PROMPT ENGINEERING FOR VISION-LANGUAGE MODELS THROUGH A TOPOLOGICAL LENS

CUSTOM PROJECT

MENTOR: PROF CHINMAY HEGDE

Ameya Joshi[1]*, Vaibhav Singh*[2]*, Akshit Gandhi*[3]

[1]ameya.joshi@nyu.edu, [2]vaibhav.singh@nyu.edu, [3]amg9556@nyu.edu

ABSTRACT

Zero-shot learning models like CLIP leverage natural language text as supervisory signals unlike traditional supervised learning approaches. They show great performance gains over other approaches. However, given that the labels consist of natural text, the choice of words and phrases used becomes important. We present a novel topological based approach that allows us to select text prompts that are most representative of a classes. We rely on MAPPER, a topology preserving projection algorithm to construct a Reeb graph of the embedding space, and further propose a margin-based approach to select subgraphs that provide the greatest predictive value. We show empirical evidence of the efficacy of our algorithm on zero-shot classification for Imagenette, a subset of the large Imagenet dataset. Consequently, we also discuss on the use of topological methods to analyse the effect of synonyms and other hypernyms on the performance of CLIP.

***Index Terms*—** Topological Mapping, language-vision models, explainability, MAPPER, Prompt Engineering

1. INTRODUCTION

Recent advances in training Vision-Language (VL) models like CLIP [1] and ALIGN [2] have noticeably advanced zero-shot classification. They demonstrate comparable performance to traditional supervised deep networks like ResNets [3]. An essential component of such models is the use of sentences and phrases as labels unlike traditional supervised learning which encodes a fixed label space. This allows for greater flexibility in terms of defining a learning task. Additionally, it is also a more "natural" approach to learning.

Training VL models involves two components: (1) Training an encoder for each input modality: images and text, and, (2) leveraging *Contrastive Learning* to then match corresponding text and image pairs while enforcing separation between non-matches. The goal here is to learn a common embedding space where in text and images with similar concepts are close to each other.

[†]Code:https://github.com/ameya005/clip_playground

Furthermore, Taori *et al.* [4] and Miller *et al.* [5] find that CLIP and related models are more robust to natural (semantic) and artificial transformations to images as compared to other state-of-the-art classifiers. The exceptional zero-shot performance in tandem with such surprising robustness properties lead to an important question– *Are there any VL-specific adversarial vulnerabilities?*

The primary challenge in answering this question is the design of the label space for VL models. Standard supervised learning tasks involve selecting a fixed set of labels apriori and using these as signals to train the model. In VL models, however, the label space during training consists of unstructured text. During inference, we construct a set of labels in the form of phrases and sentences. However, this leads to a very curious setup where we not only need to analyse the robustness of CLIP to the input images but also to the text inputs provided as label prompts.

In fact, we see several major challenges present themselves. Unlike the finite supervised learning label space, we now have a infinitely varying set of labels. For example, acceptable labels could include synonyms of objects of interest, varying forms of phrases and sentences and even using more descriptive words. In such a case, standard tools of machine learning interpretability (both theoretical and empirical) fail to provide useful analyses.

While the above challenges look unsurmountable at first glance, Topological Data Analysis (TDA) suggests a solution. It is well known that human languages demonstrate topological structure. In fact, topological data analysis has been used for natural language processing [6], understanding linguistic structure [7], understanding bias in large language models [8] and even detecting artificially generated text [9]. Therefore, TDA is a natural approach to analysing VL models. Further, TDA [10] has been used to analyse local explanations for deep models.

In this work, we focus on specifically on finding the most predictive prompts for a zero-shot classification problem. Our approach relies on using MAPPER to estimate a Reeb graph for to the CLIP embedding space. We observe that the graph consists of disjoint subgraphs separated by the margin. We

then leverage this to construct a heuristic based algorithm that selects the subgraphs that correspond to high margins. Each subgraph corresponds to a selection of prompts, that correspond to higher margins for each class. We also use MAPPER to analyse the effect of synonyms of class labels themselves and observe that CLIP is mostly robust to synonyms.

Our specific contributions are as follows:

1. We analyse the topology of the CLIP embedding space for prompts in terms of the margin.
2. We then propose a novel algorithm that leverages MAPPER to select the most predictive prompts.
3. We show empirically that our algorithm allows us to improve over prompt ensembling proposed by [1], and show an improvement of $\sim 1.3\%$ for zero-shot classification on Imagenette.
4. Finally, we also analyse the effect of synonyms on the CLIP embedding space and observe that CLIP is mostly robust to synonyms of class labels.

2. RELATED WORK

Topological Data Analysis(TDA) has been used both in statistical areas[11] as well as in various machine learning applications, [12], [13], [14]. Although one can trace back geometric approaches for data analysis quite far in the past, TDA really started as a field with the pioneering works of [15] and [16] in persistent homology. TDA is mainly motivated by the idea that topology and geometry provide a powerful approach to infer robust qualitative, and sometimes quantitative, information about the structure of data. In our work, we extensively utilize Kepler Mapper Algorithm first introduced by [17]. Mapper Algorithm aids in qualitative analysis, simplification and visualization of high dimensional data sets, as well as the qualitative analysis of functions on these data sets. It can be used to reduce high dimensional data sets into simplicial complexes with far fewer points which can capture topological and geometric information at a specified resolution. Thereby it provides both a method for mathematical data analysis and a visualization tool; the filter functions introduced through Mapper define a framework for supervised analysis. The output of the analysis approximates a collapse of the data into a simple, low dimensional shape, and the filter functions act as guides along which the collapse is done. It has been utilized in [18] to identify subgroup of cancer in high dimensional biological data. Similarly in [19], mapper algorithm finds its utility in understanding neural systems. [20] utilizes mapper in pattern recognition in point cloud data sets. In machine learning, [21], utilizes mapper for tailor-made design of metal–organic frameworks toward the desired target applications. In [22] Mapper algorithm is used to study the impact of gender based violence in news.

3. BACKGROUND

3.1. CLIP

We now present some background on CLIP, a vision-language model used for zero-shot classification. CLIP or Contrastive Language-Image Pretraining leverages natural language as a supervising signal for image classification. Formally, let $\{(x_i, y_i)\}_{i=0}^{N} \in X \times Y$ be paired image-text samples. The basic idea is to finetune two pretrained encoders: an image encoder, $f_\theta : X \to \mathbb{R}^d$ and an language encoder, $f_\phi : Y \to \mathbb{R}^d$ to maximise the cosine similarity between corresponding pairs while minimizing the same between non-matching pairs. This is achieved by constructing large batches of n matching pairs, and calculating $n^2/2$ possible inner products between all possible pairs. [1] propose the following contrastive loss function that minimizes the cosine distance between matching image-text pairs and maximizes the same otherwise,

$$L(\theta, \phi) = \sum_{i=1}^{n} \langle f_\theta(x_i), f_\phi(y_i) \rangle - \sum_{i \neq j}^{n} \langle f_\theta(x_i), f_\phi(y_j) \rangle.$$

The two encoders are then jointly trained by minimizing L using gradient descent.

For inference, we require two steps. First, we construct a set of possible labels, $Y_c = \{y_1, y_2 \dots y_k\}$ in the form of phrases or sentences. Radford *et al.*[1] note that CLIP performs better when y_i's are phrases or sentences instead of just class labels. They achieve this simply prefixing the class labels with phrases like 'a picture of –' to the labels defined for classification. They call these *prompts*.

The second step involves calculating the cosine similarity between the text encodings for Y_c and the input image x. The predicted class is the $y_i \in Y_c$ with the largest cosine similarity. This allows for zero-shot inference as Y_c could be arbitrarily different from Y. However, in practice, most empirical results involve using the same label set as Y (see Fig. 1 for a pictorial representation.). In order to reduce the effect of prompts on the overall classification, the authors propose constructing several prompts for each class label, followed by averaging their text embeddings. This is referred to as *prompt ensembling*.

Prompt ensembling provides two advantages: (1) the effect of individual prompts on the classification task is limited, and (2) additional context can be provided by ensembling over many relevant prompts. However, this also leads to a specific problem. In the case that most prompts provide misleading context, the model might misclassify an image. For example, if most of the prompts refer to "black and white" images, then CLIP would possibly misclassify an input colored image. We show in our experiments that this issue is non-trivial and actually does lead to CLIP making errors.

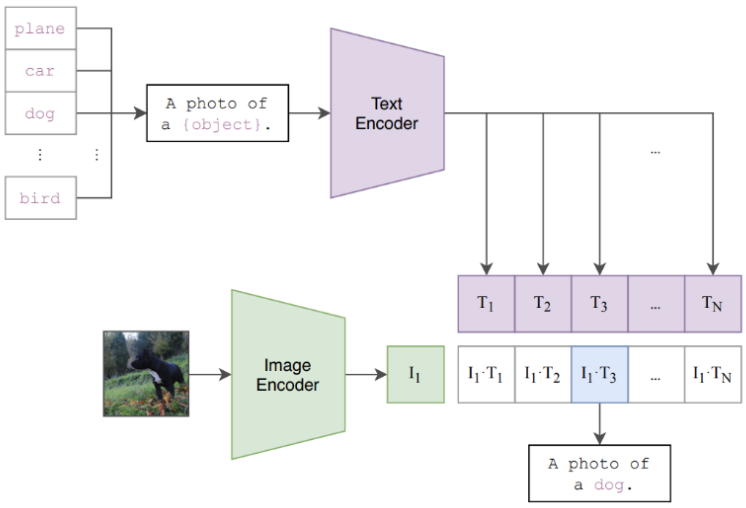

Fig. 1: **Inference using CLIP.** CLIP relies on learning a common embedding space for images and text using contrastive approaches. During the inference step, we first select a set of phrases or sentences representing the classes. The next step is to calculate the cosine similarity between the image encoding and the text embeddings. The text embedding with the largest cosine similarity is returned as the predicted class.

3.2. MAPPER

As described above, our goal is to analyse the properties of the CLIP embedding space and find prompts that have high predictive value. One approach is to visualize the space of CLIP embeddings under the lens of the predictor function. Given the high dimensionality of the embedding space, we would require a projection-based algorithm that preserves properties of the space itself. MAPPER is an algorithm proposed by Singh *et al.* [23] that perfectly suits such an application.

MAPPER allows us to construct a graph (or simplicial complex) from data while preserving some of the topological features of the space. While the algorithm only approximates topological properties, it can often be used to reveal interesting correlations through visualization. Primarily, the Mapper algorithm works by performing a local clustering guided by a projection function.

The steps are as follows:

- Project (or *map*) each datapoint to a lower dimensional space (1d or 2d for visualization) using a function of interest or the *lens* function. For machine learning, these functions can be the probability of predicting a class, difference in probabilities, feature values and others.
- The next step is to *cover* this projection with overlapping intervals/hypercubes. In theory you can use any interval shape, but most implementations support hypercubes.
- Third, we cluster the data points inside every interval indivually (either apply clustering on the projection and suffer projection loss, or cluster on the inverse image/original data). We can use any clustering algorithm (hierarchical, density-based, etc.) and distance metric or psuedometric.

Each cluster then becomes a nodes in a graph. Notice that the second step allows for a single point to be a member of many clusters due to the overlap. Teh edges of the graph correspond to these member intersections between clusters.

The primary advantage of this approach is that it preserves connectivity properties conditioned on some variable of interest. Specifically, nodes in the graph represent data points that are close in two senses– spatial distance (as used by the clustering algorithm), and functional value (as represented by the mapping function). Thus each node in the graph can be considered to be a unit of interest with similar properties. The connectivity allows for some relaxation as well as depicts the neighbourhood of our node. When carefully applied, this tool can provide interesting insights with respect to both local and global properties of our data.

For our case, we propose to leverage MAPPER to study the CLIP embedding space under the lens functions of classification margin, and the predictive confidence. We now describe our setup in detail.

4. PROMPT ENGINEERING USING MAPPER

Let us focus on the problem of prompt engineering. The approach can be trivially extended to analysing the effect of synonyms.

In order to set up the problem, let us consider a classification problem described in Sec. 3. Recall that the dataset consists of $X \times Y$ matched image-text pairs. For classification, Y consists of a list of class-names. Assume that $\mathcal{Y} = \{y_1, y_2, \ldots, y_C\}$ is the set of C classes. Radford *et al.* [1] suggests prefixing prompts to every class-name. Let

us define a set of $\mathcal{P} = \{p_1, p_2, \ldots, p_k\}$ of k prompts. We prefix every label in $\mathcal{Y}$ with every prompt in $\mathcal{P}$. This can be represented as the cartesian product of the two sets $\mathcal{P} \times \mathcal{Y}$ with Ck elements. These text elements are then input to the CLIP text encoder, $f_T(\cdot)$ to generate the text embedding set, $T = \{t_{i,j}\}$, where i refers to the $i^{\{th\}}$ class and j refers to the j^{th} prompt.

Given that the goal of prompt engineering is to find more predictive prompts, we first need to decide on a lens function that appropriately reflects this. For our purposes, we choose two lens functions: (1) *margin*, generally represented as the distance between probability of the most probable and the second most probable class, (2) the probability of the predicted class. In order to understand the effect of prompts, we consider each class individually. Note that CLIP requires both images and text embeddings to make predictions. We therefore select a representative set of d images from each class. The next step is to use CLIP to generate the logit vectors for all the images and text embeddings, and calculate the average margin for each prompt, k for an individual class i as follows:

$$\Delta_{\text{avg},l} = \frac{1}{k \cdot d} \sum_{l=1,j=1}^{k,d} , \langle f_I(\mathbf{x}_j), t_{i,l} \rangle - \max_{m \neq i} \langle f_I(\mathbf{x}_j), t_{m,l} \rangle$$

Using the $\Delta_{avg,i}$ as the lens function, we use MAPPER with agglomerative clustering over $\mathcal{P} \times \{c_i\}$ over the original text embedding space to construct Reeb graphs of the prompt space. Here each node corresponds to prompts close to each other in text embedding space, and the connections correspond to shared prompts. We also use $\Delta_{avg,l}$ to color each node with the mean average margin per node. Fig. 3 shows an example of the generated graph. In the figure, the green nodes represent high-average margin nodes, whereas blue represents low-average margin. Notice that the high and low margin nodes are disjoint. While this is an expected outcome for MAPPER, we point out that the disjointedness indicates that there are subgroups of prompts that are more predictive than others, and these are separated from the low-performing ones in the embedding space.

This informs a simple heuristic based algorithm to select the more predictive points. We select the subgraphs with higher mean average margin and drop the rest. The selected prompts should allow us to improve zero-shot performance. In the following section, we study the performance of our algorithm for the Imagenette dataset as a demonstration.

5. EXPERIMENTS

5.1. Prompt engineering for Imagenette

In order to analyse the performance of our proposed algorithm, we consider a zero shot classification task on Imagenette. Imagenette consists of ten "easy" classes from the original Imagenet dataset. For our experiments, we select 80 prompts as described in [1]. Further, we use 100 exemplar images to calculate the average margin as described above. For Mapper, we select 10 overlapping intervals with a max overlap percentage of 10%. We also use agglomerative clustering with cosine similarity and a minimum of two clusters per interval. We present a resultant Mapper graph for the 'gaspump' label in Fig. 3. Additional results can be found in the accompanying github repo. Notice that the high and low margin clusters form disjoint sub-graphs. We then drop the five simplical complexes with the lowest mean average margin per complex. The remaining prompts are collected and ensembled for each class.

We compare our approach with the two other approaches: (1) prompt ensembling as in [1], and (2) manual selection of subgraphs. We see that our approach outperforms both prompt ensembling and manual selection by $\sim$ 1.3%. See Fig. 2 for the results.

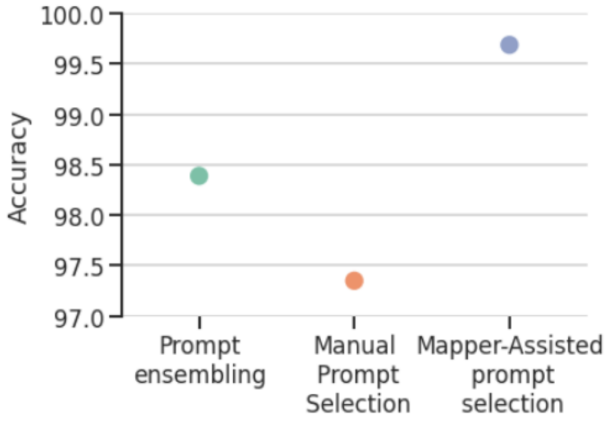

Fig. 2: **Performance of our approach.**: Mapper assisted prompt selection outperforms both prompt ensembling and manual selection by 1.3% and 2.1% respectively.

We also make two important observations from visualizing the Mapper graph for various classes.

1. The prompts with low margin are often absent from the various datapoints in the dataset. For example, the prompts "a sketch of " and "a tattoo of " are both in a single low-margin subgraph for the label 'gaspump' and these do not occur in the dataset.

2. Certain prompts like "a black and white image of " and "a picture frame of " occur in a common high-margin subgraphs throughout all classes. This leads us to hypothesize that the CLIP model appears to have learned to put semantically similar contexts close in the embedding space.

We leverage these two observations to further motivate our study of semantic changes in labels.

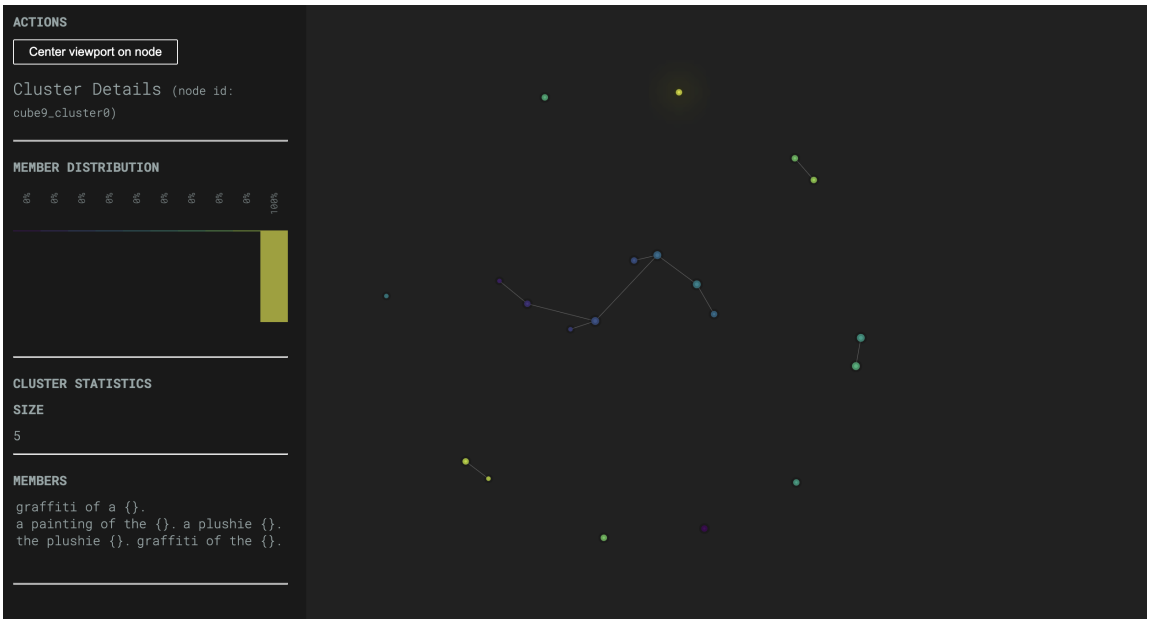

Fig. 3: **Prompt Graph generated using MAPPER**. Green nodes refer to high average margin, where as blue nodes consist of low average margin prompts. Notice that the two categories are generally disjoint, allowing us to drop low average margin prompts without noticeably affecting the performance provided by ensembling high average margin prompts. We repeat this for each class and find the top predictive prompts for every class in the dataset.

5.2. Exploring semantic changes of label space

To understand the effect of semantic alterations in label embedding space, we introduce synonyms and antonyms of the class labels. A similar approach as described above is followed, but instead of giving N exemplar images, we provide CLIP with a single image of interest with embedding vectors of prompts for each class in ImageNet dataset. In addition, we use the predictive confidence of the correct class as our *lens* function. This acts as our baseline. We then compare the performance with embedding space of synonyms as well as the antonyms for the corresponding label. Our projection function now changes to confidence or the maximum probable logit output by CLIP. This makes sense since we want to analyse whether the predictions change on providing synonyms and how similar are the mapper visualisations for the same. Fig. **??** and Fig. 6 shows the output of MAPPER algorithm.

Table 1: Graph Similarity between words and their synonyms

Word	Synonym	L1 Norm	L2 Norm	L inf Norm
chain	concatenate	14.36	5.73	3.27
junco	snowbird	16.97	10.11	7.10
buckle	wrap	16.33	7.88	5.42

We also point to the graphs in Fig. 4 and Fig. 5 as examples of MAPPER grpahs for synonyms, which indicate the effective topology of the embedding space for original and synonyms prompts are similar. On the other hand, we observe that antonyms have a stark contrast in topology (see additional results in the appendix).

Table 2: Graph Similarity between words and their antonyms

Word	Synonym	L1 Norm	L2 Norm	L inf Norm
chain	unchain	18.02	6.03	3.66
convertible	inconvertible	30.35	10.15	6.01
buckle	unbuckle	38.93	12.73	6.74

6. DISCUSSION AND CONCLUSION

We proposed a topological data analysis based approach towards analysing the embedding space for vision language models like CLIP. We leverage MAPPER to construct topology-preserving graphs of the CLIP embedding space for two tasks: prompt engineering, and understanding the effect of substituting synonyms as class-names. We show that our approach outperforms prompt ensembling and manual selection in terms of prompt engineering. We also observe that MAPPER graphs allow us to visualize the embedding space to understand the effect of synonyms and antonyms in input text. Further, we see that synonyms display similar topology with similar average predictive confidence.

Our approach shows that topological approaches can be

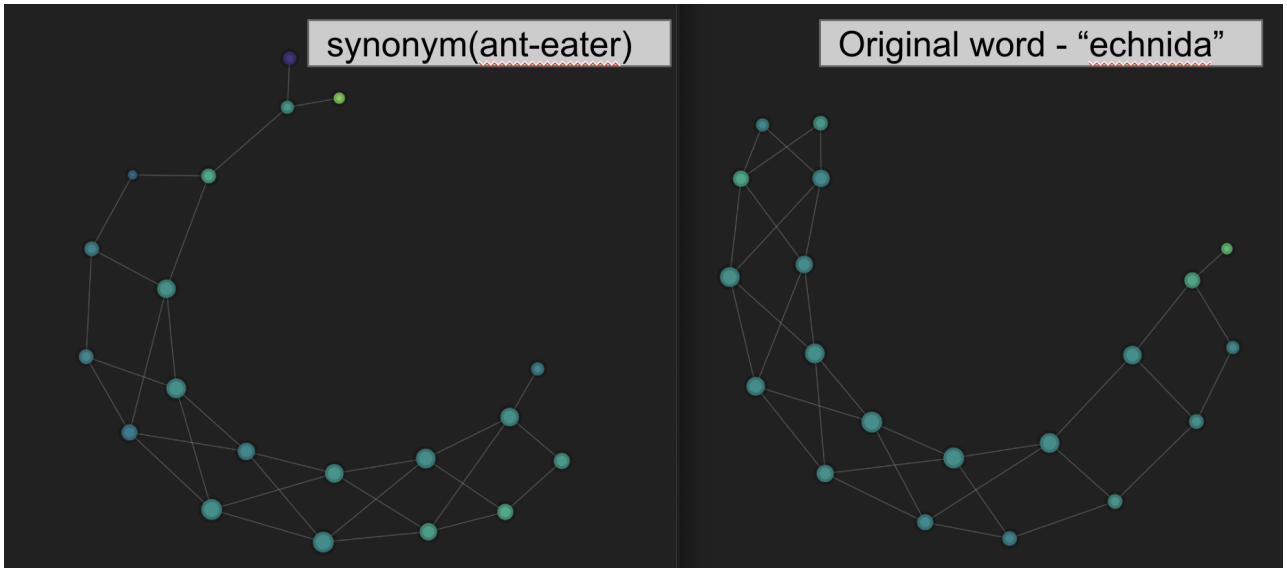

Fig. 4: **Similarity between original prompt and its synonym.**

used to analyse and understand the learned embedding space for vision language. Further, we conjecture that with different choices of lens function and clustering approaches, we might be able to quantify the limitations of vision-language models as well. Some promising future directions of interest include understanding and correcting for bias in vision-language embeddings and integrating language models to improve prompt engineering.

7. REFERENCES

[1] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever, “Learning transferable visual models from natural language supervision,” in *ICML*, 2021.

[2] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig, “Scaling up visual and vision-language representation learning with noisy text supervision,” in *ICML*, 2021.

[3] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[4] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt, “Measuring robustness to natural distribution shifts in image classification,” *ArXiv*, vol. abs/2007.00644, 2020.

[5] John Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt, “Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization,” *ArXiv*, vol. abs/2107.04649, 2021.

[6] Shafie Gholizadeh, *Topological Data Analysis in Text Processing*, Ph.D. thesis, 2020.

[7] Ketki Savle, Wlodek Zadrozny, and Minwoo Lee, “Topological data analysis for discourse semantics?,” in *Proceedings of the 13th International Conference on Computational Semantics-Student Papers*, 2019, pp. 34–43.

[8] Ramya Srinivasan and Ajay Chander, “Understanding bias in datasets using topological data analysis,” in *AISafety@ IJCAI*, 2019.

[9] Laida Kushnareva, Daniil Cherniavskii, Vladislav Mikhailov, E. Artemova, S. Barannikov, Alexander V. Bernstein, Irina Piontkovskaya, D. Piontkovski, and Evgeny Burnaev, “Artificial text detection via examining the topology of attention maps,” in *EMNLP*, 2021.

[10] Peter Xenopoulos, Gromit Yeuk-Yin Chan, Harish Doraiswamy, Luis Gustavo Nonato, Brian Barr, and Cláudio T. Silva, “Topological representations of local explanations,” *ArXiv*, vol. abs/2201.02155, 2022.

[11] Peter Bubenik et al., “Statistical topological data analysis using persistence landscapes.,” *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 77–102, 2015.

[12] Firas A Khasawneh, Elizabeth Munch, and Jose A Perea, "Chatter classification in turning using machine learning and topological data analysis," *IFAC-PapersOnLine*, vol. 51, no. 14, pp. 195–200, 2018.

[13] Grzegorz Muszynski, Karthik Kashinath, Vitaliy Kurlin, Michael Wehner, et al., "Topological data analysis and machine learning for recognizing atmospheric river patterns in large climate datasets," *Geoscientific Model Development*, vol. 12, no. 2, pp. 613–628, 2019.

[14] Yuhei Umeda, "Time series classification via topological data analysis," *Information and Media Technologies*, vol. 12, pp. 228–239, 2017.

[15] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 2000, pp. 454–463.

[16] Afra Zomorodian and Gunnar Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[17] Gurjeet Singh, Facundo Mémoli, Gunnar E Carlsson, et al., "Topological methods for the analysis of high dimensional data sets and 3d object recognition.," *PBG@ Eurographics*, vol. 2, 2007.

[18] Monica Nicolau, Arnold J Levine, and Gunnar Carlsson, "Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival," *Proceedings of the National Academy of Sciences*, vol. 108, no. 17, pp. 7265–7270, 2011.

[19] Ann E Sizemore, Jennifer E Phillips-Cremins, Robert Ghrist, and Danielle S Bassett, "The importance of the whole: topological data analysis for the network neuroscientist," *Network Neuroscience*, vol. 3, no. 3, pp. 656–673, 2019.

[20] Gunnar Carlsson, "Topological pattern recognition for point cloud data," *Acta Numerica*, vol. 23, pp. 289–368, 2014.

[21] Xiangyu Zhang, Kexin Zhang, and Yongjin Lee, "Machine learning enabled tailor-made design of application-specific metal–organic frameworks," *ACS Applied Materials & Interfaces*, vol. 12, no. 1, pp. 734–743, 2019.

[22] Hugo J Bello, Nora Palomar, Elisa Gallego, Lourdes Jiménez Navascués, and Celia Lozano, "Machine learning to study the impact of gender-based violence in the news media," *arXiv preprint arXiv:2012.07490*, 2020.

[23] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson, "Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition," in *Symposium on Point Based Graphics*, 2007.

A. ADDITIONAL RESULTS

A.1. MAPPER graphs for synonyms

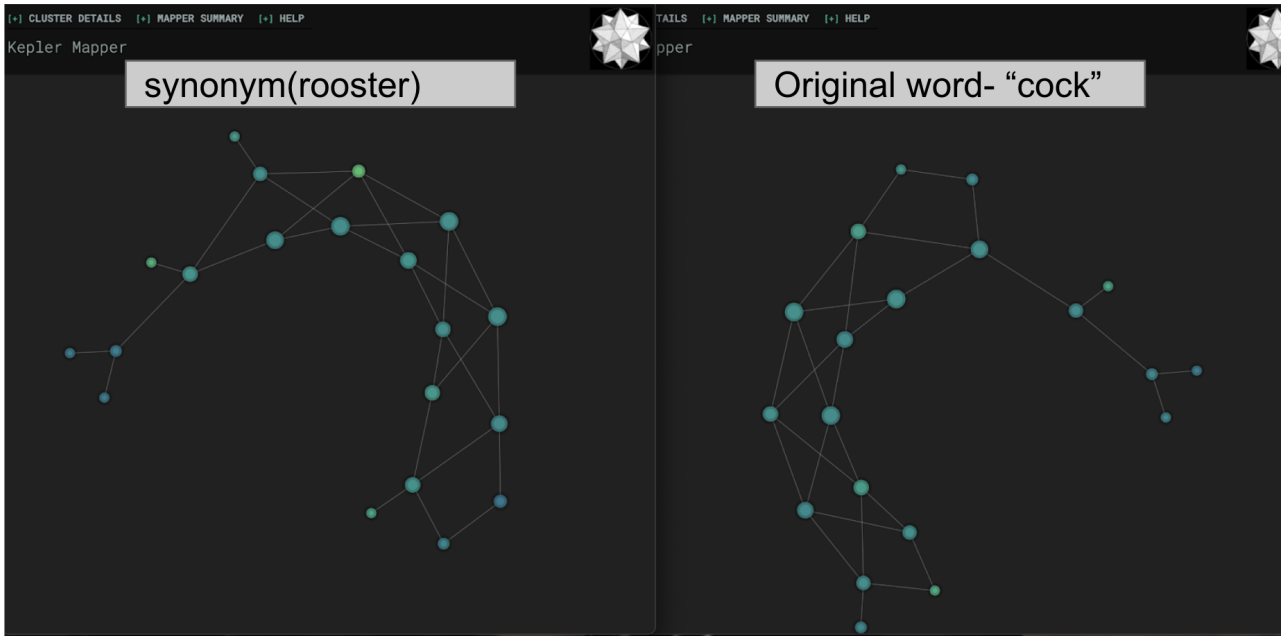

Fig. 5: **Similarity between original prompt and its synonym.**

A.2. MAPPER graphs for antonyms

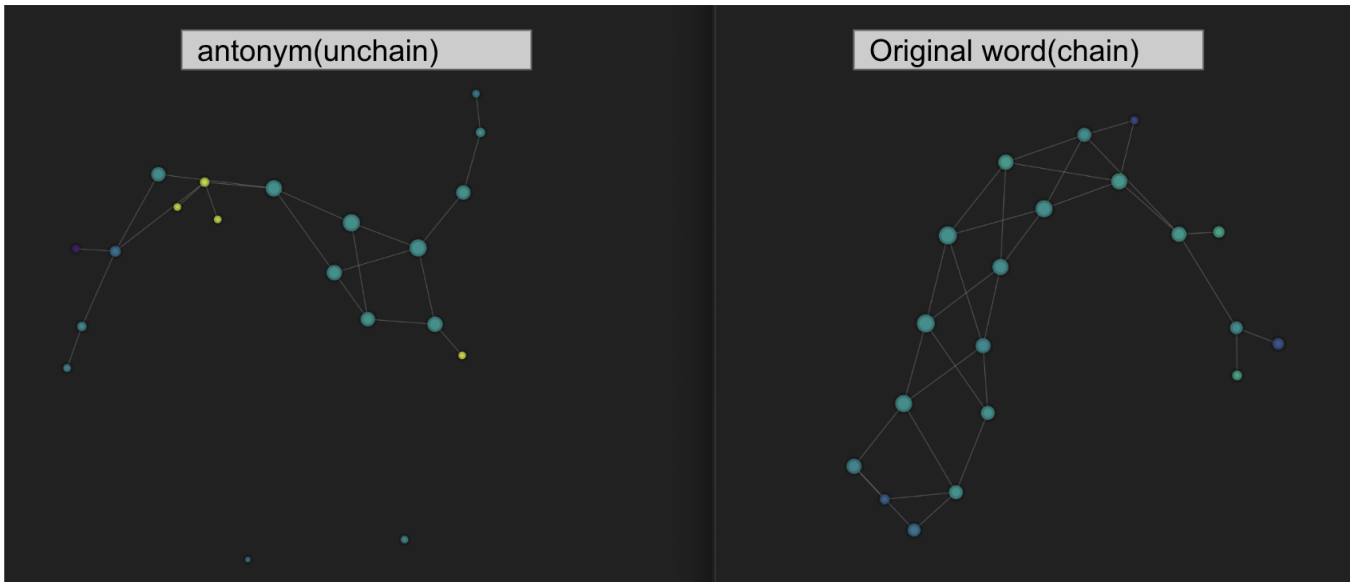

Fig. 6: **Visual Contrast between original prompt and its antonym.**

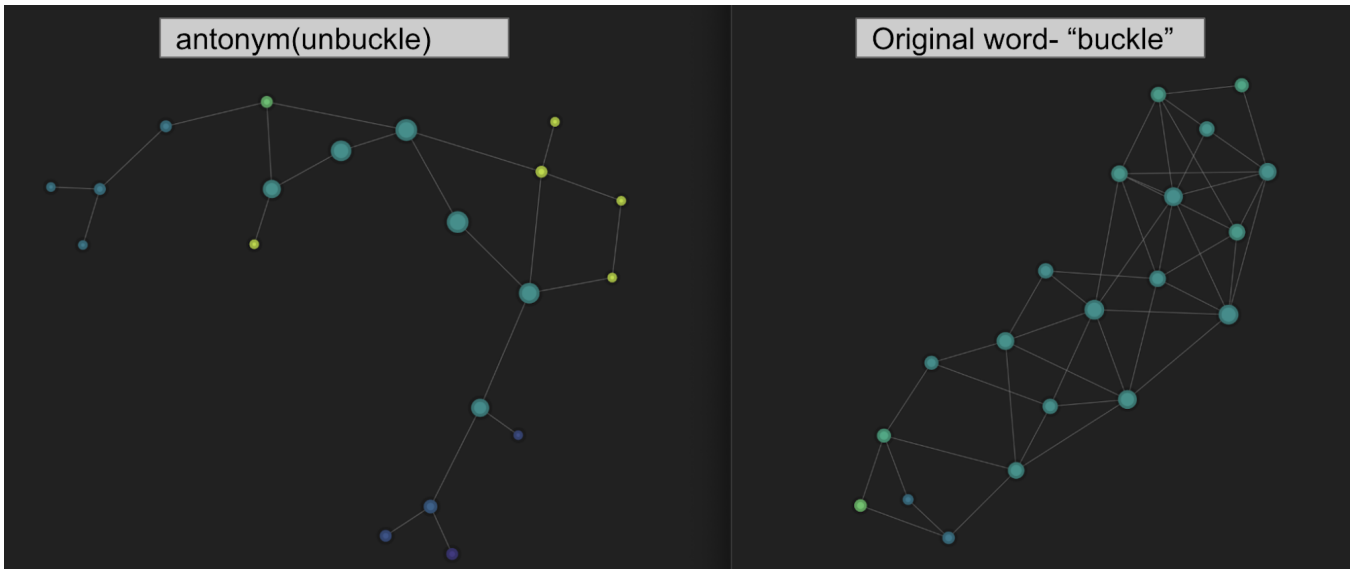

Fig. 7: **Visual Contrast between original prompt and its antonym.**