

Harnessing natural

language processing for

in-depth analysis of national

park survey responses

Austin Lackey, VU Stats Intern

Table of Contents

[Table of Figures 2](#_Toc143072626)

[Introduction 4](#_Toc143072627)

[Socioeconomic Monitoring (SEM) of NPS Visitors 4](#_Toc143072628)

[Question Examples 4](#_Toc143072629)

[Survey Data 4](#_Toc143072630)

[Data Types 4](#_Toc143072631)

[Structuring the Unstructured 5](#_Toc143072632)

[Project Goal 5](#_Toc143072633)

[Issues with free-form text 5](#_Toc143072634)

[Harnessing Natural Language Processing (NLP) 5](#_Toc143072635)

[Methodology 6](#_Toc143072636)

[Code Reference and Reproducibility 6](#_Toc143072637)

[Model Creation 6](#_Toc143072638)

[Model Initialization 6](#_Toc143072639)

[Tokenization 6](#_Toc143072640)

[Gensim 6](#_Toc143072641)

[Vectorization 7](#_Toc143072642)

[Dimension Reduction 7](#_Toc143072643)

[Issues with high dimensional data 7](#_Toc143072644)

[Dimension Reduction Algorithms 8](#_Toc143072645)

[Why is this powerful? 9](#_Toc143072646)

[Data Application 10](#_Toc143072647)

[Clustering 10](#_Toc143072648)

[Visualization and Analysis 12](#_Toc143072649)

[Conclusion 14](#_Toc143072650)

[Model Overview 14](#_Toc143072651)

[References 15](#_Toc143072652)

# Table of Figures

[Figure 1: Sentence To Vector 5](#_Toc143072615)

[Figure 2: Tokenization Process 6](#_Toc143072616)

[Figure 3: Vectorization Model 7](#_Toc143072617)

[Figure 4: Parallel Coordinates Plot 8](#_Toc143072618)

[Figure 5: Swiss Role with Hole in 2D (Feature Extraction) (ISOMAP) 9](https://doimspp-my.sharepoint.com/personal/alackey_nps_gov/Documents/Desktop/NPS%20Internship/SEM%20Visual%20Dashboard/Writeup%20Files/PARSING%20THE%20PARKGOER%20PERSPECTIVE.docx#_Toc143072619)

[Figure 6: Swiss Role with Hole in 3D (Euclidean Space) 9](https://doimspp-my.sharepoint.com/personal/alackey_nps_gov/Documents/Desktop/NPS%20Internship/SEM%20Visual%20Dashboard/Writeup%20Files/PARSING%20THE%20PARKGOER%20PERSPECTIVE.docx#_Toc143072620)

[Figure 8: Swiss Role with Hole in 2D (Feature Selection) (X/Y) 9](https://doimspp-my.sharepoint.com/personal/alackey_nps_gov/Documents/Desktop/NPS%20Internship/SEM%20Visual%20Dashboard/Writeup%20Files/PARSING%20THE%20PARKGOER%20PERSPECTIVE.docx#_Toc143072621)

[Figure 9: TSNE Dimension Reduction 10](#_Toc143072622)

[Figure 10: KMEANS Clustering 11](#_Toc143072623)

[Figure 11: Vector Topology 12](https://doimspp-my.sharepoint.com/personal/alackey_nps_gov/Documents/Desktop/NPS%20Internship/SEM%20Visual%20Dashboard/Writeup%20Files/PARSING%20THE%20PARKGOER%20PERSPECTIVE.docx#_Toc143072624)

[Figure 12: Full Model 14](#_Toc143072625)

# Introduction

## Socioeconomic Monitoring (SEM) of NPS Visitors

In 2023, the National Park Service released their inaugural SEM National Technical Report which gave a comprehensive insight into park visitation for 2022 across the entire service. The report was designed with the intent to understand how visitors utilize units within the National Park system. The information gathered can then be used to make decisions in improving future facilities and services. Each year the National Park Service will conduct surveys at 24 different parks who are drawn from a stratified random sample. These strata ensure accurate representation of parks across different categories such as Natural/Recreation types as well as High/Low visitation rates.

Park visitors were intercepted onsite at each park for a brief survey and then asked to complete a more comprehensive survey once they returned to their residence. Between both surveys, visitors were asked questions that pertained to their trip spending, travel patterns as well as their own perceptions of their recent visit to the park. Below are a few examples of some of the questions that were asked.

### Question Examples

* *“From the list below, select all forms of transportation you personally used to travel from your home to [NPS Site] on this trip.”*
* *“What did you like the most about your visit to [NPS Site]?”*
* *“This park was established because of its significance to the nation. In your opinion, what is that national significance of this park?”*

## Survey Data

While specific questions varied from park to park, the combined dataset for survey results contained 333 different variables that stemmed from survey questions. Visitors were not required to answer all questions which led to a dataset with sparse entries. All survey questions can be placed into the following datatype bins…

### Data Types

* Range Selection (e.g., 1-2 hours, 3-4 hours…)
* Binary Selection (e.g., Yes/No)
* Integer Input (e.g., Age: 25)
* Open-ended text (e.g., Comments: “Great scenery!”)
* Rating (e.g., 1 – Poor, 5 – Very good)
* Selection from group (e.g., Primary Language: English, Spanish…)

The National Technical Report did a great job at visualizing the quantitative results; however, these results did not encompass any information on the open-ended text responses. The objective of my project was not to replicate the National Technical Report, but to expand analysis on the untapped textual data.

# Structuring the Unstructured

## Project Goal

The goal of this project was to come up with a way to find patterns in textual responses and cluster those responses by meaning. Below are some examples of some questions we may be interested in.

* Of the visitors who wanted more cell coverage, what was their age demographic?
* Were the individuals who mentioned traffic as their primary concern residing in close proximity to the park, or did they commute from a distance?
* What is the age demographic of people who enjoyed the historical aspects of the park.

## Issues with free-form text

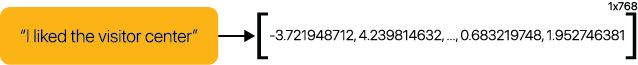
Free-form textual data by nature is inconsistent and very complex. Basic data analysis techniques are effective for numerical and categorical data; however, they contain many limitations on robustness when applied to open-ended textual responses. For example, sentences *“The scenery was wonderful”*, and *“The landscape was amazing”* have the same meaning but are vastly different from one another when it comes to words and characters. For analysis, it would require a human to read through each response and assign a code/category. This technique is very time consuming and even if the data were limited to a few hundred samples, it is very difficult to stray away from the subjectivity of the human analyst.

## Harnessing Natural Language Processing (NLP)

Natural Language Processing (NLP) is a branch of artificial intelligence that allows computers to understand the patterns and connections that lie within human language. This means that models can analyze human created text with efficiency, accuracy, and consistency. NLP models are created on the backbone of Neural Networks which allows them to learn and understand the intricacies of sentence structure and semantic meaning. NLP models achieve this by converting words into a long array of decimal numbers called vectors. Transformers are a type of NLP model that use attention heads to properly assign weight biases to words within a sentence to accurately denote the overall meaning. These vectors can then be used for numerical analysis akin to how quantitative data would be treated. This process is known as creating word embeddings.

In Figure 1: Sentence to Vector, we can see an example of how a transformer model would use word embedding to convert a statement *“I liked the visitor center”* into a vector of length 768. This vector is a numerical description of the statements meaning and can now be compared with vectors generated by other statements.

Figure : Sentence to Vector



# Methodology

## Code Reference and Reproducibility

The Jupyter notebook and Python code can be referenced using this [**GitHub link**](https://github.com/austinlackey/sem_nlp). Before each major model operation, seeds were reset to a constant value in order to maintain reproducibility and consistent results. All open-source packages that were used for this project are listed at the top of the ‘transformer.py’ document within the GitHub repository.

## Model Creation

### Model Initialization

Information and data for all models were stored in python classes, this ensures that all models are uniform and allows for multiple models to be created while maintaining a concise code structure. Information regarding the text responses, model name and the dimension reduction/cluster algorithms are stored in this class to allow for a streamlined model comparison process in the future.

### Tokenization

The vectorization process requires that all responses are first tokenized. The tokenizing process splits each response into a list of words or in this case “tokens”. This allows each word to be fed independently into the transformer so that it can properly assign each word with an attention score. Attention scoring is a crucial step in the vectorization process as it determines the importance and relevance of each word within the context of the entire response. During attention scoring, the transformer assigns weights to different tokens based on their significance in capturing the meaning of the input. Attention scores facilitate the capturing of dependencies and relationships between words, allowing the model to grasp nuances and subtle meanings in the language.

Figure : Tokenization and Attention Scoring Process



By using these attention scores in conjunction with mean pooling the model can generate a more refined representation of the input sequence. Mean pooling involves calculating the average of the token embeddings weighted by their respective attention scores. In this specific example, tokens ‘liked’, ‘visitor’, and ‘center’, which received higher attention scores, will contribute more significantly to the pooled representation. This approach ensures that the most relevant information is captured while minimizing the impact of less important tokens. This process creates a robust foundation for tasks like sentiment analysis in which we will use later.

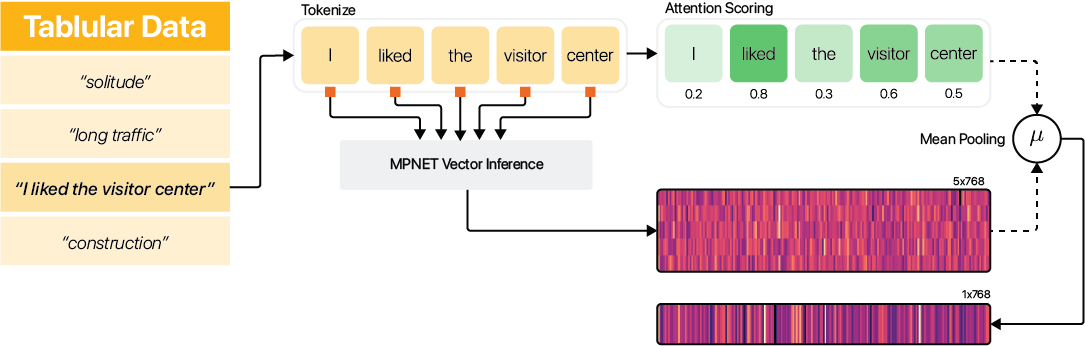
### Hugging Face

### Hugging Face is a prominent platform and company in the field of NLP that offers a diverse set of tools and resources for working with state-of-the-art models. It serves as a hub for accessing and sharing pre-trained models, datasets, and various NLP-related components. Hugging Face is also recognized for its Transformers library, which provides an easy-to-use interface for leveraging transformer-based models. The model used in this project is **Hugging Face’s (sentence-transformers/all-mpnet-base-v2)** which is a fine-tuned version of **Microsoft’s (microsoft/mpnet-base)** model. This particular model excels at sentence comparison which is perfect for comparing park survey response data. You can find more information regarding this model [here](https://huggingface.co/sentence-transformers/all-mpnet-base-v2).

### Vectorization

Vectorization is the process of turning a list of tokens into a single complex vector that captures its overall meaning. To obtain a model is robust enough, a neural network must be fed with enough data that it can learn how a given word interacts with other words in thousands of different situations. The process demands lots of training data which requires extensive time and computing power. This task is not possible with most personal computers and requires high-end GPU’s and computers. Luckily, there are open-source libraries online that contain large models for public consumption. The model used in this project was trained on primarily Reddit comments, and Question/Answer data and maps sentences in a 768-dimensional dense vector space.

Figure : Vectorization Model



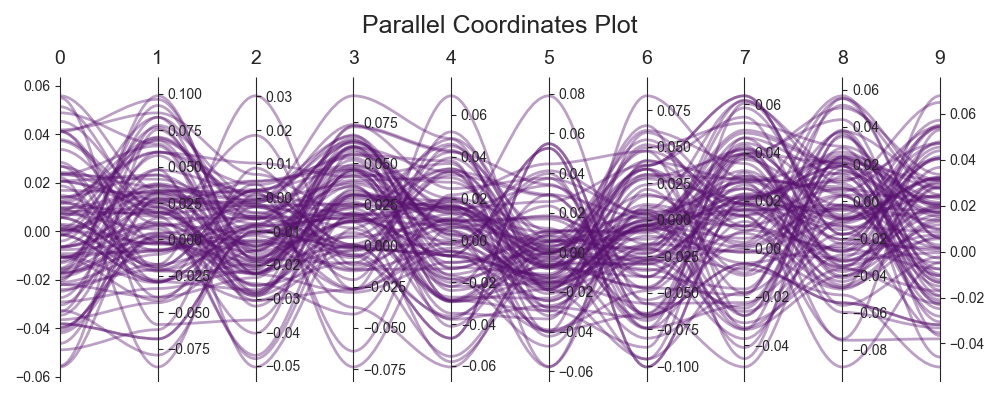
Each word from within a response is fed into the model and outputs a corresponding list of vectors that represents meaning of those words. In order to represent the words as a collective whole, the average of all the vectors within each sentence is taken. We are now left with a single 1x768 vector to represent the sentient behind that phrase or statement.

## Dimension Reduction

### Issues with high dimensional data

Dimension Reduction is an important aspect in dealing with high dimensional data. 1x768 vectors are great since they capture a large amount of detail, however it can be impossible to interpret and use for analysis. Think of a datapoint in a 768-dimensional space; it is impossible for humans to conceptualize what that datapoint means and how it relates to other points in the same space.

Figure : Parallel Coordinates Plot



Take for example in Figure 4: Parallel Coordinates Plot, we are visualizing the first 10 dimensions of almost 100 different vectors. While we can see how the numbers for each dimension vary, we do not get an understanding of how each vector relates to one another. Now if we extrapolate this idea to all 768 dimensions, the interpretation issue only becomes greater.

This is referred to as the “Curse of dimensionality”, and a [user on Quora](https://www.quora.com/What-is-the-curse-of-dimensionality) has an excellent analogy that captures the problems associated with adding more dimensions.

*“Let's say you have a straight line 100 yards long and you dropped a penny somewhere on it. It wouldn't be too hard to find. You walk along the line and it takes two minutes.*

*Now let's say you have a square 100 yards on each side and you dropped a penny somewhere on it. It would be pretty hard, like searching across two football fields stuck together. It could take days.*

*Now a cube 100 yards across. That's like searching a 30-story building the size of a football stadium. Ugh*

To enhance the comprehension of the interactions between each vector we need to reduce the number of dimensions to something that consumable and computationally efficient.

### Dimension Reduction Algorithms

Dimension Reduction Algorithms are useful tools when it comes to extracting important information from datasets. While we could take the first two or three components from our vectors; these features are unlikely to be optimal candidates in terms of information retention. This is where unsupervised learning algorithms such as **Principal Component Analysis (PCA)** or **Isometric Mapping (ISOMAP)** begin to excel. Rather than selecting the two dimensions that capture the most information (Feature Selection), PCA and ISOMAP generate entirely new dimensions that retain as much geodesic information as possible (Feature Extraction).

### Why is this powerful?

To visualize how powerful feature extraction algorithms like ISOMAP are, a synthetic dataset that resembles a three-dimensional Swiss roll can be seen below to show how Feature Selection results in data-loss. Both representations that stem from Feature Selection resulted in a loss of information in some way. Whether that be the absence of the center hole (X/Z), or the gradual color transition from one end to the other (X/Y). ISOMAP on the other hand, creates two new dimensions that can retain most of the three-dimensional geometry while residing in a two-dimensional space. This process only becomes more significant and useful as the number of original dimensions becomes larger.

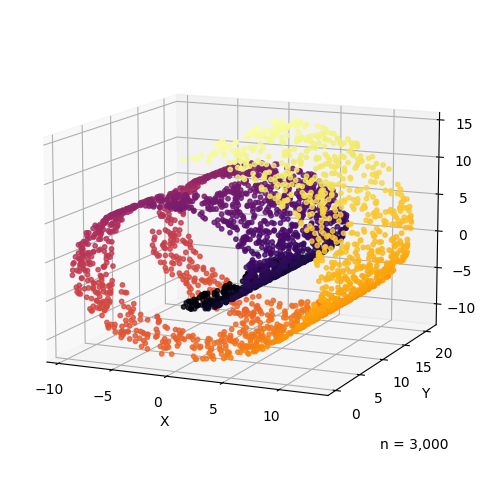


Figure : Swiss Role with Hole in 2D (Feature Extraction) (ISOMAP)

Figure : Swiss Role with Hole in 3D (Euclidean Space)

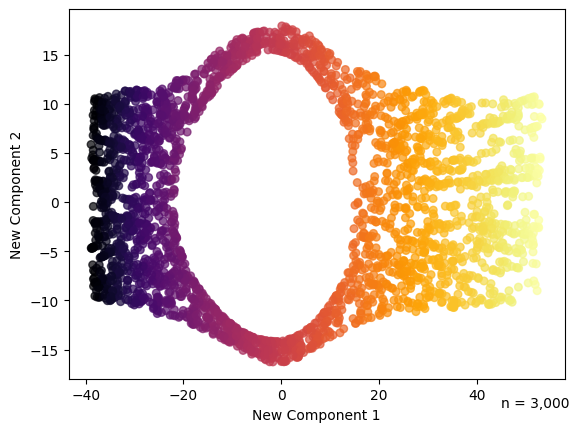


Figure 7: Swiss Role with Hole in 2D (Feature Selection) (X/Z)

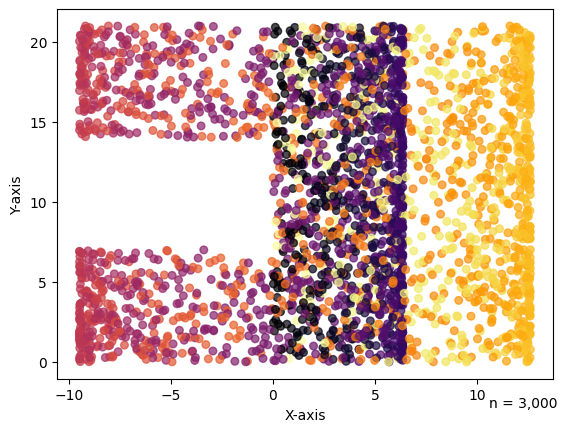
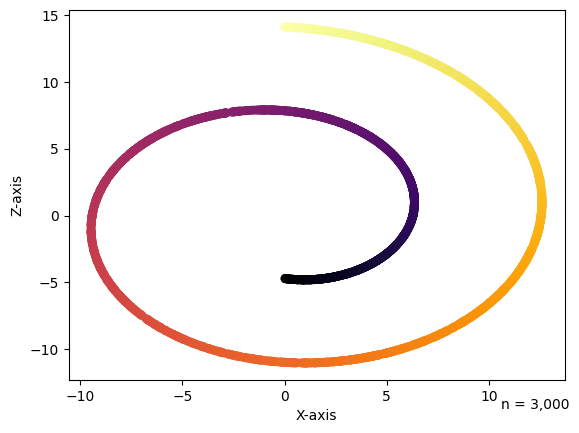


Figure : Swiss Role with Hole in 2D (Feature Selection) (X/Y)

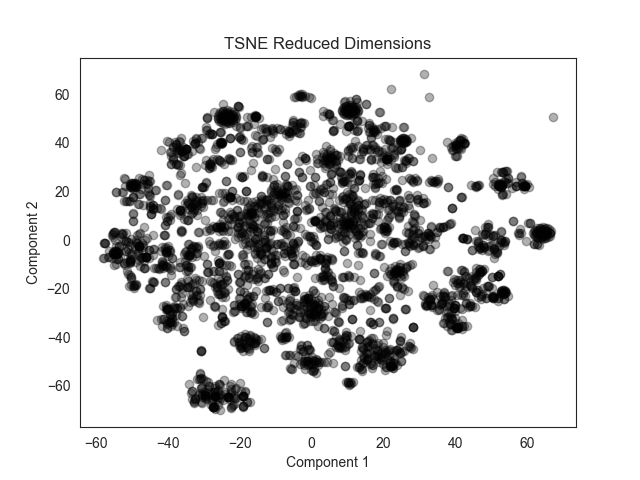


It is important to understand why dimension reduction algorithms are so important when it comes to multi-dimensional spaces. When we are visualizing the Swiss roll in a Euclidean space, the distances between the yellow and purple points are far smaller than the distances between the yellow and orange points. However, we know that if we unroll the Swiss roll into a flat strip, the yellow points are actually much closer to the orange points. This is what we call the ‘geodesic’ geometry, and it ISOMAP is a way of capturing Geodesic distances rather than Euclidean distances.

### Data Application

There are many Dimension Reduction Algorithms out there to use, I have found that **‘T-distributed Stochastic Neighbor Embedding’ (TSNE)** works the best with survey response data. Displayed below is the TSNE representation of the 768-dimensional vectors. We have now reduced our 768-dimensional data down to two-dimensions which enables it to become extremely consumable. Every response is linked to a two-dimensional data point that captures its meaning in relation to the surrounding points. From this depiction we can infer that points closer to one another exhibit similarity with one another and points on opposite ends become far different in terms of meaning.

Figure : TSNE Dimension Reduction



## Clustering

With data points now arranged in a meaning-based and interpretable space we can utilize clustering algorithms to categorize similar responses together. There are many clustering algorithms available, however k-means has yielded the best results with survey data. Data used for these visualizations are in response to the question ***“What did you like least about your visit to [NPS Site]?”*.**

Figure : KMEANS Clustering



When applying k-means to our newly reduced data, we can clearly see groups of similar meaning begin to form. Labels are randomly overlayed on top of the datapoints with the first character being positioned directly above its corresponding point. It is important to note that not all clusters exhibit the same meaning; their formation relies on proximity to other points. Certain clusters in the center are merely just a melting pot of diverse responses that lack sufficient prominence to be grouped together with homogeneity. However, groups that are located along the perimeter generally have similarities with meaning. Presented below is a table of clusters that have formed with a constant level of similarity within their respective samples.

|  |  |  |
| --- | --- | --- |
| **Cluster #** | **Overall Group Meaning** | **Sample Size** |
| 9 | Crowding | 161 |
| 11 | Nothing to dislike | 140 |
| 13 | Lack of cell service / No WIFI | 135 |
| 7 | Mosquitos / Bugs | 76 |
| 5 | Parking related issues | 138 |

## Visualization and Analysis

To come full circle, visualizing the vectors of two distinct clusters can help us understand how the vectors within each cluster are similar and how those vectors differ with other clusters. Below is a visual depicting the geometry for each vector when comparing 50 responses from both Cluster 6 and 15.

The x-axis represents each statements corresponding vector, while the y-axis denotes the indices of numbers within these vectors. represents the first number within the 1x768 vector and represents the final number in the sequence. The z-axis represents individual numerical values; regions of the mesh that are elevated and lighter in color (Yellow) represent larger values, whereas regions that are lower and darker (Black) indicate lower values.

|  |  |  |
| --- | --- | --- |
| **Vector** | **Cluster** | **Statement** |
|  | **C07** | “The bugs” |
|  | **C07** | “mosquitos” |
|  | **C13** | “GPS service lost” |
|  | **C13** | “bad cell service when really needed” |

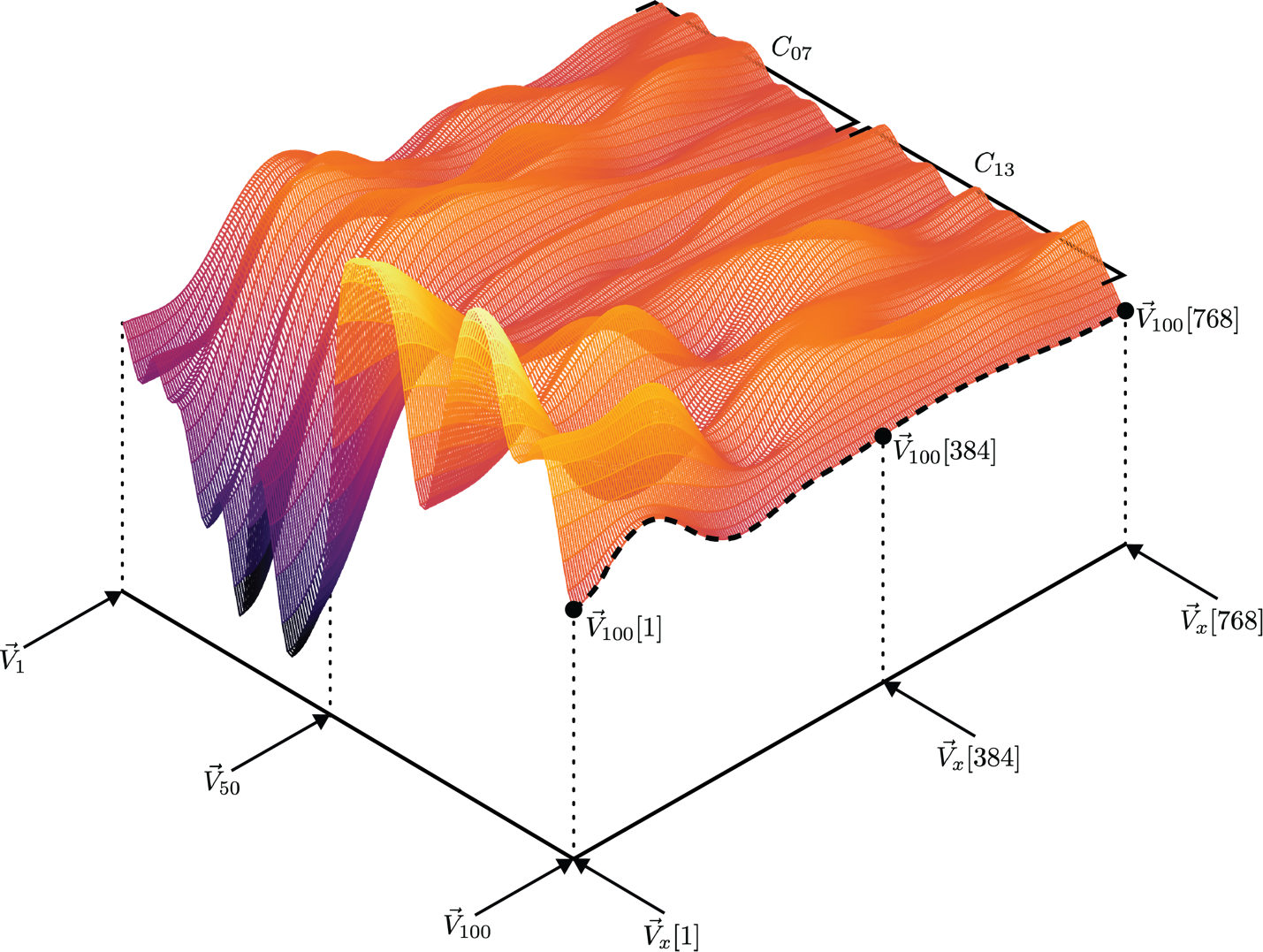


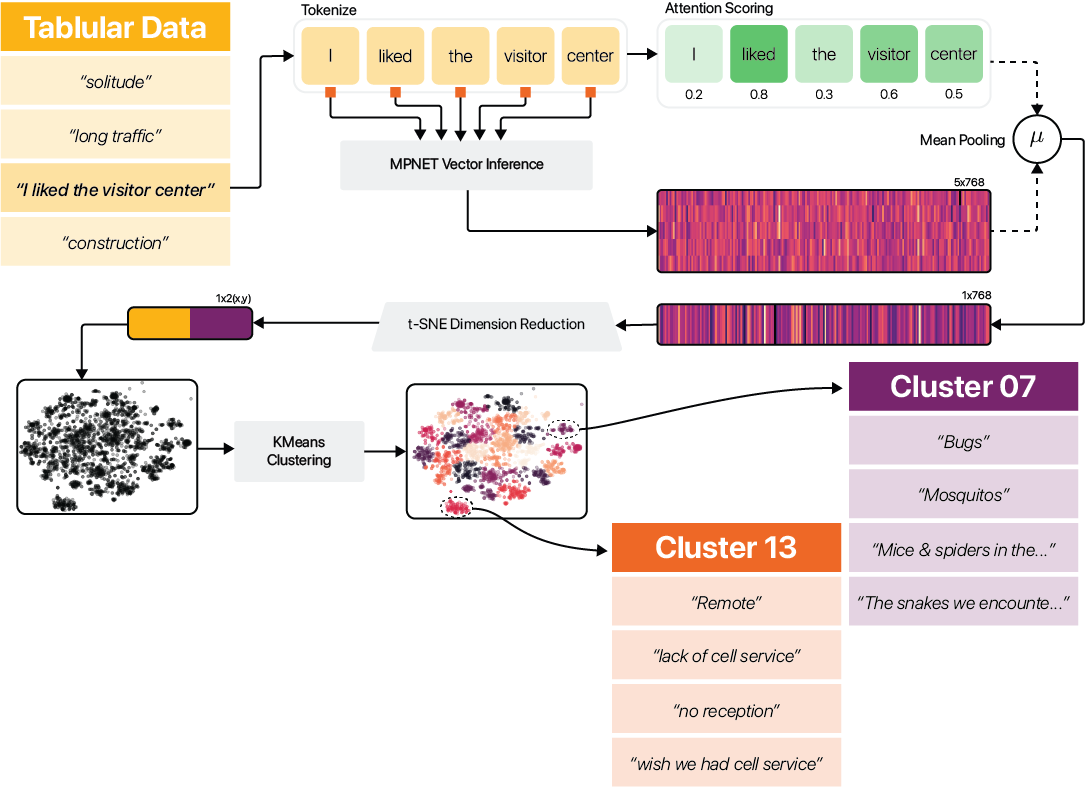
Figure : Vector Topology

We can see that vectors related to bugs begin and end with larger values when compared with vectors that are related to bad cell coverage. We can clearly see that vectors within each cluster are very similar; and vectors across each cluster are very different. However, there is an interesting characteristic that is present in both clusters that is worth noting; around the middle at there is an extreme dip in the values. This could be the nature of how survey data is understood by the model’s neural network or how the neural network deals with splitting up word information.

# Results

## Model Overview

Figure : Full Model



## Cluster Demographics

<Expanding on this later>

# References

**There are no sources in the current document.**