Clustering Neuron Explanations in Large Language Models

Yahav Ben Yaakov
Tel Aviv University
yahavb@mail.tau.ac.il

Oz Zafar
Tel Aviv University
ozzafar@mail.tau.ac.il

Mikey Shechter
Tel Aviv University
sachter@mail.tau.ac.il

Abstract

This article introduces a research centered around OpenAI's dataset as outlined in one of their latest publications, "Language Models Can Explain Neurons in Language Models" (Bills et al., 2023). The project focuses on applying various clustering algorithms and topic modeling to explore meaningful patterns inherent in the explanations of neurons within language models. Specifically, we investigate whether there exists a tendency for neuron explanations from the same (or nearby) layer to be grouped together and potentially uncover insights into language model behavior. The source code used to conduct the experiments in this paper is available at: https://github.com/MikeyShechter/nlp_project.git.

1 Introduction

Language models have achieved remarkable success in natural language understanding and generation tasks. However, understanding how these models work at the neuron level remains a challenging problem. OpenAI's paper, "Language Models Can Explain Neurons in Language Models" (Bills et al., 2023) provides a valuable dataset consisting of explanations for each of the 307,200 neurons in GPT-2 XL, with corresponding explanation scores (reflecting confidence levels) and additional information.

In this project, we aim to explore this dataset by focusing on the explanations and their associated scores. Specifically, we are interested in determining whether neurons from nearby layers are correlated, suggesting that neighboring layers have similar responsibilities within the language model. Our intuition for this behavior comes from the way layers act in a convolutional neural network, where lower layers are responsible for detecting simple features such as edges, while deeper layers are responsible for recognizing more complex patterns. To address this objective, we adopt several clustering methodologies.

2 Settings

2.1 Dataset

We utilize the publicly available dataset [1] from OpenAI's paper, which includes explanations for each of the 307,200 neurons of 48 different layers in GPT-2 XL. Each explanation is associated with a score, determining how well the explanation fits the neuron. The dataset includes additional details such as related neurons, tokens with high average activations, and more. The dataset was created by using GPT-4. For more information about the complete dataset and its creation process, check out OpenAI's paper (Bills et al., 2023). In this article, we will focus on the explanations and their scores. Below is a sample data row for demonstration.

Layer	Neuron #	Score	Explanation
47	6367	0.304	names of organizations, teams, and locations.

2.2 Neuron Selection: Filtering Low Score Explanations

Some of the explanations have low scores, which makes them less reliable for analysis. Therefore, we decided to filter some of the lowest scoring explanations when analyzing the dataset.

While setting a fixed threshold (e.g. 0.5) and filtering explanations with scores below it may sound reasonable, it would result in keeping much more lower layers explanations compared to deeper layers, because neuron explanations in deeper layers

[1] https://github.com/openai/automated-interpretability

consistently exhibit lower scores in this dataset. For instance, in layer 40, the median score falls below 0.2, while in layer 0, the median hovers around 0.4. This distribution is visually represented in Figure 1. Therefore, we decided to filter the lowest scoring neurons from each layer separately. We analyzed several score percentiles: 0% (indicating no filter), 50% and 90%. For each percentile, only neurons with explanation score higher than the value of the percentile in their respective layer are included in the analysis. Throughout the paper, we use the term 'filtering percentile' to denote this concept. This approach allows us to focus on neurons that exhibit higher levels of confidence and potentially observe distinct behaviors, while avoiding bias towards specific layers.

2.3 Clustering

2.3.1 Embeddings

To preprocess the data for clustering, we initiate the process by generating embeddings for all neuron explanations. Utilizing the sentence-transformer model all-mpnet-base-v2[2], each explanation is transformed into a vector representation, encapsulating its semantic meaning.

This approach enables the application of clustering algorithms such as K-Means and DBSCAN (Ester et al., 1996). Furthermore, we employed BERTopic, which internally uses a sentence transformer as well to generate its own embeddings for the clustering process.

2.3.2 Clustering Methods

We applied two clustering algorithms to the generated embeddings: K-Means and DBSCAN. For K-Means, we set K=48, corresponding to the number of layers. In the case of DBSCAN, the algorithm automatically determined the number of clusters. However, a challenge arose with DBSCAN, as many embeddings were labeled with '-1', signifying the algorithm's inability to cluster them. To address this, we tried several approaches, such as fine-tuning clustering parameters to improve its clustering performance, filtering out unclustered embeddings, and assigning unclustered embeddings to the cluster with the nearest center. Notably, across all approaches we consistently obtained similar results in our experiments. For simplicity and consistency, the presented findings are based on the last option.

We also applied topic modeling using BERTopic (Grootendorst, 2022), which utilizes HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise), an extension of DBSCAN. HDBSCAN transforms DBSCAN into a hierarchical clustering algorithm. Within this hierarchy, a technique is applied to extract a flat clustering based on the stability of clusters, enhancing the precision and reliability of the topic modeling results.

In addition, we applied random clustering with 48 clusters to provide a baseline for comparison and to evaluate the performance of the previously mentioned algorithms. Random clustering simply means assigning a random index between 0 to 47 to each explanation.

3 Experiments

In the following section we present our experiments and analysis, which examine the clustering results from various perspectives. Our main goal is to understand how well explanations from the same layer (or close by layers) are correlated to each other by assessing how effectively different clustering algorithms group them together. We used a few methods to understand this correlation between neurons which we will go into in this section.

3.1 Experiment 1: Layer's Distribution Across Clusters

Initially, our focus was on understanding the concentration of explanations from the same layer across clusters. In simpler terms, we wanted to figure out whether a layer's explanations were mostly grouped in just one or few clusters (indicating high concentration) or if they were spread out across many clusters without a substantial presence in any particular cluster (indicating low concentration).

To assess that, we calculated the Gini coefficient (Gini, 1912) for all the layers. For each layer, we counted the number of explanations from that layer in each cluster. Then, we used these counts as input for calculating the Gini coefficient. The Gini coefficient measures the inequality among the values of a

[2] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

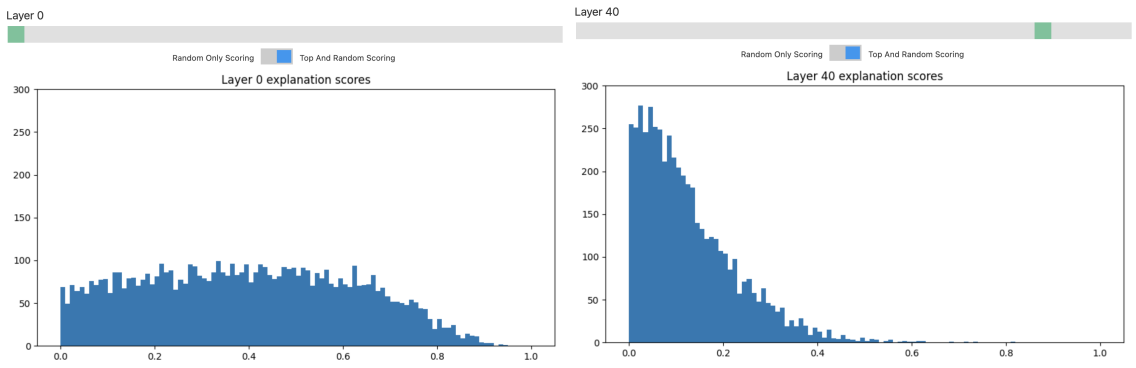

Figure 1: Explanations scores histogram of layers 0 (left) and 40 (right). Referenced from (Bills et al., 2023)

frequency distribution. A Gini coefficient of 0 means that there's perfect equality, indicating that all the clusters received the same number of explanations from that layer. On the other hand, a Gini coefficient of 1 signifies the highest level of inequality among values, where a single cluster contains all the explanations from that layer. If the latter happens, it implies that neurons in the same layer have closely related embeddings, suggesting that they share semantic similarities. The side-by-side figures (Figure 2) shows the Gini coefficients for K-Means, DBSCAN, BERTopic and random clustering across different layers. We applied each clustering method for each of the filtering percentiles (see Subsection 2.2). While trends within the same clustering method remained consistent across different percentiles, distinct behaviors were observed between the various clustering methods. Overall, we find that there is a significant correlation between explanations from the same layer.

Random Clustering As mentioned before, we used random clustering to get a baseline for the performance of the other clustering methods. The Gini coefficient for each layer ranged from 0.04 to 0.05 for 0% filtering percentile, 0.06-0.08 for 50%, and 0.12-0.18 for 90%. The difference comes from the fact that randomly assigning more values will lower the variance. As expected, across all percentiles, the Gini coefficient is very low, indicating an almost equal distribution of layers between clusters.

K-Means First, we note that when using K-Means, the Gini coefficient across all layers is higher than the one observed in random clustering, with the lowest coefficient (across all percentiles) being 0.275 (layer 15, 90% filtering percentile). Second, and perhaps more interesting, lower layers (0-8) display a considerably higher Gini coefficient, approximately in the range of 0.45-0.55, in contrast to higher layers (40-48) where the coefficient falls within the range of 0.3-0.45. The Gini coefficients on both ends surpass those in the middle layers, which consistently exhibit coefficients ranging from 0.27 to 0.35. This concentration in the lower layers is in line with our intuition that the neurons in these layers have simpler explanations that are more similar to each other. For instance, the explanations *'words or word parts containing the letters "ED"'* and *'the word "ME" or a part of a word containing "ME"'* are both explanations of neurons from layer 0, and have a very similar embeddings.

BERTopic When using BERTopic, we found that the Gini coefficient is consistently higher than that of K-Means across all layers. It ranges between 0.4 to 0.55 in all layers and reaches a peak of 0.6 in the highest layer. This is in line with the fact that BERTopic uses HDBSCAN which is generally considered to be better and more effective than the basic K-Means approach. It is interesting to note that in BERTopic, the Gini coefficient in the middle layers doesn't fall off as much as in K-Means, which might indicate that those layers can be clustered together just as well, but K-Means is unable to do so.

DBSCAN In the case of DBSCAN clustering, we observed a distinct trend in the Gini coefficients across layers. First, as we progressed from lower to upper layers, we observed a consistent upward trend

in the Gini coefficient. Second, and of greater significance, the Gini coefficient consistently registered extremely high values, particularly at the 50% and 0% filtering percentiles, surpassing 0.75 across all layers. While this may seem like effective clustering, it's important to approach our analysis from multiple angles to gain a comprehensive understanding of its performance. We will explore these perspectives shortly.

3.2 Experiment 2: Cluster Variance

While the Gini measurement is a good way to understand the concentration of each layer, it is not sufficient on its own. To understand the quality of the clustering algorithm, we have to also look at it from the clusters' perspective and measure how diverse are each of our clusters. An intuitive and extreme example of this, is the case where the clustering algorithm assigned every explanation to the same cluster. The Gini coefficient will be 1 for all layers but it doesn't really mean that the clustering reveled an interesting phenomena. In that extreme case, if we had introduced a measurement to assess the diversity of the clusters, we would likely have obtained a very low score.

In our second experiment, we aimed to do that while also testing another assumption we had, that closely positioned layers share similar semantic meanings. We examined whether layers that are close to each other in the neural network, are more clustered together than faraway layers. Meaning, instead of looking at each of the layers in a discrete way, we want our experiment to reveal if there are some clusters which are composed of explanation mainly from nearby layers.

To explore this, we calculated the layer variance of each cluster. The variance of a cluster is the variance of the layers' indices corresponding to the neurons assigned to it. A low variance indicates that the explanations within that cluster originate from neurons in layers that are close to each other in the neural network, since low variance indicates concentration of indices. On the other hand, high variance indicates that the cluster explanations aren't concentrated around nearby layers. After calculating the variances of all clusters, we examined the resulting distribution, including measures such as median, mean, and a weighted average based on cluster size, which we denote as WAVG for brevity. We specifically examined the WAVG because of its resilience to outliers, especially in the case of tiny clusters with low variance. The details of our findings are depicted at Figure 3. In general, the results aren't remarkable, although there are certain clusters that stand out as interesting. We will present these clusters during the drill-down analysis (Section 3.3).

Random Clustering Like in the previous experiment, we used random clustering as a baseline for the other clustering methods. For both 0% and 90% filtering percentiles, the random clustering statistics match the expected results. First, the observed WAVG of the clusters variances is around 190. The expected variance of the unified distribution is denoted by the formula $\frac{n^2-1}{12}$, which for $n = 48$ equals to 191.91. Second, the results are very centered around the WAVG, which is expected for random clustering since all the explanations are handled uniformly.

BERTopic First, note that for BERTopic, the WAVG of variances is lower than that of random clustering, with around 175 for both 0% and 90% filtering percentiles. Even though the average variance of the clusters isn't much lower than of random clustering, there's still a noticeable difference. This gap suggests that certain clusters show a notably lower variance.

K-Means For K-Means, we observed very similar results to BERTopic. The WAVG of variances is also lower than that of random clustering, with around 175 and 170 for 0% and 90% filtering percentiles respectively. The similarity of results obtained with a different clustering algorithm, enhances our confidence in the robustness of the results.

DBSCAN In the case of DBSCAN, we observed the highest WAVG, almost as high as random. We also noticed that the mean was the lowest among all clustering methods. Further investigation revealed that this was because a single cluster contained 83% of the neurons, as displayed in Table 1.

This example highlights why the Gini coefficient alone may not provide a full understanding. While the Gini coefficient for DBSCAN was higher than in

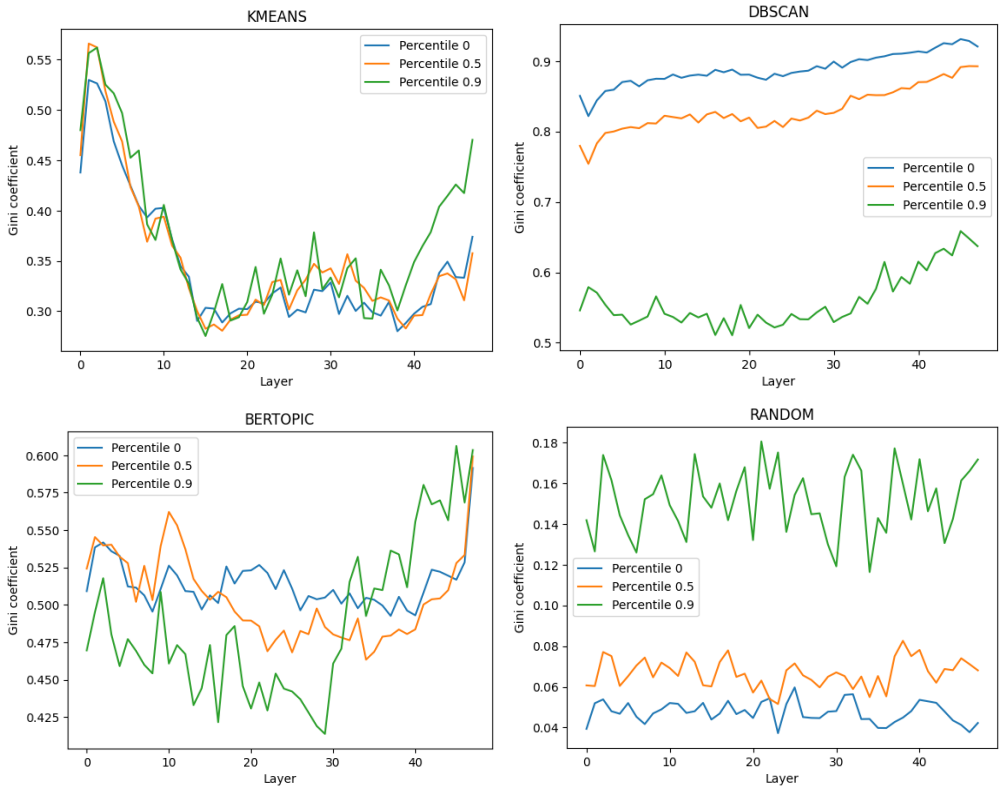

Figure 2: Gini coefficients for K-Means, DBSCAN, BERTopic and random clustering across layers. Different lines within each method represent various filtering percentiles.

Clustering Method	Total Neurons	Largest Cluster	2nd Largest	3rd Largest	4th Largest	5th Largest
DBSCAN	307,200	**249,313 (83.10%)**	658 (0.22%)	602 (0.20%)	497 (0.17%)	493 (0.16%)
K-Means	307,200	16,292 (5.43%)	14,199 (4.73%)	11,574 (3.86%)	11,366 (3.79%)	11,300 (3.77%)
BERTopic	307,200	7,640 (2.55%)	6,881 (2.29%)	5,155 (1.72%)	4,492 (1.50%)	4,190 (1.40%)
Random	307,200	6,552 (2.18%)	6,527 (2.17%)	6,525 (2.17%)	6,514 (2.17%)	6,509 (2.17%)

Table 1: Top 5 cluster sizes for each clustering method, including the percentage they represent out of all neurons, using a filtering percentile of 0.

other clustering methods, indicating a more concentrated cluster distribution, examining it from another angle revealed that this wasn't due to an interesting clustering pattern, but rather due to issues with the clustering algorithm.

3.3 Cluster Drill Down

In this section, we used BERTopic to manually examine selected clusters, with a particular focus on those demonstrating low layer variance, as previously discussed in Section 3.2. The choice of BERTopic was deliberate due to its capability to provide an interpretable representation of clusters, revealing topic names, representative words, etc. (see Figure 4).

To begin with, let's focus on the cluster with the minimal variance achieved with BERTopic using a filtering percentile of 0. This particular cluster exhibits a remarkably low variance of 55.8, a significant deviation from the expected variance in random clustering (191.91). As depicted in Figure 5, the low variance is due to a large number of neuron explanations belonging to the initial layers, with a median layer index of 9 and a 0.75th percentile layer index of 13. This result could indicate that low-level explanations, such as the recognition of individual letters, are associated with the lower layers, whereas explanations for deeper layers may lean more towards conceptual aspects rather than focusing on syntactic features.

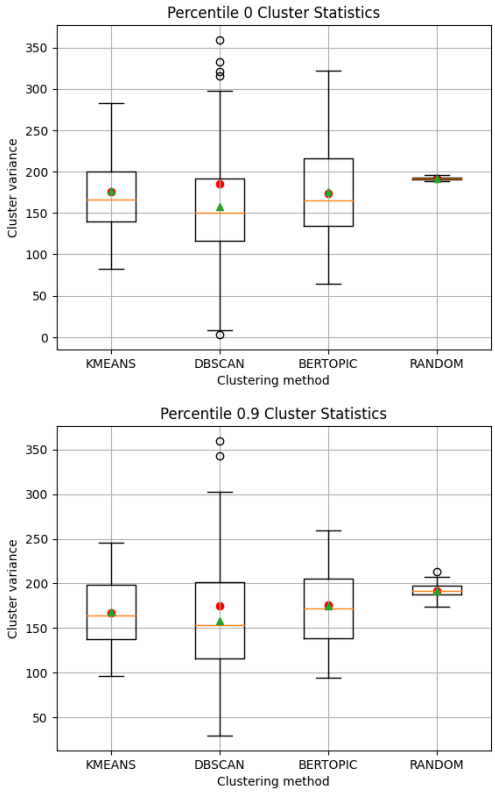

Figure 3: Cluster Variance boxplots for K-Means, DBSCAN, BERTopic and Random Clustering at Percentile 0 (top) and Percentile 90 (bottom). The red dot is the WAVG and the green triangle is the mean.

Additional details regarding other percentiles and their clusters are provided in Appendix A.

This examination shows that despite the average cluster variance not being particularly impressive, some clusters with low variance displayed significant correlations among the neurons in nearby layers. This observation suggests that neurons within these clusters might share similar responsibilities, which aligns with our desired outcome.

4 Limitations

Dataset Our analysis draws on the neuron explanations dataset from OpenAI, specific to GPT-2 XL. While providing valuable insights, it's crucial to acknowledge limitations in generalization to other language models, architectures, or datasets. In particular, the dataset is in English, which introduces inherent language bias, raising questions about applicability across languages with distinct linguistic

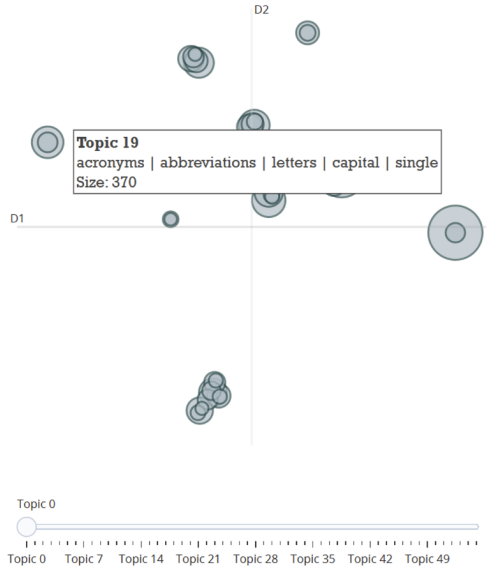

Figure 4: BERTopic visualization for percentile 90

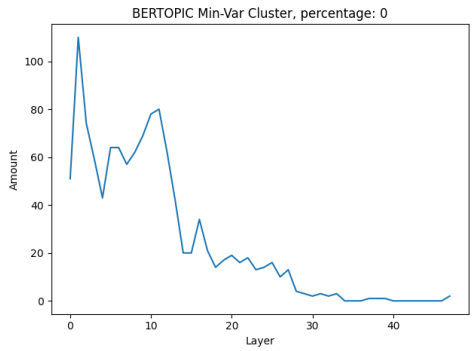

Figure 5: Distribution of layers (of neurons) that are assigned to the cluster with the minimal variance. The clustering was performed using BERTopic with no filtering percentile (0%).

structures. Additionally, the absence of a ground truth for neuron explanations in the dataset presents a significant challenge, particularly in cases where neurons have low scores, raising concerns about the reliability of their explanations.

Clustering Sensitivity to clustering algorithms and parameters is evident, as different choices may yield alternative patterns. The embedding algorithm, a general sentence transformer, is not fine-tuned for explanations. This can lead to situations where explanations with very similar structures have close embeddings but are relatively distant in terms of semantics. For example, *'words and phrases related*

Cluster	Variance	Median Layer	Topic	Representative Explanations
26	55.8	9.0	containing-letter-ending-combination	"words containing the letter 'e'." "parts of words or names containing the letter 'u'." "words or parts of words containing the letter 'é'."

Table 2: Details about the cluster with minimal variance. The clustering was performed using BERTopic with no filtering percentile (0%).

to scientific concepts and processes' and *'words and phrases related to numbers or numerical values'* are clustered together.

Addressing these limitations will not only enhance the validity of our findings but also guide future research directions for a more comprehensive understanding of neuron explanations in language models.

5 Future Work

Our exploration of neuron explanations in language models using clustering algorithms suggests several key areas for future research.

First, optimizing clustering parameters and exploring alternative algorithms is crucial for enhancing reliability and robustness. Another critical focus is the practical applications of understanding the correlation between proximate neurons in natural language processing tasks. In addition, a deeper dive into understanding the clusters with the highest and lowest variance, and the factors contributing to their variance, would provide valuable insights.

Expanding the study to language models trained on various languages may uncover language-specific patterns. Additionally, addressing challenges associated with low-scoring neurons is vital for ensuring the reliability of identified clusters.

Future work in these areas holds promise for advancing our understanding of language models and improving interpretability across diverse NLP tasks.

6 Conclusion

In this project, we have explored the dataset provided by OpenAI in their paper, "Language Models Can Explain Neurons in Language Models," and applied clustering algorithms, including K-Means, DBSCAN, and BERTopic, to the neurons' explanations.

We analyzed the clustering results from various perspectives, and also conducted manual examinations of a few clusters to gain deeper insights. We've observed that several clusters included neurons from neighboring layers, suggesting that neurons from nearby layers might be correlated in their role within the language model. Furthermore, we noticed a complementary finding, that neurons from the same layer often tend to concentrate in a small amount of clusters. This observation indicates that there may be some underlying structural or functional relationships between neurons in close proximity within the language model.

These insights may contribute to a better understanding of language model behavior at the neuron and layer level.

References

Steven Bills, Nick Cammarata, Dan Mossing, Henk Tillman, Leo Gao, Gabriel Goh, Ilya Sutskever, Jan Leike, Jeff Wu, and William Saunders. 2023. Language models can explain neurons in language models. https://openaipublic.blob.core.windows.net/neuron-explainer/paper/index.html.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

Corrado Gini. 1912. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]*. Tipogr. di P. Cuppini.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.*, 2(11):205.

Appendix

A Clusters with minimal variance

Percentile	Cluster	Variance	Topic	Representative Explanations
0	26	55.8	containing-letter-ending-combination	"words containing the letter 'e'." "parts of words or names containing the letter 'u'." "words or parts of words containing the letter 'é'."
0.5	229	70.41	prepositions-time-location-position	"prepositions related to time or location."
0.9	39	72.09	questions-asking-interrogative-question	"phrases related to asking questions." "asking-related phrases or questions."

Table 3: Details about the cluster with minimal variance, in different filtering percentiles. The clustering was done using BERTopic.

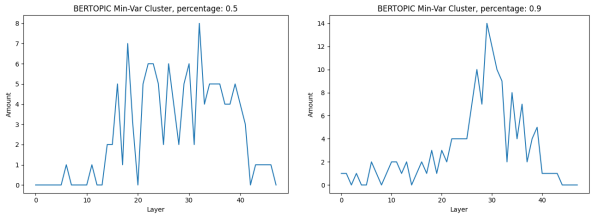

Figure 6: Distribution of layers (of neurons) assigned to the cluster with the minimal variance. Clustering was performed using BERTopic with filtering percentiles of 50 (left) and 90 (right).

B Top cluster sizes

Method	Total Neurons	Largest Cluster	2nd Largest
DBSCAN	153,600	111,219 (72.33%)	476 (0.31%)
K-Means	153,600	9,069 (5.90%)	7,303 (4.75%)
BERTopic	153,600	12,909 (8.41%)	3,221 (2.10%)
Random	153,600	3,324 (2.16%)	3,309 (2.15%)

Method	Total Neurons	Largest Cluster	2nd Largest
DBSCAN	30,720	7,491 (24.38%)	1,669 (5.44%)
K-Means	30,720	1,377 (4.48%)	1,291 (4.20%)
BERTopic	30,720	4,195 (13.65%)	1,429 (4.66%)
Random	30,720	690 (2.24%)	683 (2.23%)

Figure 7: Top 2 cluster sizes for each clustering method, including the percentage they represent out of all neurons, using filtering percentiles of 50 (top table) and 90 (bottom table).