector $\mathbf{v}_w = \mathbf{v}_{w'}$. However, this does not affect the relation of the space of meanings $\mathcal{M}$ to the space of word vectors $\mathcal{W}$ in any way. The situation is summarized in Figure 6.[4]

of dimension extends in a straight-forward manner.

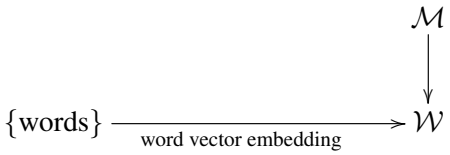

Figure 6: The relation of the space of meanings $\mathcal{M}$, the space of word vectors $\mathcal{W}$ and the set of words of a language. Synonyms may get identified under a given word vector embedding, symbolized here by the horizontal map. Multiple meanings get identified to a single word vector under the vertical map

With this discussion out of the way, we will from now on again simplify our terminology by identifying words with their associated vectors, and refer to $\mathcal{W}$ as word space.

3.2 A topological measure of polysemy

As explained at the end of Section 2.1, we can distinguish a singular point of a pinched manifold from a non-singular point by counting the connected components of a small punctured neighbourhood of the point. What is more, the number of these components reflects the number of points that were glued together in the pinching process. Thus, according to our view of the word space $\mathcal{W}$ as a pinched quotient of the manifold of meanings $\mathcal{M}$, the number of different meanings of a word should be reflected by the number of components of a punctured neighbourhood of the word.

Of course, the relevant number of components is not directly visible from the discrete point cloud formed by the word vectors. Rather, the components can only be estimated by some form of clustering. In this section, we describe a measure of the number of components based on degree zero persistent homology, as introduced in Section 2.2.

Fix a word vector embedding, a target word w, and a neighbourhood size n. As already indicated, we will abuse language by identifying a word with its vector under the embedding in the following. The **topological polysemy** $\mathrm{TPS}_n(w)$ of w with respect to our fixed word vector embedding and our chosen neighbourhood size n is the Wasserstein

[4] It is of course debatable whether the equation $\mathbf{v}_w = \mathbf{v}_{w'}$ would really hold for any pair of synonyms in practice. It seems more likely that the vectors $\mathbf{v}_w$ and $\mathbf{v}_{w'}$ would simply lie very close together.

norm of a normalized punctured neighbourhood of w. That is, $\mathrm{TPS}_n(w)$ is computed as follows:

1. Normalize all word vectors v to have L_2-norm $\|v\| = 1$.

2. Consider the punctured neighbourhood $\mathcal{N}_n(w)$ consisting of the n closest neighbours of w, excluding w itself.

3. Pass to the normalized punctured neighbourhood $\mathcal{N}'_n(w)$ by translating w to lie at the origin and projecting all vectors to the unit sphere:

$$\mathcal{N}'_n(w) := \left\{ \frac{v - w}{\|v - w\|} \;\middle|\; v \in \mathcal{N}_n(w) \right\}$$

4. Compute the degree zero persistence diagram of $\mathcal{N}'_n(w)$.

5. $\mathrm{TPS}_n(w)$ is the Wasserstein norm of this persistence diagram, i.e. the Wasserstein distance between the computed and the empty persistence diagram.

The general normalization in Step 1 is included because word embeddings are trained only on cosine similarity; the length of each vector has no apparent meaning. The normalization allows us to compute directly with difference vectors between word vectors of high cosine similarity. The normalization in Step 3 is included because we have fixed the *cardinality* n of the neighbourhood, not its diameter. Without any normalization in this step, we would be measuring mostly the density of the word cloud around w. The normalization by projection onto the unit sphere may seem somewhat radical, but it is topologically motivated: the topological invariants we use cannot distinguish a punctured ball from its boundary sphere. (In technical terms, the punctured ball and its boundary are *homotopy equivalent*; cf. Hatcher (2002), Chapter 0.)

4 Empirical evidence

We present two pieces of empirical evidence that support our view of the word space as a pinched manifold. The experiments in Sections 4.2 and 4.3 show that the topological polysemy defined above correlates with the actual number of meanings of a word. In Section 4.4, we describe a simple approach to the SemEval-2010 task on word sense induction based on our topological intuition.

"Don't forget the Tatun Mountains, which shelter the town. In the old days , Tanshui folk who cultivated farms on the slopes had to walk for an hour to get to their crops. These days you can take a local mini-bus."

Figure 7: An exemplary context of an instance of the target word "cultivate"

4.1 Experimental setup

All experiments are based on data provided with the SemEval-2010 task on *Word Sense Induction & Disambiguation* (Manandhar et al., 2010). The task is as follows: Assign a total of 8 915 **instances**, extracted from various sources including CNN and ABC, of 100 different polysemous **target words** (50 nouns and 50 verbs) to clusters based on their **context**, such that instances with different meanings get mapped to different clusters and instances with the same meaning get mapped to the same cluster. A context is simply a paragraph of text that the target word appears in. Figure 7 shows an exemplary context for an instance of the target word "cultivate". Note that labels are only provided for a test set; this is an unsupervised learning task.

The training set provided comprises 65 M occurrences of 127 151 different words. We use this corpus to train our own vector representations using the python module for fastText (Bojanowski et al., 2017). For the computation of the persistence diagrams and the Wasserstein distance we use the GUDHI library (The GUDHI Project, 2020).

4.2 Correlation of TPS with the SemEval gold standard

The SemEval data set includes a gold standard for the 100 target words. The number of clusters in this gold standard is equal to the number of true meanings of each word, as perceived by humans. Figure 8 shows our measure of polysemy $\mathrm{TPS}_{50}(w)$ for the 100 target words w plotted against these cluster counts.

Correlation coefficients between the gold standard and $\mathrm{TPS}_n(w)$ for varying neighbourhood sizes n are displayed in the first column of Table 1. We found the highest correlation for $n = 50$, equal to 0.424. Neighbourhoods consisting of just ten or less words are clearly too small to capture multiple meanings. On the other hand, for high values of n, the neighbourhoods become too large to adequately reflect the local structure of the word space around

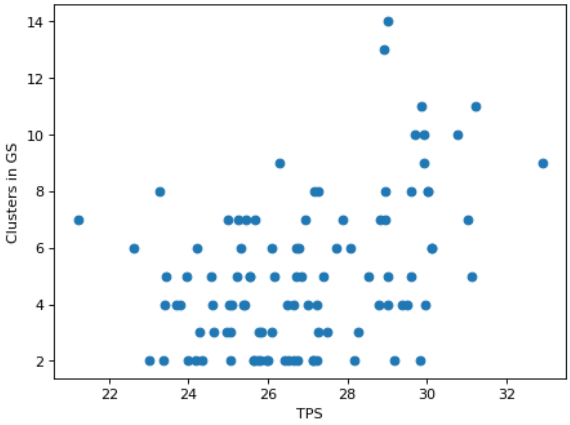

Figure 8: The topological polysemy $\mathrm{TPS}_{50}(w)$ plotted against the number of clusters in the SemEval gold standard, for the 100 SemEval target words w

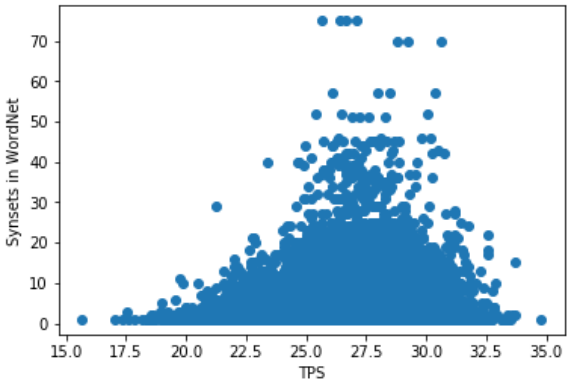

Figure 9: The topological polysemy $\mathrm{TPS}_{50}(w)$ plotted against the number of synsets in WordNet, for all 62 049 words w in the SemEval corpus that have a WordNet entry

n	**TPS_n vs. GS**	**TPS_n vs. synsets**	**TPS_n vs. frequency**
10	−0.001	0.122	0.002
40	0.411	0.096	−0.003
50	0.424	0.085	−0.006
60	0.414	0.076	−0.008
100	0.333	0.055	−0.013
sample size	100	62 049	127 151

Table 1: Correlations between $\mathrm{TPS}_n(w)$ and the number of meanings of w according to the SemEval gold standard (Section 4.2), the number of WordNet synsets (Section 4.3), and the frequency of w in the SemEval training corpus. The last line indicates the number of words on which the correlation is computed. The gray entry is not statistically significant, but all other entries are (p-value $< 10^{-3}$)

	GS	**WordNet**	**TPS_{50}**
sniff	3	3	27.262
reap	2	2	26.658
bow	5	14	28.533
chip	13	14	28.910
house	14	14	28.999

Table 2: Some examples of words and their corresponding cluster count in the SemEval gold standard and WordNet as well as their TPS-measure for $n = 50$

the target word. This is likewise unsurprising: recall from Section 2.1 that the manifold condition is a *local* condition around each point. Larger neighbourhoods of a point on a manifold can be arbitrarily complicated, and so can larger neighbourhoods of singular points on a pinched manifold.

The third column of Table 1 shows that $\mathrm{TPS}_n(w)$ does *not* correlate with the frequency of the words in the SemEval corpus. This is important, as frequency itself correlates with polysemy. The absence of correlation between $\mathrm{TPS}_n(w)$ and frequency strengthens our assertion that $\mathrm{TPS}_n(w)$ indeed measures polysemy.

4.3 Correlation of TPS with WordNet synsets

The correlation with the gold standard is a good indication of the validity of our method, but it is based on just 100 samples. The number of meanings, as perceived by humans, of a much larger set of words can be extracted from WordNet (Fellbaum, 1998), specifically the number of synsets associated with each word. Of course, we cannot expect the correlation between our topological polysemy and these numbers of synsets to be as high as for the SemEval gold standard. Firstly, we have trained our fastText vectors specifically on the SemEval training set, which does not capture the breadth of WordNet, and which does not comprise enough data to yield adequate embeddings for non-target words. Secondly, WordNet captures distinctions in meaning far more granular that one could hope to detect within the, say, 50 closest neighbours of a word.

Nonetheless, plotting $\mathrm{TPS}_{50}(w)$ against the number of synsets for all 62 049 words of the SemEval corpus that have a WordNet entry indicates a clear trend, see Figure 9. Correlation coefficients

for varying n are included in Table 1.

4.4 The SemEval task

Our hypothesis that the word space is a manifold pinched at polysemous words also suggests the following, direct approach to the SemEval-2010 task itself, which we call **Overlap with Punctured Neighbourhood (OPN)**. Fix a neighbourhood size n. In a first step, we cluster punctured neighbourhoods of size n of the 100 target words using a common clustering algorithm like k-means or dbscan (Ester et al., 1996). The different clusters of the neighbourhood cloud obtained in this way are taken to represent different meanings of the target word. In a second step, we assign a given instance of the target word to the cluster of the neighbourhood cloud that has the highest relative word overlap with the context of that instance.

For clustering with dbscan, we found that the best results are achieved with parameter values $\mathrm{Eps} = 0.09$ and $\mathrm{MinPts} = 2$ and large neighbourhood sizes n. The k-means clustering algorithm requires the number k of clusters aimed for as a parameter. We experimented both with fixed values of k and with a word-dependent variable value $k(w)$, predicted using TPS as follows. Define the TPS-percentile $\%(w)$ of a target word w as

$$\%(w) := \left\lceil \frac{\mathrm{TPS}_{50}(w) - \mathrm{TPS}_{\min}}{\mathrm{TPS}_{\max} - \mathrm{TPS}_{\min}} \cdot 100 \right\rceil,$$

where $\mathrm{TPS}_{\min}$ and $\mathrm{TPS}_{\max}$ denote the minimum and maximum values that $\mathrm{TPS}_{50}(\cdot)$ assumes on all target words, respectively, and where $\lceil \cdot \rceil$ denotes rounding to the next largest integer. Thus, the percentile is an integer between 0 and 100 that reflects how large $\mathrm{TPS}_{50}(w)$ is in comparison to all other target words. The expected number of clusters $k(w)$ is defined as

$$k(w) := \begin{cases} 2 & \text{if} \quad \%(w) \leq 1 \\ \%(w) + 1 & \text{if } 1 < \%(w) < 100 \\ 100 & \text{if} \quad \%(w) = 100 \end{cases}$$

Thus, the predicted number of clusters $k(w)$ varies between 2 and 100.

The performance is commonly measured by two scores, the F-score and the V-measure, which capture to what extent a clustering agrees with the gold standard clustering. Since both scores are important, we rank different approaches based on the product of these scores. This automatically discounts the performance of trivial approaches: MFS, which assigns each occurrence to the same cluster, and 1cl1inst, which assigns each occurrence to its own cluster. Table 3 shows the results for OPN with different clustering algorithms and different parameters. For comparison, the table moreover includes the best performing models of the SemEval task, as well as some other models published since. Our best performing set-up (OPN with dbscan, $n = 5000$) achieves the second best results, outperforming much more complex methods. Note that, unlike Arora et al. (2018) and Mu et al. (2017), we do not use any additional data to train embeddings.

For OPN with k-means clustering, we found that $k = 30$ gives the best results among possible fixed values for k. As Table 3 shows, the performance of our method with the TPS-informed variable value $k(w)$ is better than the performance with this fixed value. This provides further evidence to our claim that TPS is positively correlated with the true number of meanings. A comparison of the performance of OPN with dbscan and of OPN with k-means indicates that the size n of the neighbourhood to be considered for clustering needs to be an order of magnitude larger when we do not incorporate any information from TPS. Our interpretation is that TPS witnesses the disturbance that an additional meaning causes in a small neighbourhood of a word, even when no word related to that meaning is present in the neighbourhood.

In Table 4, we single out the three best performing and the three worst performing target words with our best performing model and give the associated scores as an illustration.

5 Conclusion

In this work, we challenge the manifold hypothesis for static word vector embeddings and experimentally show that it is more accurate and helpful to view the space of word embeddings as a pinched manifold. We introduce a topological measure of polysemy that correlates well with the number of meanings of a word according to the gold standard of the SemEval-2010 task on *Word Sense Induction & Disambiguation*. We also produce a surprisingly simple, but topologically motivated solution to the task itself that achieves highly competitive results.

We stress that our measure of polysemy, TPS, is computed solely on the topology of the point cloud consisting of the vectors of a fixed word vector embedding. Of course, any solution to the described

Method	Parameters	V-Measure	F-Score	Product
UoY (Korkontzelos and Manandhar, 2010)		0.157	0.498	**0.0782**
OPN with dbscan	$n = 5000$	0.175	0.420	0.0735
OPN with dbscan	$n = 2000$	0.135	0.493	0.0666
(Mu et al., 2016)	$k = 5$	0.145	0.441	0.0639
OPN with k-means	$n = 500$, $k = k(w)$	0.165	0.356	0.0588
KSU KDD (Elshamy et al., 2010)		0.157	0.369	0.0579
OPN with k-means	$n = 500$, $k = 30$	0.161	0.352	0.0567
(Arora et al., 2018)	$k = 5$	0.115	0.464	0.0533
(Mu et al., 2016)	$k = 2$	0.073	0.571	0.0417
OPN with dbscan	$n = 500$	0.070	0.571	0.0400
(Arora et al., 2018)	$k = 2$	0.061	0.586	0.0357
1cl1inst		**0.317**	0.090	0.0285
MFS		0.000	**0.634**	0.0000

Table 3: Performance of different methods on task 14 of SemEval-2010. According to our ranking by product of V-measure and F-score, the algorithms UoY and KSU KDD were the strongest contenders in the initial challenge. The algorithms MFS and 1cl1inst in the last two rows are trivial baseline algorithms

Target word	F-Score	Precision	Recall	V-Measure	Homo-geneity	Com-pleteness	Product
presume.v	0.827	0.957	0.728	0.477	0.683	0.366	0.3945
cultivate.v	0.657	0.648	0.667	0.518	0.564	0.479	0.3403
accommodate.v	0.465	0.476	0.455	0.605	0.777	0.495	0.2813
⋮							
violate.v	0.153	0.813	0.085	0.023	0.292	0.012	0.0035
root.v	0.574	0.405	0.984	0.000	0.000	1.000	0.0000
sniff.v	0.453	0.295	0.969	0.000	0.000	1.000	0.0000

Table 4: The performance of our best solution to SemEval on the three best performing vs. the three worst performing words, as evaluated according to the product of F-score and V-measure

SemEval task will also predict, in particular, the number of meanings of the target words. However, these predictions rely on access to the underlying corpus, or at least parts thereof. Similarly, the first step (clustering) of our solution to the SemEval task is performed directly on the word vectors, without recourse to any corpus. This is in sharp contrast with early clustering approaches to word sense disambiguation such as (Schütze, 1998) (which of course had to rely on far less sophisticated word vector embeddings than are now available).

A number of avenues could be pursued to further improve the results presented here. To allow a fair comparison with other solutions to the SemEval task, we have used word vector embeddings trained on a fairly small corpus. We have used only *degree zero* persistent homology. Our method of taking the Wasserstein norm of a persistance diagram is rather crude. The elimination of noise from the embeddings could also improve the results.

We conjecture that other NLP tasks that also rely, implicitly or explicitly, on the manifold hypothesis could similarly benefit from a more refined topological analysis.

Acknowledgments

We thank Claudius Zibrowius for Figures 1, 2 and 5 and Peter Arndt, Michael Heck and Carel van Niekerk for helpful discussions. The results of this publication are part of the project DYMO, which has received funding from the European Research Council under the grant agreement no. STG2018 804636. Computational resources were provided by Google Cloud.

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6(0):483–495.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

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.

Philipp Dufter, Mengjie Zhao, Martin Schmitt, Alexander Fraser, and Hinrich Schütze. 2018. Embedding learning through multilingual concept induction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1520–1530, Melbourne, Australia. Association for Computational Linguistics.

Herbert Edelsbrunner and John Harer. 2010. *Computational Topology: An Introduction*. American Mathematical Society.

Wesam Elshamy, Doina Caragea, and William H. Hsu. 2010. Ksu kdd: Word sense induction by clustering in topic space. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 367–370, USA. Association for Computational Linguistics.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231. Institute for Computer Science, University of Munich.

Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. 2016. Testing the manifold hypothesis. *J. Amer. Math. Soc.*, 29(4):983–1049.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA. https://wordnet.princeton.edu/.

Allen Hatcher. 2002. *Algebraic topology*. Cambridge University Press, Cambridge.

Geert Heyman, Bregt Verreet, Ivan Vulić, and Marie-Francine Moens. 2019. Learning unsupervised multilingual word embeddings with incremental multilingual hubs. 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 1890–1902, Minneapolis, Minnesota. Association for Computational Linguistics.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Association for Computational Linguistics (ACL)*, Jeju, Republic of Korea.

Ioannis Korkontzelos and Suresh Manandhar. 2010. Uoy: Graphs of unambiguous vertices for word sense induction and disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, page 355–358, USA. 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.

Nurul Lubis, Michael Heck, Carel van Niekerk, and Milica Gasic. 2020. Adaptable conversational machines. *AI Magazine*, 41(3):28–44.

Suresh Manandhar, Ioannis P. Klapaftis, Dmitriy Dligach, and Sameer S. Pradhan. 2010. Semeval-2010 task 14: Word sense induction & disambiguation. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, SemEval '10, pages 63–68, USA. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6294–6305.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2016. Geometry of polysemy.

Jiaqi Mu, Suma Bhat, and Pramod Viswanath. 2017. Representing sentences as low-rank subspaces. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 629–634, Vancouver, Canada. Association for Computational Linguistics.

Arvind Neelakantan, Jeevan Shankar, Alexandre Passos, and Andrew McCallum. 2014. Efficient non-parametric estimation of multiple embeddings per word in vector space. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1059–1069, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

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

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

The GUDHI Project. 2020. *GUDHI User and Reference Manual*, 3.2.0 edition. GUDHI Editorial Board.

Ludwig Josef Johann Wittgenstein. 1953. Philosophische Untersuchungen. §43.