P3: Topological Data Analysis for Word Embeddings

Rishabh Choudhary, Mithun Bharadwaj, Michael Rawson, and Samuel Dooley

November 26, 2019

1 Research Question

Can topological data analysis of word embeddings sufficiently recover word senses? Will this also work for pseudowords?

2 Proposed Approach

To answer our Research Questions, our project will use existing approaches in Topological Data Analysis to examine the shape of common word embeddings. Existing topological approaches have seen significant success in other data analysis areas, and are just beginning to be used in NLP applications (as we discussed in our literature review). Our research agenda will incldue the following high-level tasks.

Firstly, we will compute word embeddings with word2vec, GloVe, and fasttext on the datasets SEMEVAL-2013, SemCor, and SemCor+OMSTI. Then, we will use a topological data analsyis technique in persistent homology called barcodes (see our literature review) to understand the topology, or shape, of the word embedding space. We will analyze the results of this algorithm by finding the topological components that each word belongs to, as output from the barcodes algorithm. We hypothesize that this information will have some signal about the number of word sense for each word (details on this will follow). In addition, we will construct pseudowords by concatenating unrelated words, replace them in the corpus and check if the barcode algorithm recovers the senses of the two words.

We now describe in more specifics each part of the above plan. The majority of our time will be spent discussing the Barcode Algorithm, as it is the crux of the research, and also likely the most unfamiliar.

2.1 Word Embeddings

We will use the datasets SEMEVAL-2013, SemCor, and SemCor+OMSTI. We will compute word embeddings for each dataset using all three methods: word2vec, GloVe, and fasttext. This would output a total of nine word embeddings. We will perform this computation on the CSCAMM servers for mass parallelization. However, there may be resource constraints which will hinder our ability to compute all nine combinations; if this is the case, we will prioritize a sensible combination to allow for fair comparisons.

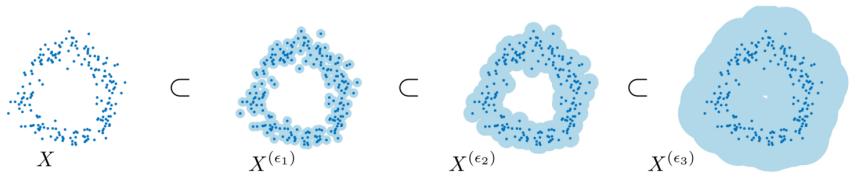

Figure 1: Illustration of Topology

2.2 Barcode Algorithm

Once we have the word embeddings, we will run a topological data analysis algorithm to test our Research Question.

Recall, that the study of topology examines the surfaces that are generating the data. For instance, we assume that our word embeddings are produced from some underlying manifold that has particular structures of interest. For instance, word embeddings make "similar" words cluster together in the embedding space. We note that the definition of similar is somewhat nebulous. In topology, we call these clusters *connected components* because if we were to connect each of the points/word embeddings to any other point that is say ϵ away, then they would form one connected piece. This point can be seen in Figure 1. Recall that persistent homology is the study of the evolution of these topological features as we increase ϵ.

We explain the barcode algorithm for the 0th persistent homology[1]. We start with a list of N data points, X, each in d dimensional Euclidean space. Then we calculate the distance between every pair of points, which corresponds to the e at which an edge is added between the pair in VR_e. Take the list of distances, E, and sort it in increasing order. Remove duplicate elements in E and call it D. We build a matrix, M, with a column for each edge and a row for each vertex in VR_∞ (the complete graph). Set $M(i,j)$ to t^a when i is a vertex of edge j and where a is the index of edge j in D which ranges from 1 to the length of D. All other entries of M are set to 0. We make the entries to be polynomials of variable t with coefficients restricted to 0 or 1. The next step is to column reduce this matrix so that what remains is lower triangular. Then the remaining nonzero diagonal entries, say t^b, correspond to the barcode or interval $(0, b)$. With this list of intervals, the algorithm is complete.

The output for this algorithm will be a diagram like in Figure 2. This diagram will tell us which are topological features of the word embedding space that are meaningful. In Figure 2 they are the three signals on the right which fall above the noise-thresholded diagonal. With the topological information output from the barcode algorithm, we can look at words that have multiple senses. The essential quesiton we ask will be, if we remove this word/point from the embedding, does the topology change. To think of this more concretely, consider the word "bank" which has a financial and a alluvial meaning, each with their own cluster of similar words. However, "bank" falls between them and so if it is removed, the connection between the two clusters will be broken and one connected component will become two. We shall compute these changes for various words to test our Research Question.

[1]See: https://www.math.upenn.edu/ ghrist/preprints/barcodes.pdf. Our description is inherently reductive because of the constraints on the length of the assignment.

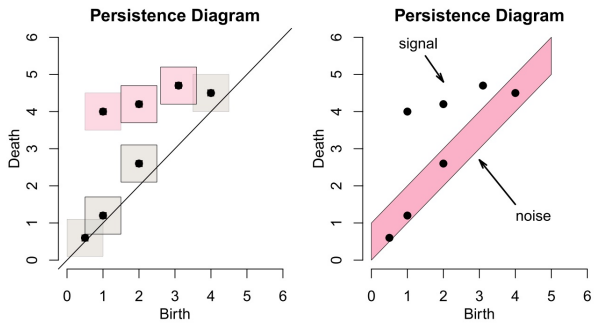

Figure 2: Diagram Algorithm Output

2.3 Pseudowords

Once we run the above barcode algorithm on the word embeddings for the different datasets and perform the analysis, we will repeat the process but with pseudowords. For a word that has multiple meanings or senses, we can break that word up into its component senses, create new dummy words, and replace the original word with the respective dummy word. For example, if the word "foo" has two meanings, we can create words "foo$1" and "foo$2" and replace "foo" with the respective pseduoword we just created.

Once we have done this, perhaps on some subset of the words in the datasets, we will replace the words in the corpora, retrain the embeddings, and rerun the barcode algorithm. We will proceed entirely as before with our analysis and draw conclusions about the appropriateness of topological data analysis in word sense disambiguation/induction.

3 Measures of Success

We will compare the number of senses calculated for each word with the ground truth from the annotated datasets SEMEVAL-2013, SemCor, and SemCor+OMSTI. We will compute the number of senses based on the topological notions described at the end of Section 2.2. We shall look at the topological features of a word with multiple sense. We will then compute how the topology changes when that word is omitted. The average relative error, comparing word by word, will be the measure of success with a perfect match of average relative error $= 0$. We will measure the success against the hyperparameters δ, the locality radius, and ϵ, the noise sensitivity. That is if g is the ground truth vector of number of word sense per word with length n, and $\tilde{g}$ is our approximate, then the average relative error is

$$e_{\delta,\epsilon} = \frac{1}{n}\sum_{i=1}^{n}\frac{|(g)_i - (\tilde{g}_{\delta,\epsilon})_i|}{(g)_i}.$$

As stated above, we will also calculate this error on the pseudowords to measure our success there.

This measure is appropriate for our Research Questions. The above measures the relative difference in word sense that our barcode algorithm predicts versus what are given as ground truth. This will measure effectively how this particular topological approach worked with

recovering word senses. However, we note that even if this measure is large, there might be useful information encoded with the barcode algorithm. Therefore, we will also perform qualitative analyses to derive reasons for the results.