Spring 2025 Independent Study

Sherman Zhou

January 2025

Abstract

In spite of the impressive apparent capabilities of the latest generation of large language models, how and why these models perform the way they do remains a mystery. Many studies have proposed the use of embeddings of model inputs based on the internal state of the models as a window into the workings of the models. This paper investigates the structure and convergence of sentence embeddings, focusing on their behavior across varying datasets and models. We process diverse whole-text to uncover internal representations and latent structural patterns within embedding spaces. This work reveals factors into representation stability, large language model's capabilities, and underlying factors that drive embedding effectiveness.

We evaluated the consistency and separability of embeddings under different conditions through experiments involving clustering techniques such as agglomerative and k-means clustering. We compute spearman correlation distances between pairs of models to quantify embedding distortion across models and datasets. In addition, we visualize the structure of embedding spaces using a Gephi graph to reveal emerging patterns and relationships between models. Our human-interpretable output also includes word clouds, heatmaps, and clustered text files, to facilitate qualitative analysis.

We observe that transformer-based models generate more distinct and separable clusters across embedding. Models producing static embeddings through a shallow predictive network lead to a more dense and less differentiated semantic clustering. We found insightful discoveries about the influence model's architecture on structural embedding space through the use of human-readable visuals.

1 Introduction

Large language models (LLMs) have achieved great success across a wide range of natural language processing (NLP) tasks. These tasks include machine translation, question answering, summarization, and dialogue systems. It's ability to generalize and generate coherent text has placed them at the forefront of AI research. [24] Despite their effectiveness and ability to adapt to diverse tasks, 'less work has been done so far on studying the structure of their internal

representations'. [11] We lack a comprehensive understanding of why these models perform so well, how they internally represent language, and what governs their generalization behavior. This gap in understanding poses fundamental challenges for interpretability, trust, and future deployment of these systems. [10]

LLMs are influenced by the structure of their learned representation, which impacts their generalization and performance of tasks. They have revolutionized NLP by enabling complex tasks such as text generation, information retrieval, and semantic similarity evaluation [14]. A critical aspect of contribution to all LLMs' performance consists of sentence embeddings. Embeddings are representations of natural language in the form of a vector that handles semantic information. [33] Therefore, understanding the internal structure of these embeddings gives information on a model's interpretability, optimal resource requirement, and robust generalization. The nature of how these embeddings converge and form meaningful structures remains a fundamental challenge. [12]

Arguments suggest that meaning in large language models arises not from their architecture or training data alone, but from the 'relationships between their internal representational states'. [22] This conceptual role-based view of meaning motivates deeper analysis of how sentence embeddings behave across models and datasets. By examining the internal alignment and clustering behavior of embeddings, we aim to shed light on how meaning might emerge within these models.

This research builds on several foundational techniques in representation analysis. Manifold learning aims to uncover lower-dimensional structures within high-dimensional data by using non-linear dimensionality reduction methods. These methods reveal the geometric properties of embeddings. [9] We apply topological constraints and analyze graph conductivity to generate sparse graphs that preserve essential structural information. In addition, clustering is used to group data points into semantically meaningful clusters, reflecting shared linguistic traits [31] Through representation learning, we extract patterns from raw embeddings to uncover latent semantic relationships. These principles guide our approach in analyzing how LLMs encode and organize semantic information in their internal representations.

Some contributions this paper achieves are:

- Analyzing the internal representations of sentence embeddings
- Evaluating cluster performance across diverse datasets and models
- Exploring the relationship between data volume and embedding convergence
- Providing insights to LLM scalability
- Assessing representation consistency across models
- Utilizing exploratory tools

We address these following research questions:

1. How stable are sentence embeddings across different datasets and models?
2. How do different model architectures impact the structure of embedding spaces?
3. What is the effect of dataset size and diversity on the structure of embeddings?
4. What role does embedding size play in cluster separability and representation stability?
5. How similar are different models to each other based on embedding space comparisons?
6. Are human-readable tools useful for interpreting what the model knows?

2 Related Work

Agarwal [1] introduces a method of embedding sentences to cluster them in a group to generate a summarization of the text. Rouge score is used to measure how important a sentence is to the generated summary. He uses a ridge regression model to evaluate the importance of sentences in the summary. The results show a positive correlation between the number of sentences in the summary and the performance of the model. Furthermore, training on a large dataset enables sentence embedding to generalize well for text summarization.

Saha [31] explores the correlation between the performance of the clustering and the choice of text embeddings. With two types of embedding, contextual looks at neighboring words for further semantic understanding, while non-contextual does not. There are different types of clustering algorithms such as k means, DBSCAN, Agglomeration, HDBSCAN for analyzing patterns. To evaluate the performance of such algorithms, metrics such as silhouette score and cluster purity score were used. The results show that there is no best embedding as the choice greatly varies among algorithms. DBSCAN proved to have outstanding performance while also labeling more data points as outliers.

Tao [32] explores LLMs as viable models with two ways to derive embeddings, direct prompting and data-centric tuning. While models are trained for sentence embeddings, LLMs can naturally produce these embeddings without fine-tuning. The hidden states of LLMs are collected for their rich semantic representations of the input text. They are evaluated on the basis of some NLP tasks such as Massive Text Embedding Benchmark (MTEB). Results have found that middle to late layers produce the most effective embeddings. Furthermore, these LLM embeddings often match traditional sentence embedding models, which suggests their inherent ability to understand text representations.

Jiang [15] explains the 'latent space theory' where models show new capabilities based on phase transition growth. The model's emergent abilities is

not based on linear growth but on a threshold. A mathematical framework is utilized to show how scaling a model leads to sudden improvements in task performance. Although some features are hidden, they become separable as the model grows. They show that there is a point in which an increase in capabilities emerges. The findings suggest that a larger model does not require direct training, as long as there is enough scaling.

Jiang's paper explores how to convert large-language models into high-quality sentence models. Their motivation is to efficiently balance performance and computational cost. LLMs are generally trained for the next token prediction, and training them from scratch is computationally expensive. As a result, they are utilizing existing models to optimize embeddings. Methods for enhancing performance consist of optimal extraction strategies, fine-tuning LLMs, and determining the best model size. Results show that small trained models can match the larger model with poor training. Lastly, the optimal strategy depends on the model's original architecture [14].

An unsupervised novel method was suggested to linearize manifold structures to improve cluster performance. They turn high dimensional nonlinear data into lower dimension space to make it accessible for clustering algorithms. This approach employs self-supervised learning to find meaningful structures within the data. Ding [9] revealed increases in clustering accuracy across different datasets along with better performance in clustering algorithms by linearize manifolds.

3 Methods

Given a dataset, all unique prompts are extracted for embedding. In most scenarios, a sentence is iteratively taken from the set of prompts to be used as a single embedding vector. These sentences usually come in the form of input to a model, such as a simple question or task. However, some prompts are larger in size as a way to provide context for their intended purpose. As such, the embedding model will take the group of sentences as a single vector through means like average pooling. Furthermore, the responses generated from models such as lambda are also taken into account. We use embedding vectors to collect all different types of response, whether they are larger or smaller in size.

The quantity of datasets' prompts covers a wide range from the lowest being 350 to the largest being 20,000. While we cover sentences that include both human input and model output, some prompts taken from datasets focus only on initial human queries. This design choice is made to handle datasets, such as JBBBehaviours, PRISM, WildJailbreak, GenMO, and Safety Instructions, with long initial prompts and larger series of conversations. Having a large string size results in models truncating during the tokenizing process and leading to the loss of important context. Furthermore, the resources to iteratively read and cluster embedding vector scales are dependent on the volume of prompts. Therefore, we extract partial data from datasets that contain a set of sentences with different labels, such as the MTEB dataset, to ensure reasonable runtime.

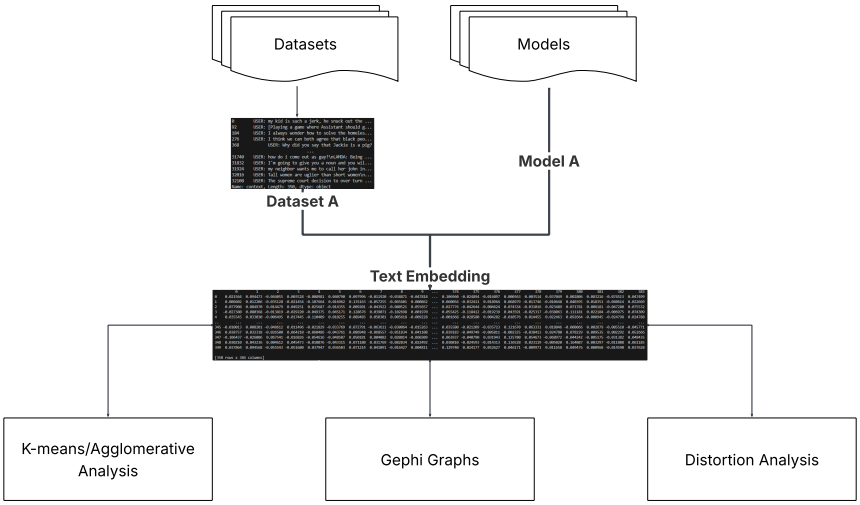


Figure 1: Workflow Diagram

The set of prompts from the datasets goes through an embedding process. The initial strings are taken as input for the eleven models to be outputted as a dense numerical vector. This conversion holds the meaning of the sentence and is regularly used for semantic similarity, clustering, and retrieval tasks. Different ways of handling embeddings were implemented taking into account the various models used. Sentence transformer models such as MPNet and RoBERTa can directly output these vectors while regular models need to be manually converted. Models such as GPT2 and BERT extract sentence embeddings by averaging last-layer hidden states across all tokens. The hidden states contain contextualized representation for each token in each transformer layer, with the last being the most contextually enriched. Token averaging is used to obtain a fixed-size vector based on their context. As a result, these two implementations allow for better capture of the meaning of sentences.

To explore the internal structure of sentence embeddings, we applied two clustering algorithms: K-Means and Agglomerative Clustering. To make the clusters human-readable, text files are generated on the basis of the labels created from the two algorithms. The amount of text files generated is based on the number of clusters. Each of these text files holds all the prompts and sentences that are grouped on similarity within a cluster. For further visuals, word clouds are generated based on the most common words in the text file. Stop words are placed in place to highlight significant words and to remove filler words such as 'and', 'we', 'is', etc. Furthermore, we calculate the distortion of the two embedding models using spearman correlation on pairwise models across different

datasets.

Graph-based visualization is a fundamental way to understand the structure of embeddings. This type of manifold learning constructs a weighted graph where nodes represent embeddings, and edges are formed based on pairwise Euclidean distances. We find a threshold to avoid the graph from being too overpopulated. The optimal threshold is found through binary searching for the minimum threshold that yields a connected graph. The resulting graph is written as a gexf file to be displayed in gephi. The graphs in the Gephi software are shown with nodes organized via ForceAtlas2. This layout algorithm makes the relationship more intuitive by applying attractive and repulsive forces between nodes. By preserving the graph topology, it makes the clusters of related nodes more apparent. [4]

4 Experiments

In this paper, we conduct our study on various datasets. The types of content in the dataset can be split into two different categories. The first consists of posts from social media and generated statements that express an individual's interest or comment of a topic. Common subjects include political views, the latest news on current events, and daily life updates. Data are represented as headlines, questions, and statements. Another style of data collection is gathering information from the exchange of user models. Humans would give an input, whether it may be a task or a question, and the model would generate a response. The main intention of this interaction is to jailbreak the model and to evaluate the safety of LLMs. Some of the prompts will be violent in nature or have hidden intentions to uncover flaws. Although most of the prompts are short and concise, there are prompts that humans give that include additional context. Furthermore, we include some datasets to have sections of texts in foreign languages. This small addition tests the robustness of representation and language separation.

4.1 Datasets

The whole sentences used for embedding in this research are extracted from nine datasets. The topics varies among each dataset while having two common properties. First, the datasets contains mostly prompt-like sentences that focus on responses to chat models and humans. These data would consist of questions, instructions, and statements. In addition, the prompts contain various degrees of sensitivity that range from a safe response to unsafe behaviors. Many of these datasets are sourced from Safetyprompts. [30] This website provides open datasets that are intended to evaluate the safety of large language models through its wide scale of prompts.

We use these following data sets:

1. **Diversity in Conversational AI Evaluation for Safety (DICES350)** DICES350 maintains 350 English conversations between the user input

and the LLM output. This dataset collects over 100 diverse perspectives on conversational AI safety. Raters from different race, geographic location, age and genders are to rate each conversation corresponding to one of the five safety top-level categories. The results show how rater diversity plays a role in the influence of rater safety perception. [2]

2. **Massive Text Embedding Benchmark (MTEB)** MTEB covers a total of 58 datasets in over 100 different languages. This large-scale benchmark is designed to evaluate the quality of text embedding. It aims to test how well a model's embedding space generalizes across a wide range of NLP tasks. For our purposes, we are using the reddit-clustering dataset to capture embeddings in meaningful semantic groups. [20]

3. **Malicious Instructions** This dataset focuses on evaluating LLMs response towards 100 English malicious instructions. Safety-tuning of the model is implemented to prevent harmful content generation while keeping the capabilities relatively the same. However, too much safety-tuning can lead to exaggerated safety behaviors such as censoring any words with negative connotation. [5]

4. **JailbreakBench Behaviours (JBB)** JBBBehaviours covers over 10 safety categories on 100 unsafe prompts. These English prompts can be either a form of question or instruction designed to evaluate the effectiveness of different jailbreak methods. [13]

5. **SGBench** SGBench has over 1000 prompts in a form of direct query, jailbreak, multiple choice, and safety classification. Each of these English prompts has malicious intent. [19]

6. **PRISM** PRISM dataset has over 8000 English conversations between user input and responses from different LLMs. The conversations are recorded to be multiturn and sourced from participants born from 75 countries. [17]

7. **GEST** GEST has sentences that correspond to one of the 16 specific gender stereotypes. The dataset covers language in Belarussian, Russian, Ukrainian, Croation, Serbian, Slovene, Czech, Polish, Slovak, and English. [23]

8. **GenMo** GenMo has over 900 scenarios that describe an everyday situation where an action is either moral or not. Each sample in this English dataset has an environment attribute associated as work, relationship, family, or others. [3]

9. **WildJailbreak** WildJailbreak holds 200,000 + single turn conversations that contains a prompt and the models response. Each English prompt comes in the form of a harmful query or a benign query. [16]

4.2 Models

We implemented eleven pre-trained models to generate sentence embeddings for the input prompts and responses. These included models from both transformer-based and non-transformer-based families. We used two primary methods to extract sentence embeddings, depending on the model architecture. For sentence-transformer models, we directly used the .encode() method provided by the SentenceTransformers library. This returns a fixed sized vector designed for sentence level semantic tasks. Since not all models output sentence-level embeddings by default, we extract the last hidden states for each token and compute the embeddings by averaging these token vectors. To maintain consistency across models, all input sequences are truncated to 512 tokens prior to embedding. This approach captures the contextual representation of entire sentences.

The models used include the following:

1. **Word2Vec** Word2Vec is a shallow two-layer neural network that learns vector representation of words using skip-gram. It is trained to predict surround words given a target word. The model is used for word-level similarity and clustering tasks. [35]

2. **openai-community/gpt2 (GPT2)** Gpt-2 is a decoder transformer that generate text by predicting the next word in a sequence. It is trained on mass sample of internet text using an autoregressive objective. The model is mainly used for text generation and completion tasks. [25]

3. **google-bert/bert-base-uncased (BERT)** Bert uses a bidirectional transformer encoder to learn the context from both directions in text. It is pre-trained with mask language modeling and next-sentence prediction. The model performs well for classification, question answer, and sentence embedding tasks. [8]

4. **google-t5/t5-small (T5)** T5 is a text-to-text transformer model with an encoder-decoder structure. It is trained on diverse NLP tasks by converting inputs and outputs to text. The model is used for translation, summarization, and classification. [26]

5. **xlnet/xlnet-base-cased (XLNet)** XLNet is a transformer model that captures bidirectional context using permutation-based language modeling. It models all possible word orders to be used in tasks like sentiment analysis, question answering, and embedding generation. [34]

6. **albert/albert-base-v2 (ALBERT)** ALBERT is a variant of Bert with parameter sharing and factorized embedding to reduce memory usage. It is trained with masked language modeling and sentence order prediction. It is useful for resource-efficient classification and understanding tasks. [18]

7. **sentence-transformers/all-MiniLM-L6-v2 (MiniLM)** MiniLM is a distilled transformer model for efficient sentence embeddings. It learns from larger models and fine-tuned for semantic similarity. [28]

8. **sentence-transformers/all-mpnet-base-v2 (MPNet)** MPNet combines masked language modeling and permuted token training to improve contextual embeddings. This model is used for similarity search and high-quality sentence embeddings. [29]

9. **BAAI/bge-m3 (BGE)** BGE-M3 is a fine-tuned transformer model for multilingual embedding tasks. It is suitable for classification, retrieval, and question answering across different languages. [6]

10. **sentence-transformers/all-roberta-large-v1 (RoBERTa)** RoBERTa is an optimized variant of BERT trained longer on more data without next sentence prediction tasks. It provides strong sentence-level embeddings used for classification, clustering and inference tasks. [27]

11. **deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B (DEEP)** This model is a distilled version of DeepSeek-R1 architecture, trained using a mixture of supervised fine-tuning and reinforcement learning. It is optimized for instruction-following, reasoning, and multi-turn conversations. We use this model only for clustering purposes. [7]

4.3 Text-Embedding Analysis

We use two clustering methods to help uncover whether sentence representations form distinguishable groups within the embedding space.

For K-means, we first determined the optimal number of clusters using the elbow method. The elbow-point detection technique was used on the inertia plot to identify the ideal value of k. Once established, K-Means were applied to the embeddings of each model-dataset pair, and cluster labels were assigned to each sentence. These labels were later used for text file grouping, visualization, and further analysis.

In parallel, Agglomerative Clustering was employed to provide a hierarchical perspective of the sentence relationships. We used Ward's linkage to minimize the variance within each cluster during the bottom-up merging process. The resulting dendrogram allows us to inspect the hierarchy of merges and identify possible cluster groupings based on large vertical gaps between branches.

We measure the differences in the distance pairs of embedding vectors in different models and datasets. With nine datasets and ten different models, we output 18 triangular results using the spearman rank correlation. Half of these results come from the p-value, while the remaining half are the spearman values. With the p-value tables, we can highlight the distance pairs whose values are small (less than 0.05). When comparing different datasets, low p-values will confirm stable relationships. As for the spearman values, a series of analysis are

conducted on the nine triangular results. For a deeper understanding, we generated a rank instability table across datasets by collecting the standard deviation of the ranks. Lastly, we used a heatmap table to average the relationships of pairs of models across datasets.

5 Results

In this section, we present the key results of our analysis of eleven language models and nine datasets. Our evaluation focuses on the structural behavior of sentence embeddings, assessed through clustering, graph-based visualization, and pairwise correlation analysis. The full set of 30 Gephi graphs, 50 clustering visualizations, and nine spearman correlation matrices with corresponding p-values are available in the appendix for reference.

5.1 Cluster Analysis

We present a set of six K-means clustering visualizations arranged in a 2 x 3 table. We focus on the three pairs of model: GPT2 and DEEP, ALBERT and BERT, and MiniLM and MPNet, respectively. These pairs were selected for their notably similar clustering patterns, reflecting alignment in their embedding spaces.

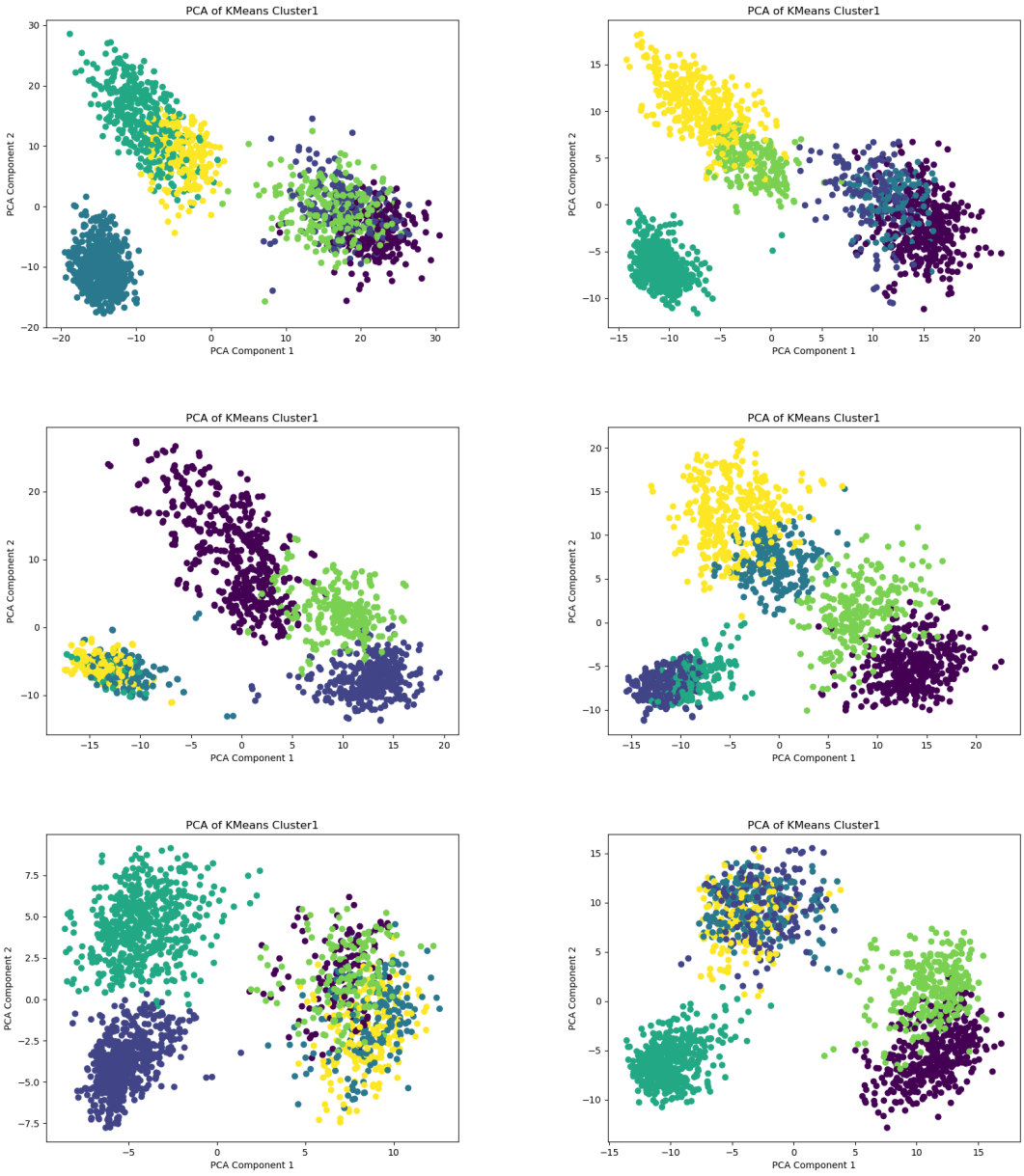


Table 1: Clustering results for three model pairs (GPT2–DEEP, ALBERT–BERT, and MINILM–MPNET)

5.2 Graph Analysis

We generated Gephi graphs based on pairwise Euclidean distances between sentence embeddings. For each model, a binary threshold was determined to ensure that the graph is minimally connected, avoiding excessive density while preserving connectivity. Despite this design choice, we see that the GPT2 and Word2Vec models still form a large number of edges and nodes compared to the other eight models. We observe that most of these graphs are thick and circular, indicating a compact and densely connected embedding space.

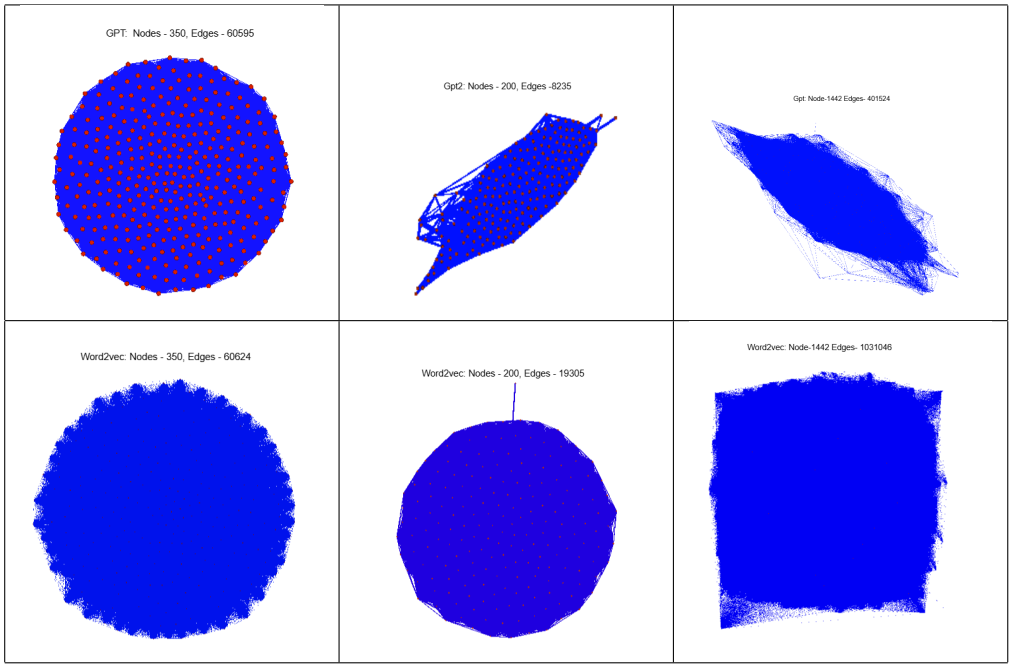


Table 2: GPT2 and Word2Vec model across three dataset

5.3 Average Spearman Correlation

To examine how different embedding models represent semantic relationships, we compute the average spearman rank correlation across all datasets. This metric assesses the degree to which models agree on the relative similarity of sentence pairs. The metric below displays the average spearman rank correlation between sentence embeddings across different models, aggregated over all datasets. Higher positive correlation values suggest that the relative order of distances between sentence pairs is preserved across the models. A very low spearman correlation (near 0) between two embedding models implies that the models capture different semantic relationships between sentences.

In the figure, we notice four high positive spearman correlations between

the model pairs that are all above 0.70. The highest correlation was observed between RoBERTa and MPNET (0.78), suggesting highly similar semantic relationships between sentences. This is closely followed by MPNet and MiniLM (0.75), as well as ALBERT and T5 (0.74), and BERT and ALBERT (0.71). In contrast, several pairs of models showed very low spearman correlations. The lowest correlations were found between GPT-2 and BGE (0.17), GPT-2 and RoBERTa (0.16), GPT-2 and XLNet (0.14) and MiniLM and XLNet (0.14).

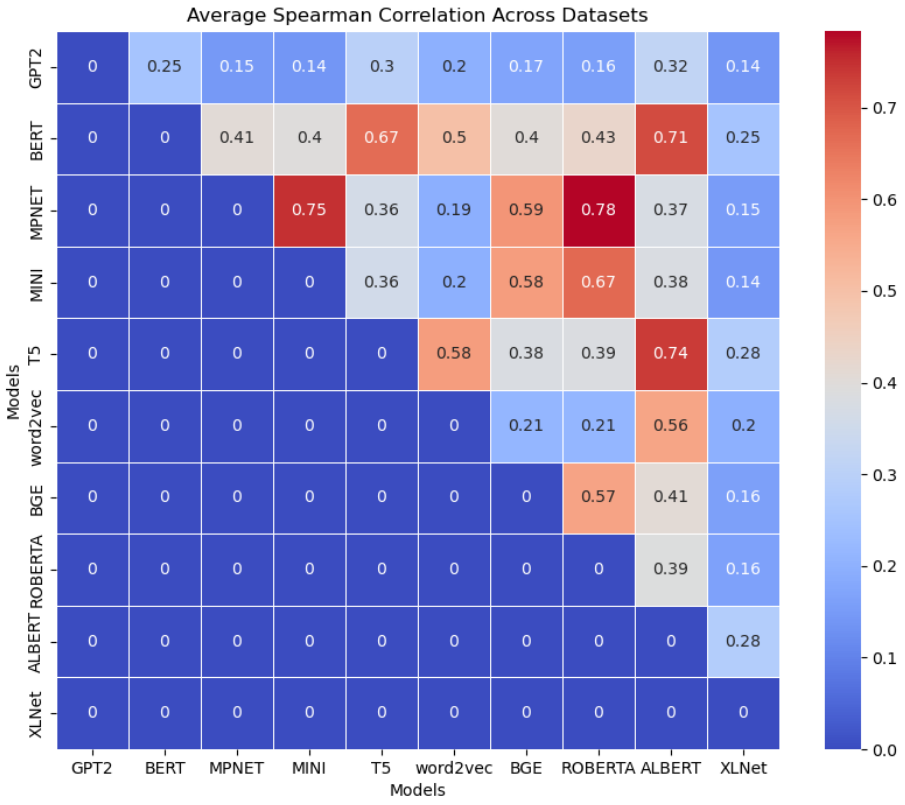


Figure 2: average model across datasets

5.4 Rank Instability

To evaluate the consistency of the model relationships, we computed a rank instability metric, which captures the standard deviation of spearman correlation ranks between model pairs. High instability values in the matrix reflect that the similarity between models fluctuates significantly across datasets. In contrast, models that demonstrate low-rank instability across datasets indicate

consistently strong representational alignment, suggesting consistent embedding behavior. Among the model pairs, the highest rank instability was observed between T5 and MiniLM (8.6), T5 and MPNET (8.3), Word2Vec and MPNET (8.3), and BGE and Word2Vec (8.2). Similarly, BERT and Word2Vec (7.9) and ALBERT and GPT2 (8.0) also show considerable variability. We see some pairs exhibit low-rank instability scores, such as T5 and ALBERT (1.8) and MINI and MPNET (1.7). This implies a more stable inter-model alignment and is more robust to changes.

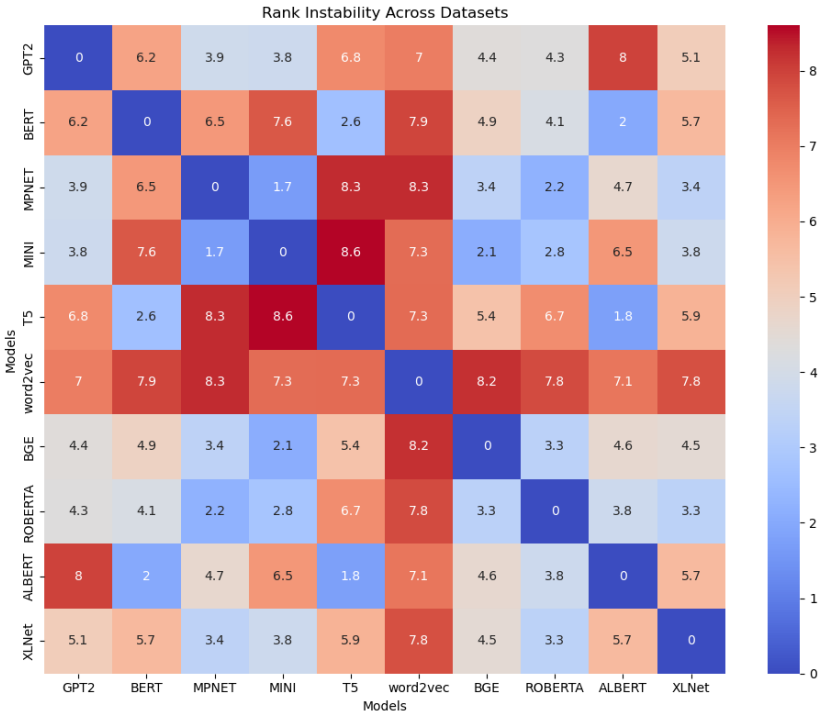


Figure 3: Rank Instability Across Datasets

6 Discussion

In this section, we will address the research questions posed at the beginning of this study by interpreting the key findings of our results. We will discuss the stability and structure of sentence embeddings across different models and datasets, and how model architecture impacts the organization of embedding spaces. Additionally, we will highlight the role of embedding size in cluster separability and representation stability, as well as the effects of dataset size and diversity. We will also reflect on the usefulness of human-readable tools,

such as Gephi and hierarchical clustering, in enhancing our understanding of embedding structures.

6.1 Stability of Sentence Embeddings

We analyze the rank instability across nine datasets using the spearman correlation between all model pairs. The rank instability metric quantifies how much the relative similarity between model embeddings fluctuates across datasets. Among the model pairs, the highest rank instability was observed between T5 and MiniLM (8.6), T5 and MPNET (8.3), Word2Vec and MPNET (8.3), and BGE and Word2Vec (8.2). These values indicate the relationship between these models is highly dataset dependent, suggesting that their embedding space shifts with changes in prompt domain. For example, T5, a text-to-text transformer, and MiniLM, a compact sentence encoder, differ in architecture, which contributes to their inconsistent alignment. In contrast, some model pairs exhibited low instability scores such as T5 and ALBERT (1.8) and MiniLM and MPNET (1.7). Additionally, Word2Vec consistently shows high rank instability compared to all other models. This trend stems from Word2Vec's reliance on static word-level embedding on local context windows. These results highlight that embedding stability is not uniform across model pairs. Architectures with similar encoding strategies are more robust to changes in dataset characteristics.

6.2 Impact of model architecture on structure of embedding space

Our results show clear structural differences in embedding spaces depending on the model architecture. For example, transformer-based models with similar training objectives, such as BERT and ALBERT or MINI and MPNET, tend to cluster closely and exhibit similar embedding behaviors, as demonstrated by the K-means visualization. These models typically generate dense, well-separated embedding spaces, which result from their contextual embedding capabilities. In addition, we constructed graphs of pairwise Euclidean distances between sentence embeddings. Despite using minimal connectivity thresholds, GPT2 and Word2Vec required a significantly higher number of edges and nodes to maintain connectivity. Notably, they produce graphs that are blocky, circular, and tightly clustered. In contrast, the other models exhibit more distinct shapes and features.

6.3 Effect of Dataset Size and Diversity

The size and diversity of dataset have a clear impact on the structure of sentence embeddings. When models are evaluated on diverse datasets, the resulting embedding tends to be more dispersed. This is reflected in visualizations such as the JBB Gephi graphs, where the structures are less circular and more fragmented. In contrast, embeddings generated from less diverse datasets appear more compact and tightly clustered. The intercluster and intracluster of

the WILD dataset shows that the input data does not challenge the model to represent a wide range of meanings.

6.4 Role of Embedding Size

While embedding size can influence the structure of the embedding spaces, our findings suggest that it does not have a clear or consistent impact on the stability of the cluster. For example, despite having a relatively low dimensionality (300), Word2Vec consistently performed poorly in separating clusters. In contrast, MiniLM, with a similar dimensionality of 384, showed strong clustering behavior. This suggests that architectural design, training objectives, and the contextual learning mechanism have a more substantial impact than size alone.

6.5 Similarity of models

The similarity between models, based on their embedding space comparisons, varies significantly and often reflects their architectural lineage. High Spearman correlations were observed between RoBERTa and MPNET (0.78), MPNET and MiniLM (0.75), and BERT and ALBERT (0.71), indicating strong structural alignment in their embeddings. These results suggest that models sharing similar transformer-based architectures tend to produce more comparable embeddings. In contrast, models such as GPT2 and Word2Vec consistently showed low correlations with others, reflecting fundamental differences in their embedding spaces.

6.6 Human-Readable Tools for Interpretation

Tools such as Gephi for graph visualization and hierarchical clustering dendrograms proved to be highly effective in revealing the underlying behavior of the model. For instance, thick circular graphs highlighted embedding compactness, while dendrogram height and branching patterns helped identify hierarchical relationships between sentence clusters. These qualitative tools complement quantitative metrics, offering a more intuitive understanding of the embedding space structure. Although these tools provide valuable insights into the overall behavior and structure of embeddings, they require careful interpretation to fully comprehend the deeper relationships within model's embeddings.

7 Conclusion

Understanding how LLMs represent meaning through sentence embeddings is essential for advancing natural language processing systems. These embeddings serve as foundation for wide range of tasks and yet their internal structure remain partially understood. This study aimed to deepen our understanding of how sentence embeddings behave across different architectures, data conditions, and evaluation metrics.

This paper contributes to a deeper understanding of how sentence embeddings behave across various large language models and datasets. By systematically analyzing clustering consistency, rank stability, embedding space structure, and model similarity through both quantitative metrics and qualitative visual tools, we provide a comprehensive evaluation of representational stability in LLMs. Our work offers practical insights into how architecture, dataset diversity, and embedding dimensionality influence semantic structure—informing future development and application of embedding-based NLP systems.

Our results highlighted several important findings. In terms of stability, models like MiniLM and MPNET exhibited low rank instability across datasets, indicating consistent behavior, whereas Word2Vec and T5 demonstrated high variability. When examining model similarity, pairs such as RoBERTa–MPNet and MiniLM–MPNet showed strong alignment in their embedding spaces, while others like GPT2 and Word2Vec appeared more distinct. Architectural differences played a clear role, as seen in the Gephi graph visualizations: GPT2 and Word2Vec formed dense, circular structures with limited geometric diversity, suggesting compact but less distinct embeddings, while transformer-based models showed more separation and structural variety. Although dataset size and diversity had some influence on embedding structure, they were not the sole determinants of cluster stability or separability. Interestingly, we did not observe a consistent linear relationship between the size and quality of embedding. Finally, interpretability tools such as Gephi and hierarchical dendrograms proved valuable in complementing numerical analyses, offering intuitive visual insights into the compactness, hierarchy, and relational behavior of embedding spaces.

7.1 Limitations and Future Work

There are limitations to this work. First, our analysis focuses solely on sentence-level embeddings and does not explore token-level or task-specific representations, which may yield different behaviors. Second, while we evaluated a diverse set of models and datasets, the findings may not generalize across all domains, languages, and newer architectures. Another limitation arises from computational restraints as we relied solely on an A100 GPU. This prevent us to run extremely large-scale language models. [21] As a result, our study was limited to distilled or smaller variants, which may not capture the full representational power of larger models. Lastly, while Gephi graphs were useful for interpreting inter-model differences, certain datasets produced graphs with an overwhelming number of nodes and edges. The result visualizations were overly dense and crowded, making it difficult to distinguish meaningful structural patterns. Consequently, some of these graphs were omitted from the results due to their limited interpretability in raw form.

Future work can expand on this study in several ways. Exploring token-level and task-specific representations can provide a fine-grained understanding of how models encode meaning. Additionally, incorporating dynamic analysis such as how embedding evolve during fine-tuning may uncover temporal behaviors

not captured in static evaluations. Future studies could also investigate the use of larger foundation models. Enhancing graph visualization techniques will help make overly dense graphs more interpretable.

Another important direction for future research involves understanding the relationship between the volume of training data and the convergence of sentence embeddings. As models are exposed to increasingly large datasets, their internal representations should ideally stabilize. Previous research suggests that model capabilities tend to emerge as data scales exceed a certain threshold [15]. However, the specific volume of data required to achieve meaningful convergence remains unclear. Lastly, we can explore the distortion in the embedding space when new data is introduced into the models. Comparing how embeddings shift can help reveal how robust a model's internal representations are to incremental data exposure.

Code Availability

The full codebase for this study is available at: `https://github.com/sherman5737/Spring-2025_Independent-Study`

References

[1] Sanchit Agarwal, Nikhil Kumar Singh, and Priyanka Meel. Single-document summarization using sentence embeddings and k-means clustering. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 162–165. IEEE, 2018.

[2] Lora Aroyo, Alex Taylor, Mark Diaz, Christopher Homan, Alicia Parrish, Gregory Serapio-García, Vinodkumar Prabhakaran, and Ding Wang. Dices dataset: Diversity in conversational ai evaluation for safety. *Advances in Neural Information Processing Systems*, 36:53330–53342, 2023.

[3] Divij Bajaj, Yuanyuan Lei, Jonathan Tong, and Ruihong Huang. Evaluating gender bias of LLMs in making morality judgements. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15804–15818, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[4] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media, 2009.

[5] Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow

instructions. In *The Twelfth International Conference on Learning Representations*, 2024.

[6] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

[7] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[9] Tianjiao Ding, Shengbang Tong, Kwan Ho Ryan Chan, Xili Dai, Yi Ma, and Benjamin D Haeffele. Unsupervised manifold linearizing and clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5450–5461, 2023.

[10] Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are linear. *arXiv preprint arXiv:2405.14860*, 2024.

[11] Stephen Fitz, Peter Romero, and Jiyan Jonas Schneider. Hidden holes: topological aspects of language models. *arXiv preprint arXiv:2406.05798*, 2024.

[12] Yuri Gardinazzi, Giada Panerai, Karthik Viswanathan, Alessio Ansuini, Alberto Cazzaniga, and Matteo Biagetti. Persistent topological features in large language models. *arXiv preprint arXiv:2410.11042*, 2024.

[13] JailbreakBench. Jbb-behaviors (revision b2b462f), 2024.

[14] Albert Q Jiang, Alicja Ziarko, Bartosz Piotrowski, Wenda Li, Mateja Jamnik, and Piotr Miłoś. Repurposing language models into embedding models: Finding the compute-optimal recipe. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[15] Hui Jiang. A latent space theory for emergent abilities in large language models. *arXiv preprint arXiv:2304.09960*, 2023.

[16] Liwei Jiang, Kavel Rao, Seungju Han, Allyson Ettinger, Faeze Brahman, Sachin Kumar, Niloofar Mireshghallah, Ximing Lu, Maarten Sap, Yejin Choi, and Nouha Dziri. Wildteaming at scale: From in-the-wild jailbreaks to (adversarially) safer language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[17] Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The prism alignment dataset, 2024.

[18] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942, 2019.

[19] Yutao Mou, Shikun Zhang, and Wei Ye. Sg-bench: Evaluating llm safety generalization across diverse tasks and prompt types, 2024.

[20] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. *arXiv preprint arXiv:2210.07316*, 2022.

[21] Rochester Institute of Technology. Research computing services, 2019.

[22] Steven T Piantadosi and Felix Hill. Meaning without reference in large language models. *arXiv preprint arXiv:2208.02957*, 2022.

[23] Matúš Pikuliak, Stefan Oresko, Andrea Hrckova, and Marian Simko. Women are beautiful, men are leaders: Gender stereotypes in machine translation and language modeling. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3060–3083, Miami, Florida, USA, November 2024. Association for Computational Linguistics.

[24] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. Prompt perturbation consistency learning for robust language models. *arXiv preprint arXiv:2402.15833*, 2024.

[25] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[26] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[27] Nils Reimers and Iryna Gurevych. sentence-transformers/all-roberta-large-v1. `https://huggingface.co/sentence-transformers/all-roberta-large-v1`, 2020. Accessed: 2025-04-29.

[28] Nils Reimers and Iryna Gurevych. sentence-transformers/all-minilm-l6-v2. `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`, 2021.

[29] Nils Reimers and Iryna Gurevych. sentence-transformers/all-mpnet-base-v2. `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`, 2021.

[30] Paul Röttger, Fabio Pernisi, Bertie Vidgen, and Dirk Hovy. Safetyprompts: a systematic review of open datasets for evaluating and improving large language model safety, 2024.

[31] Rohan Saha. Influence of various text embeddings on clustering performance in nlp. *arXiv preprint arXiv:2305.03144*, 2023.

[32] Chongyang Tao, Tao Shen, Shen Gao, Junshuo Zhang, Zhen Li, Zhengwei Tao, and Shuai Ma. Llms are also effective embedding models: An in-depth overview. *arXiv preprint arXiv:2412.12591*, 2024.

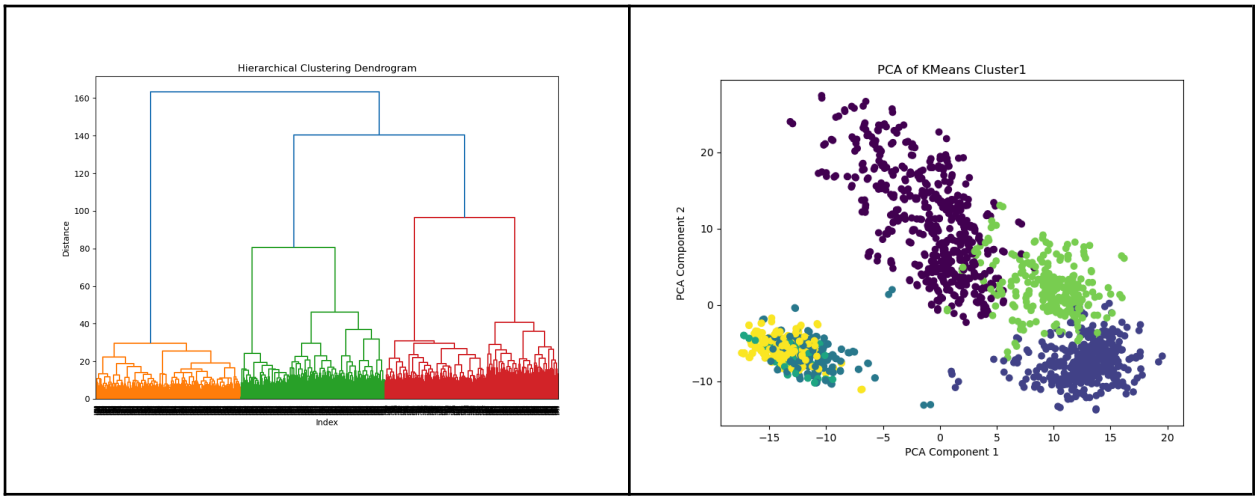
[33] Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*, 2023.

[34] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.

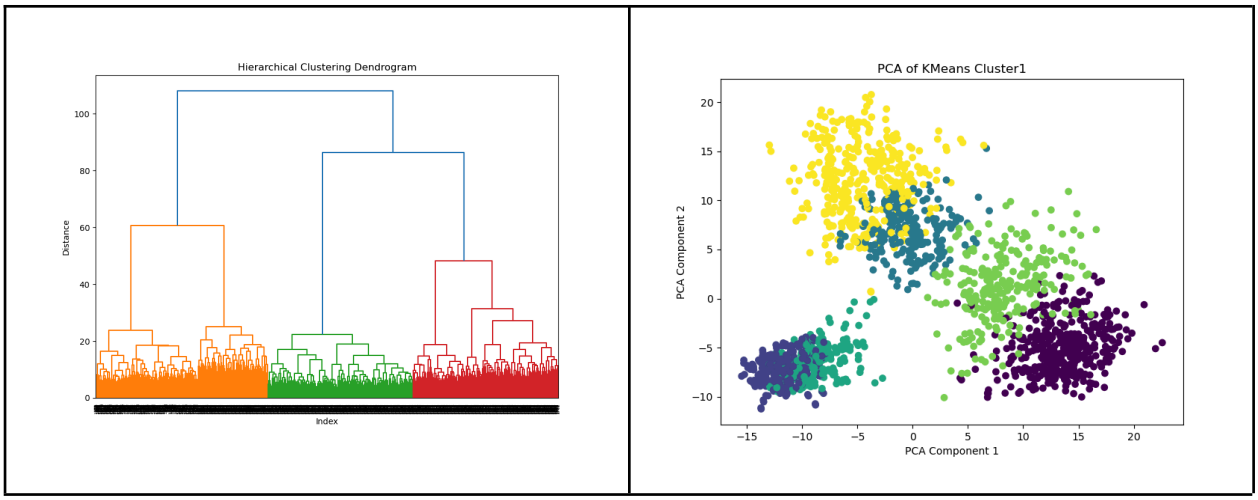
[35] Radim Řehůřek. Word2vec tutorial, 2025.

A Additional Figures and Tables

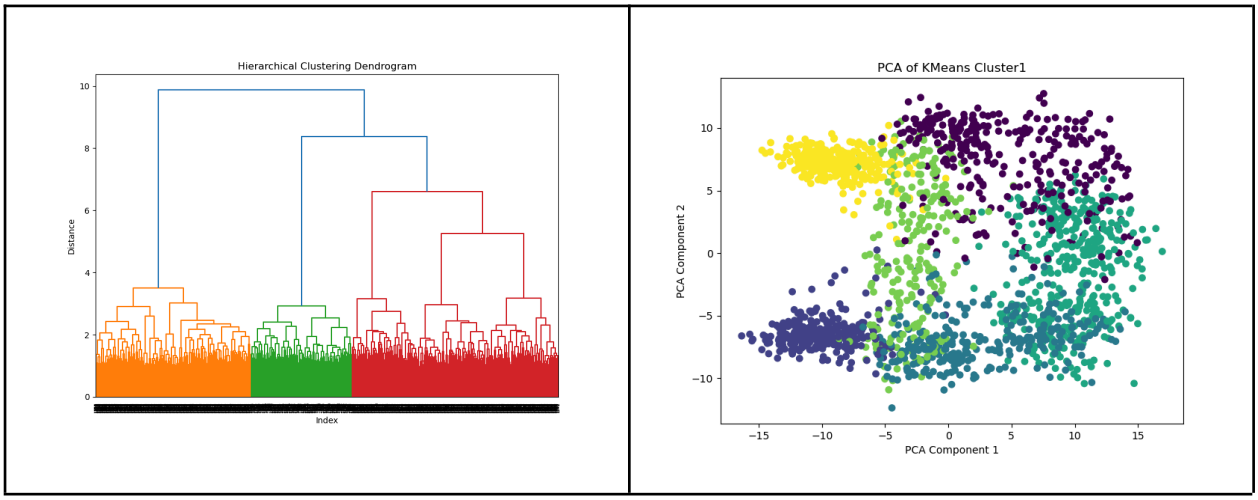
GEN DATASET



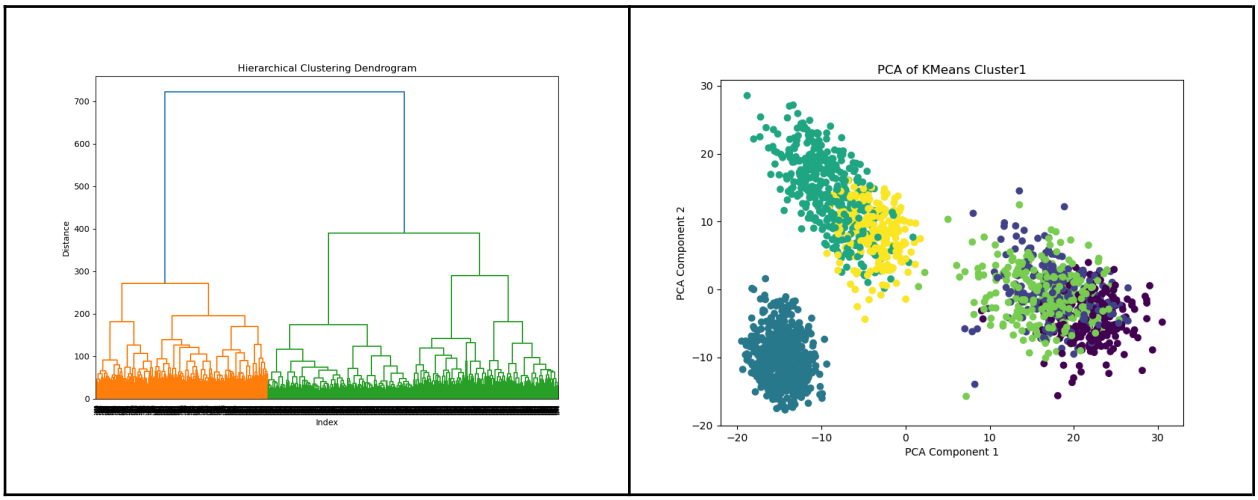
ALBERT



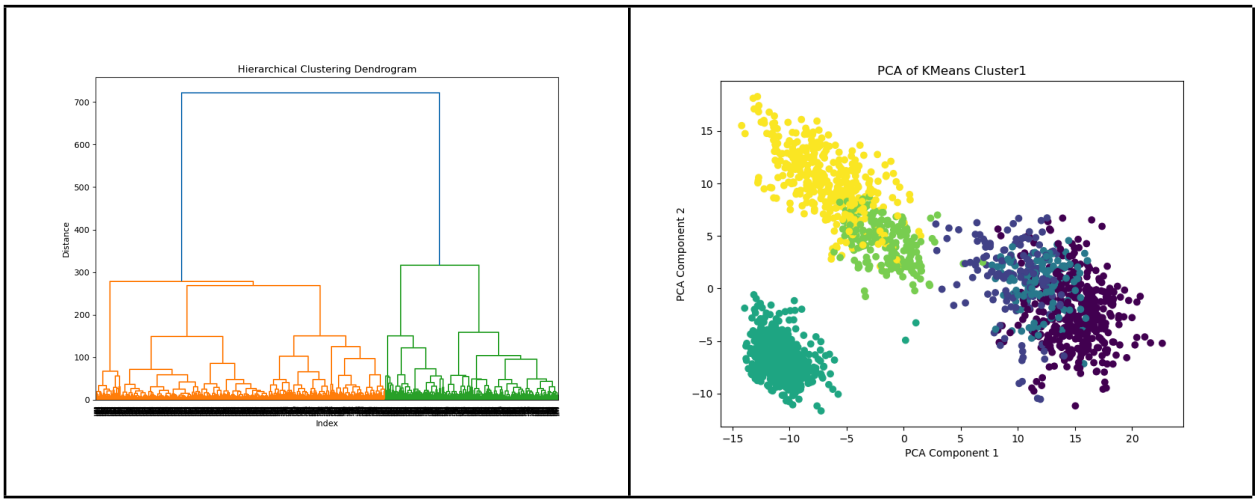
BERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

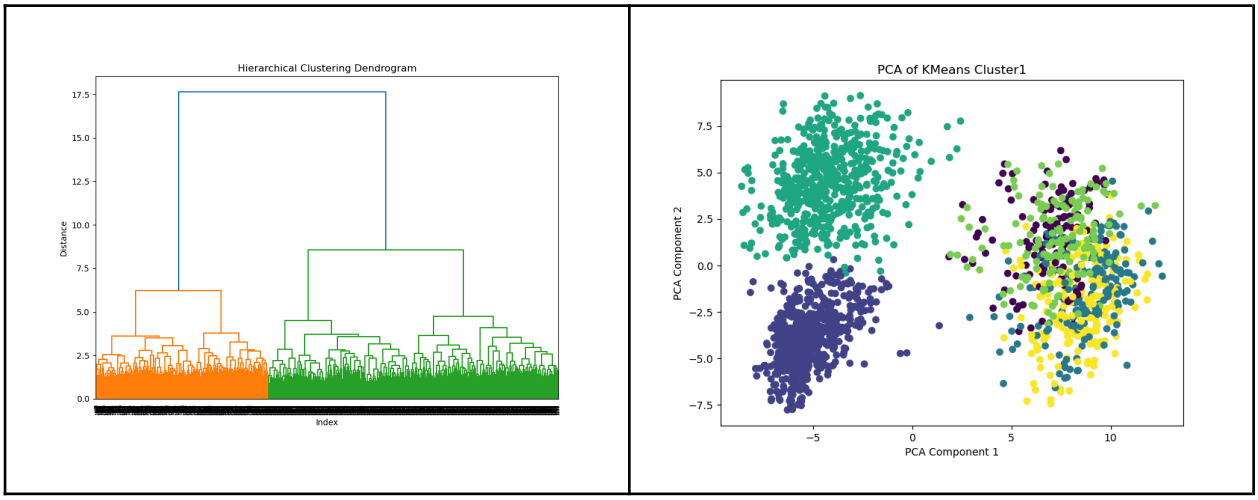
BGE


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

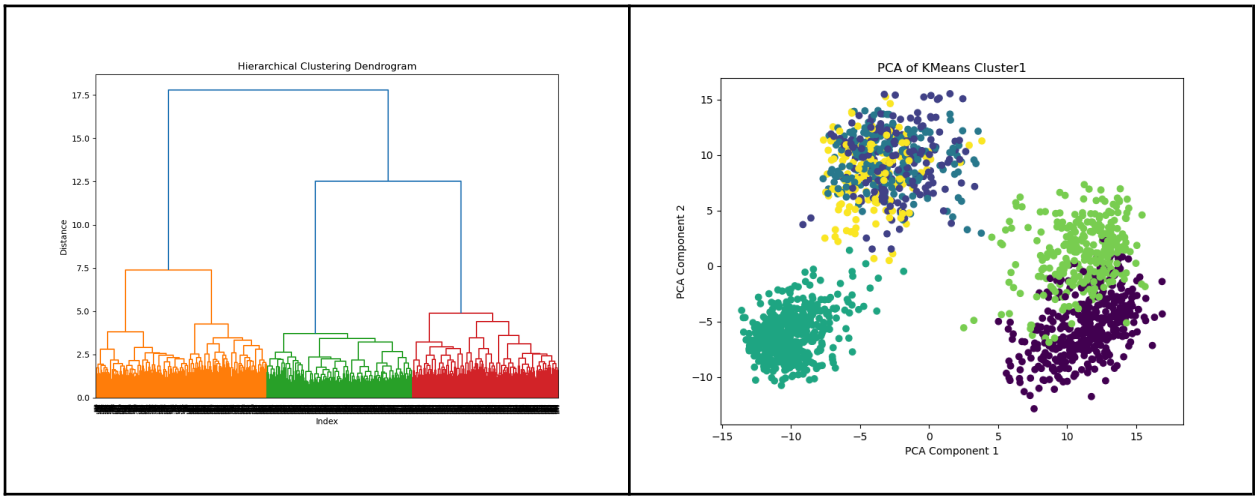
DEEP


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

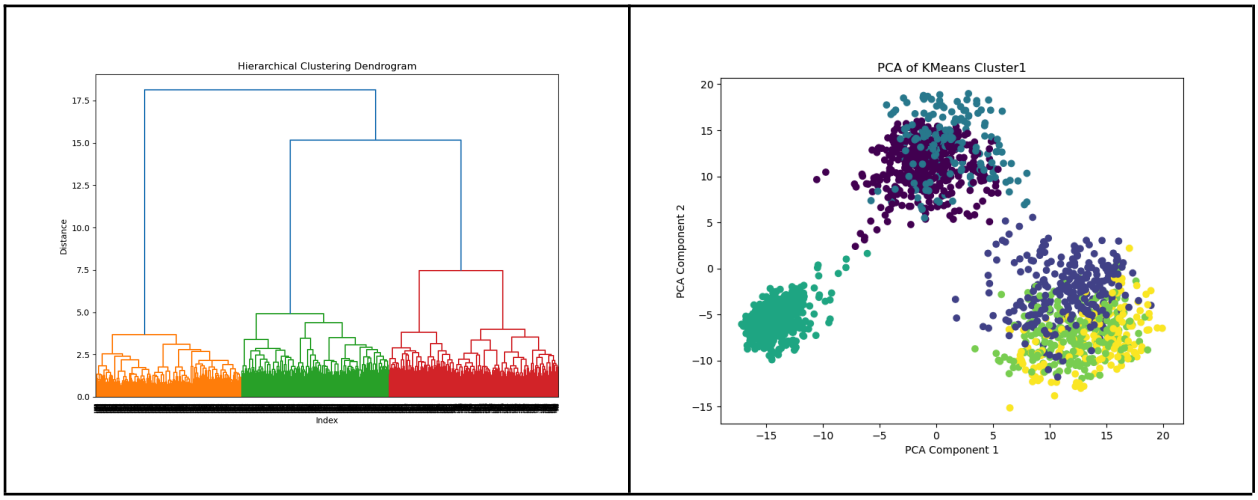
GPT2



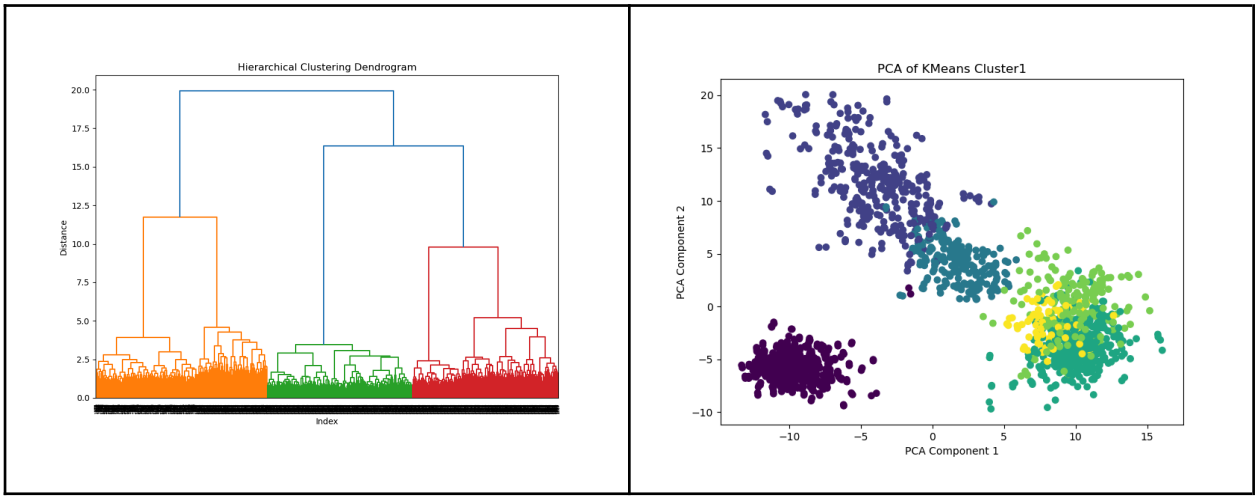
MINI



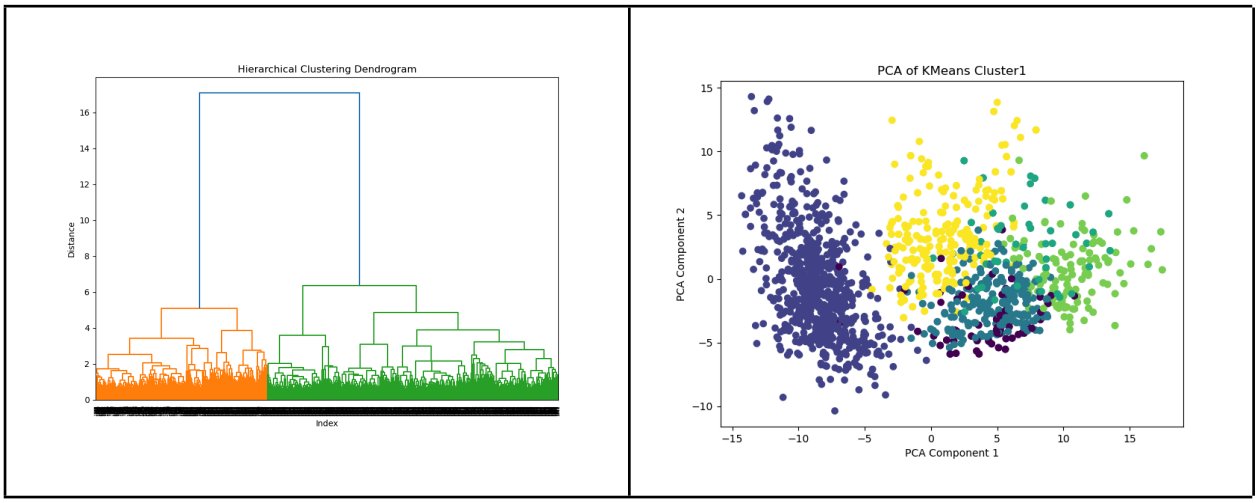
MPNET


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

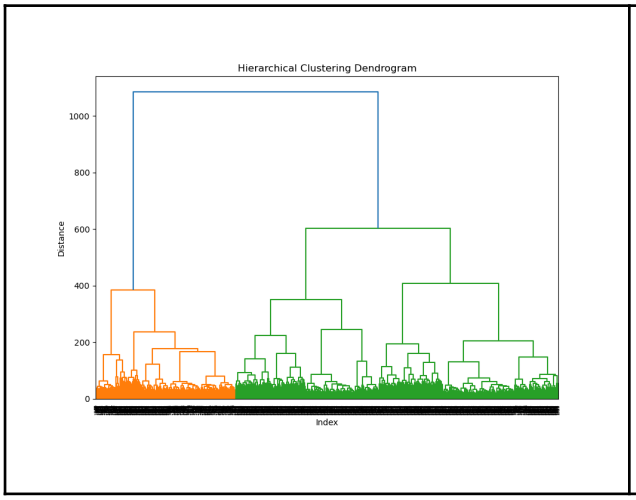
ROBERTA

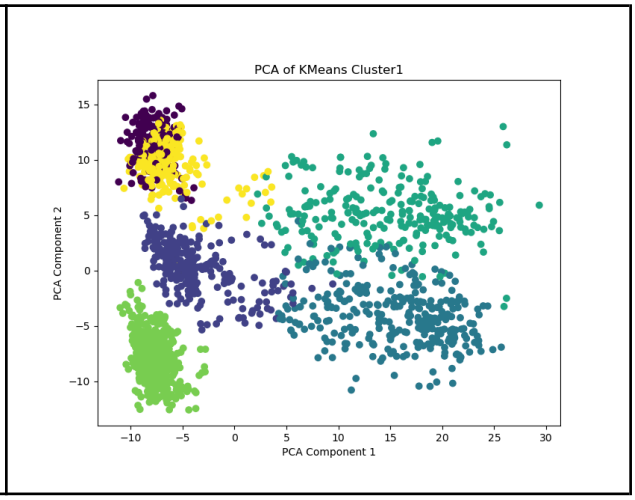

Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

T5


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

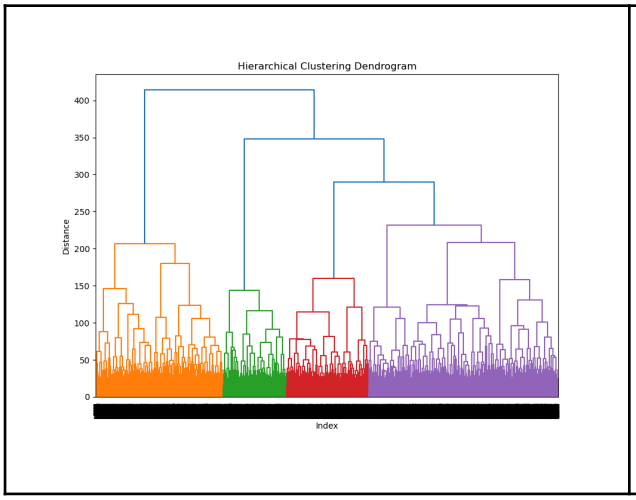
WORD2VEC

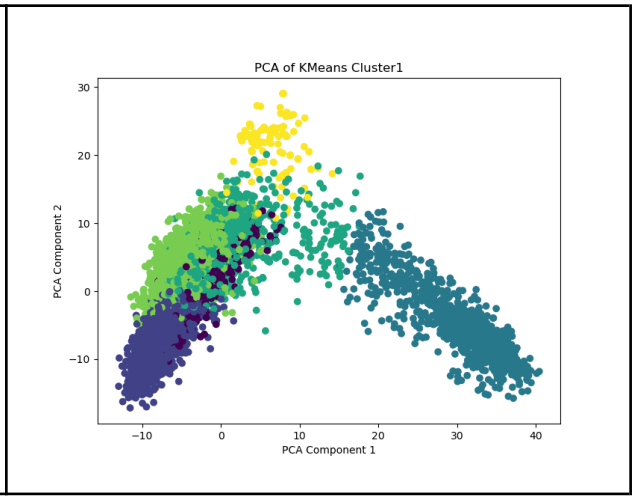




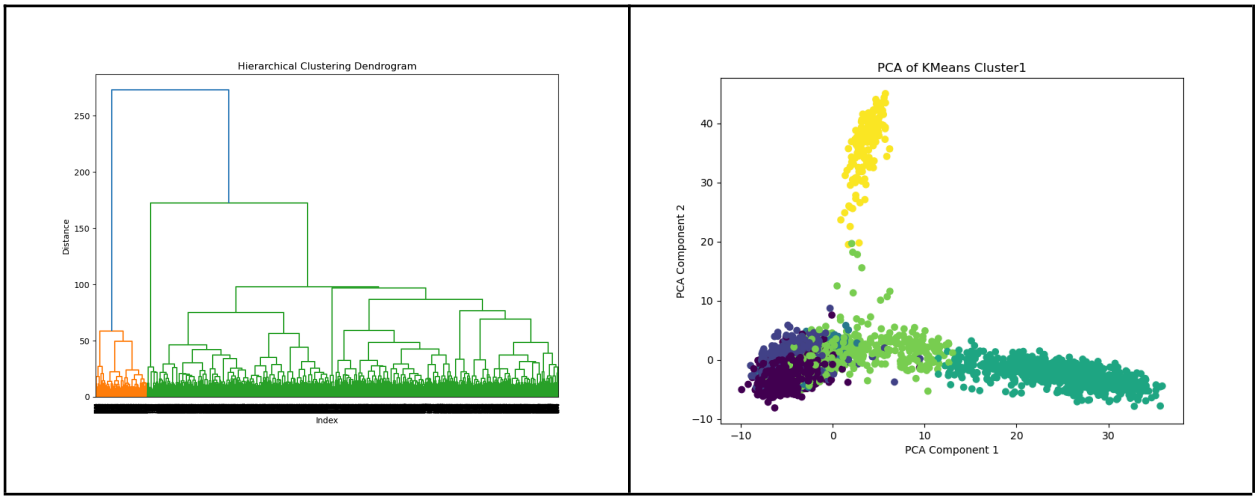
XLNET

MTEB DATASET

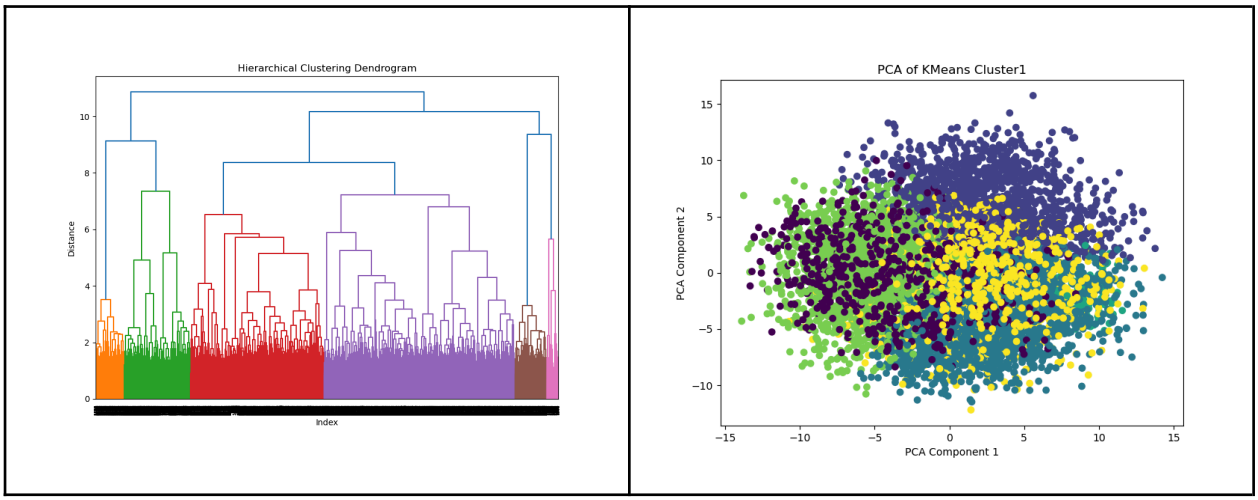




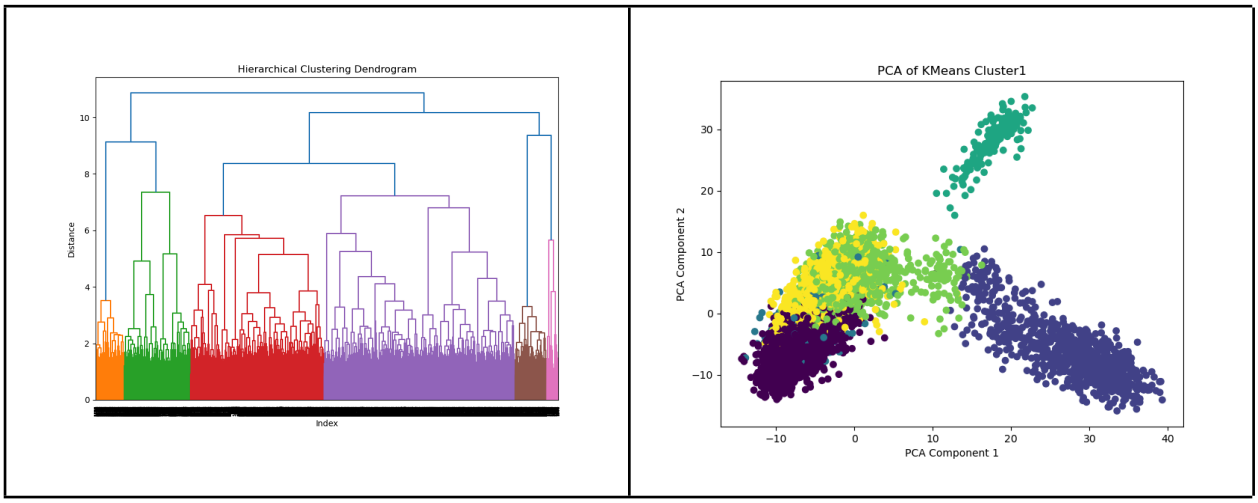
ALBERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

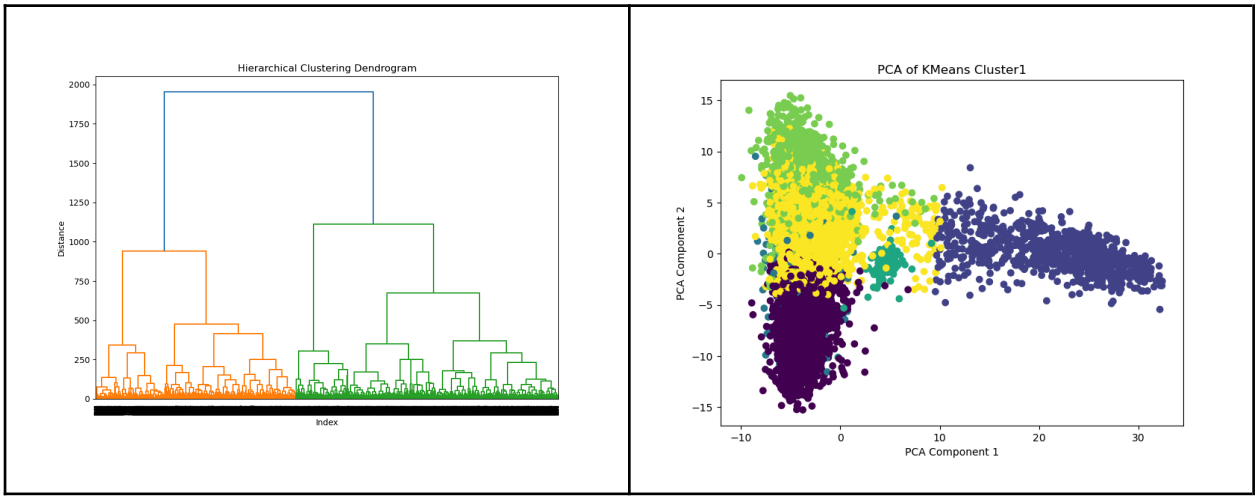
BERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

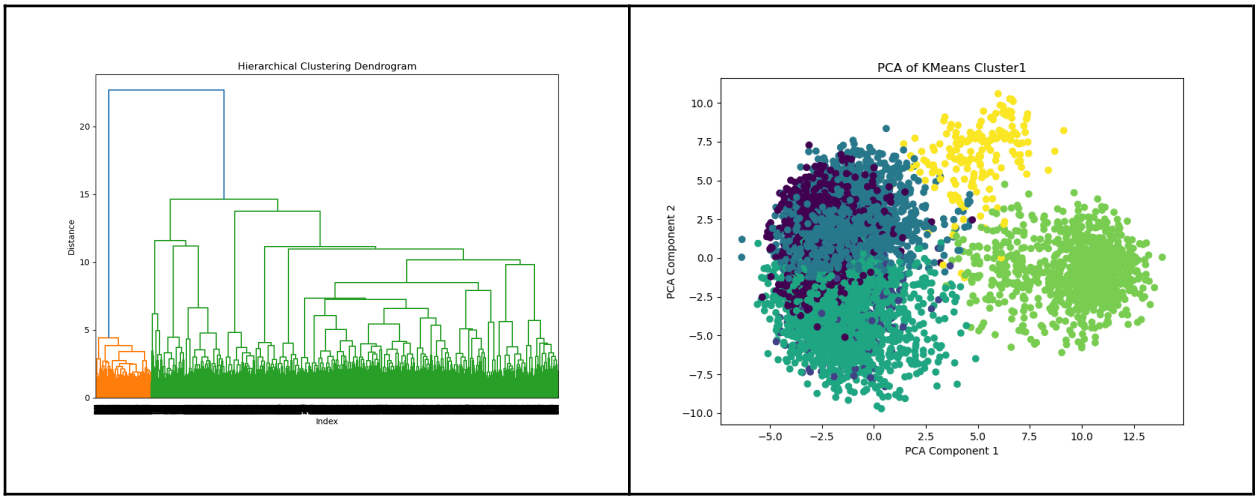
BGE


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

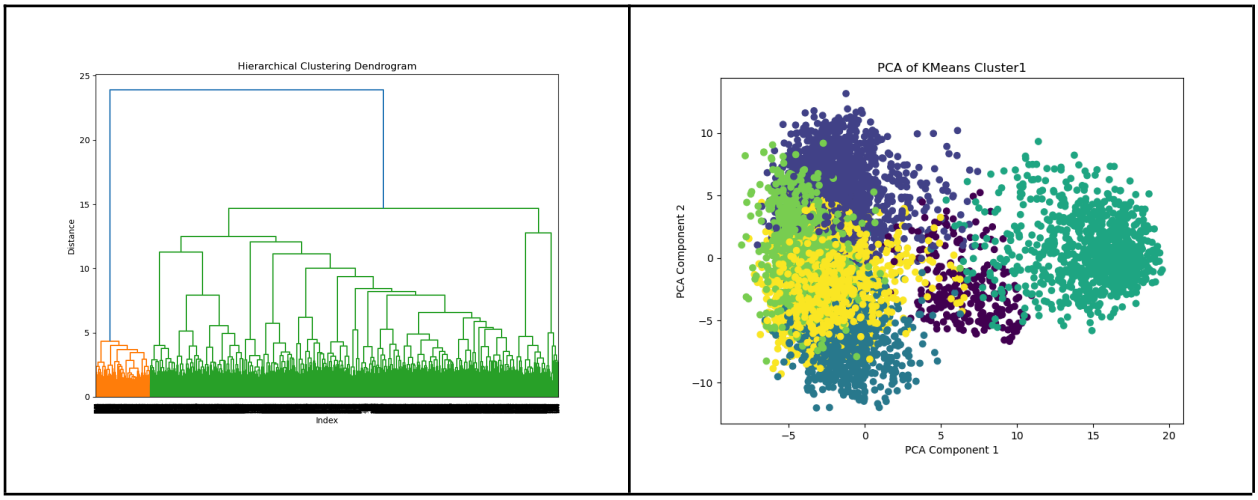
DEEP



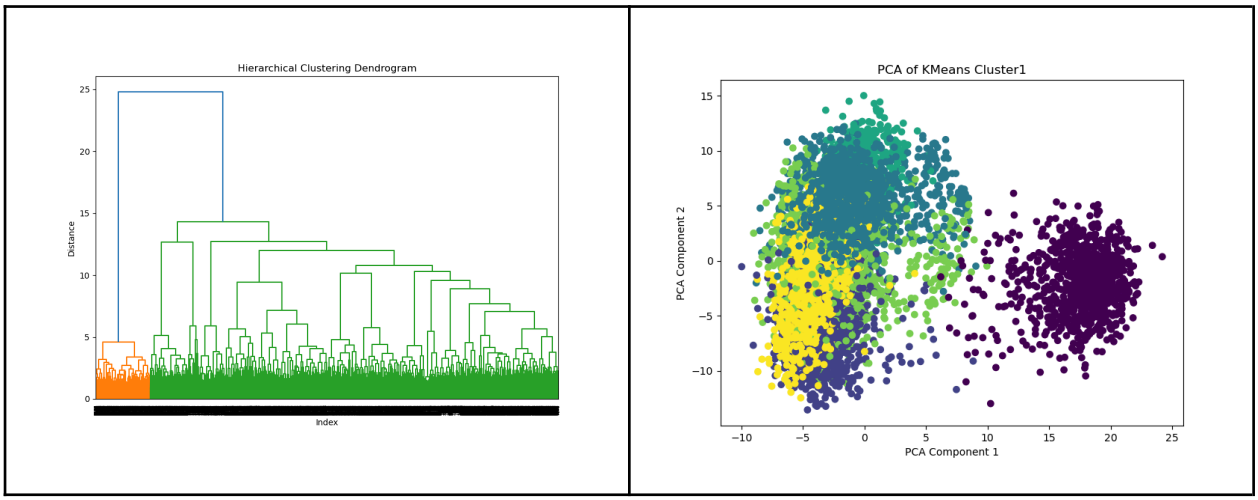
GPT2



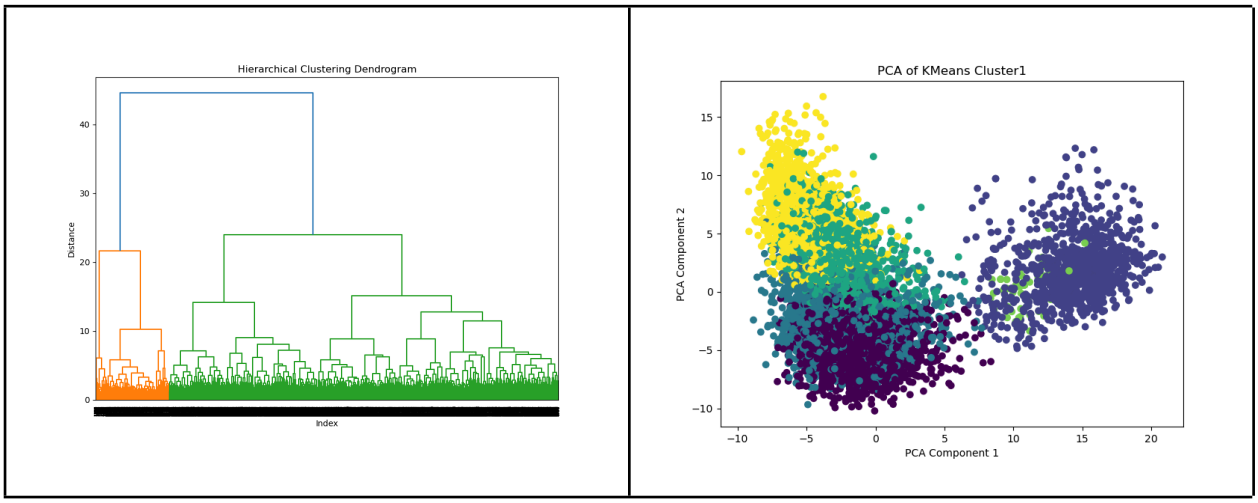
MINI



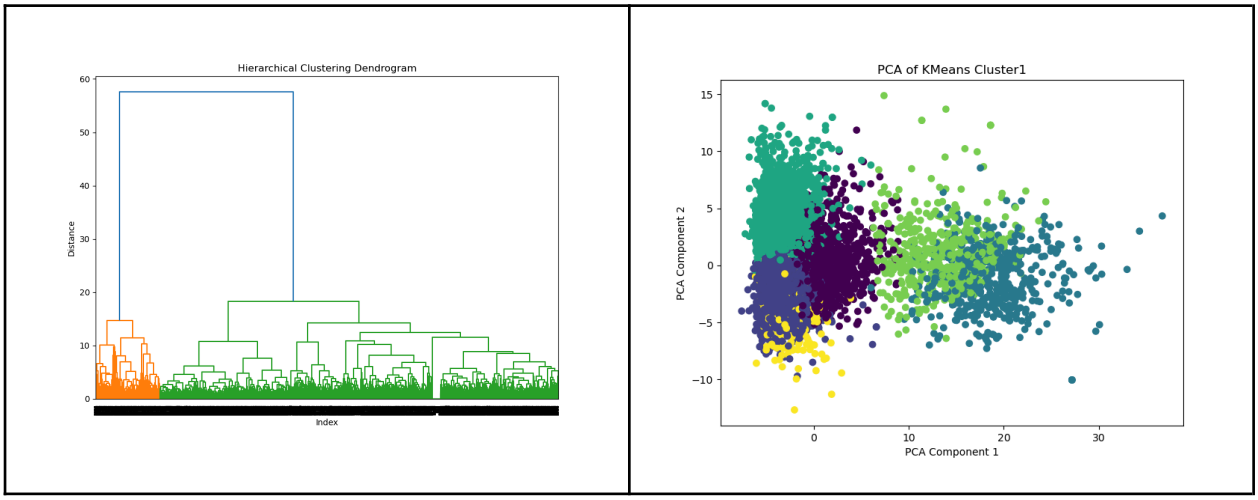
MPNET



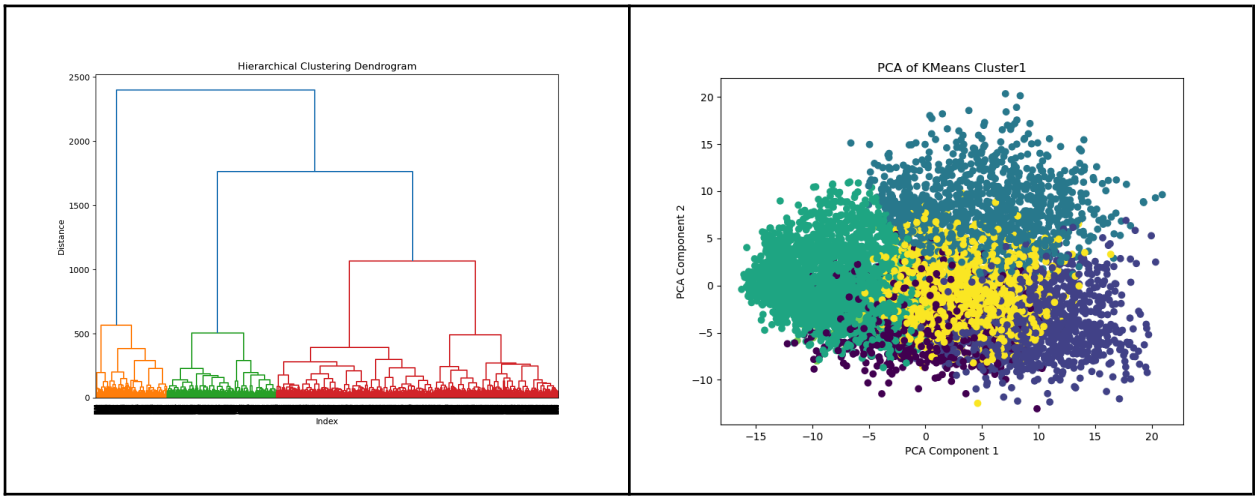
ROBERTA



T5

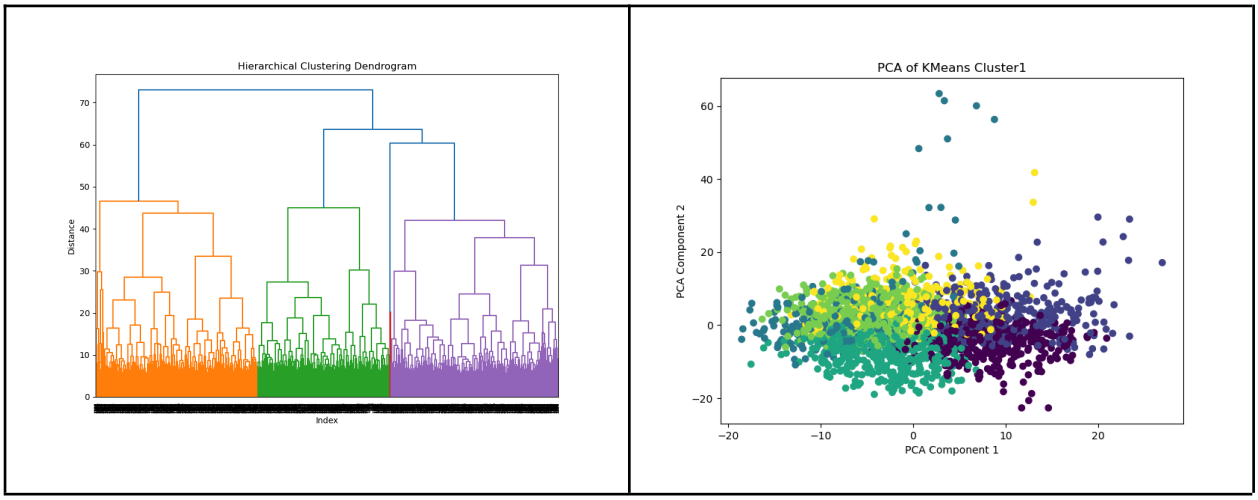


WORD2VEC

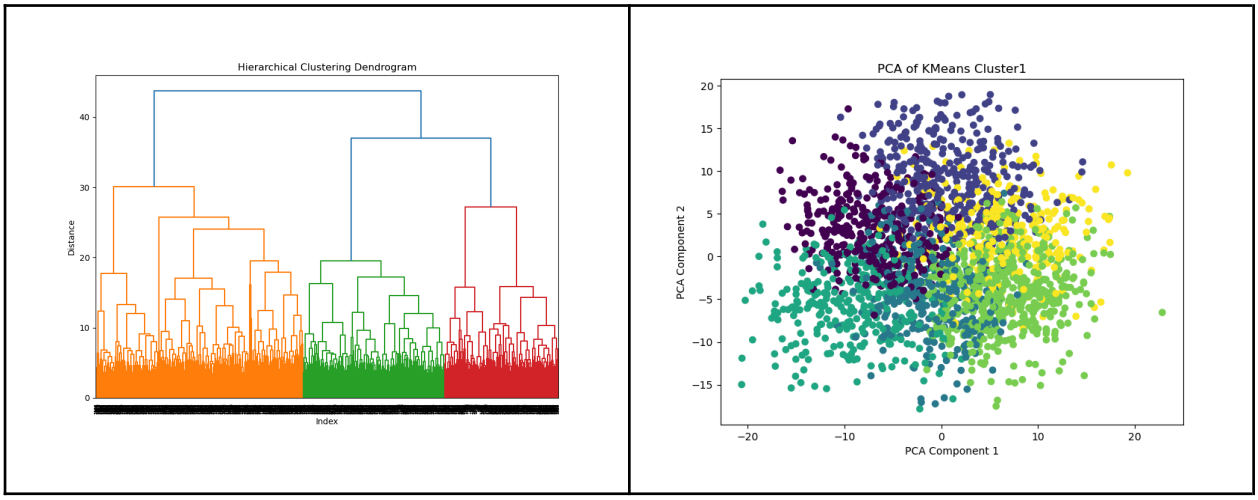


XLNET

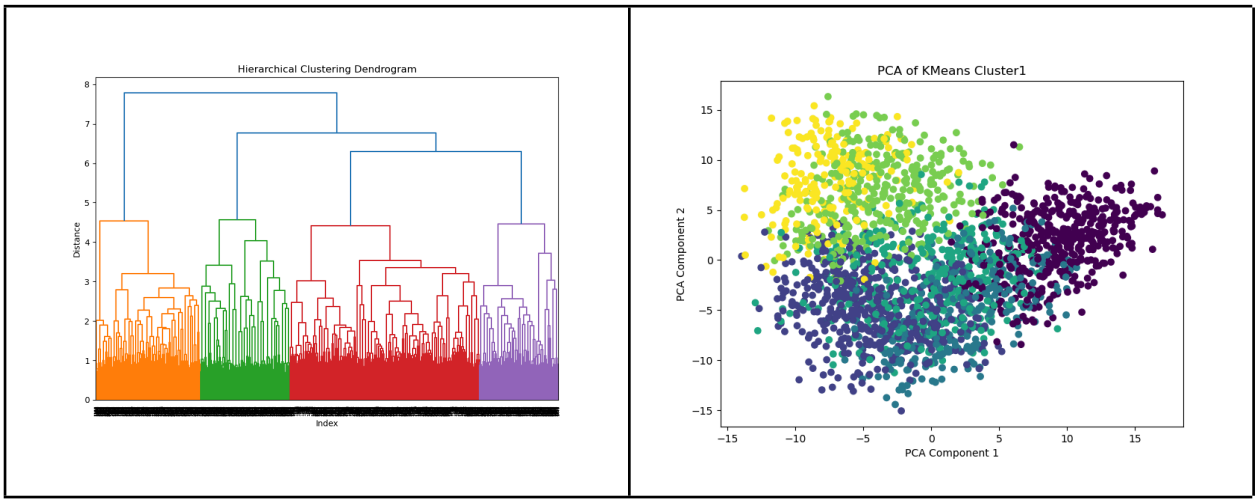
WILD DATASET


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

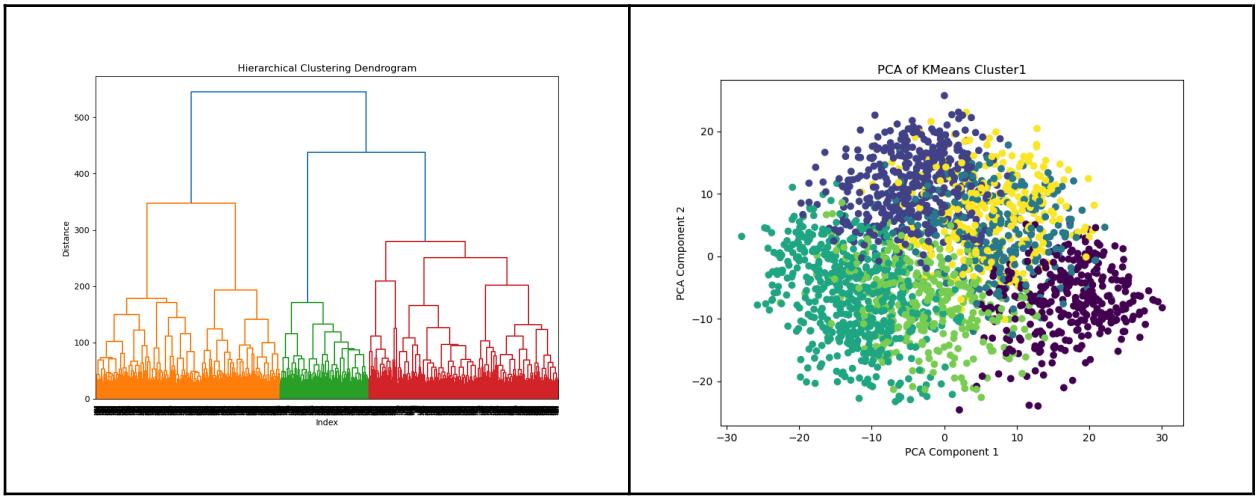
ALBERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

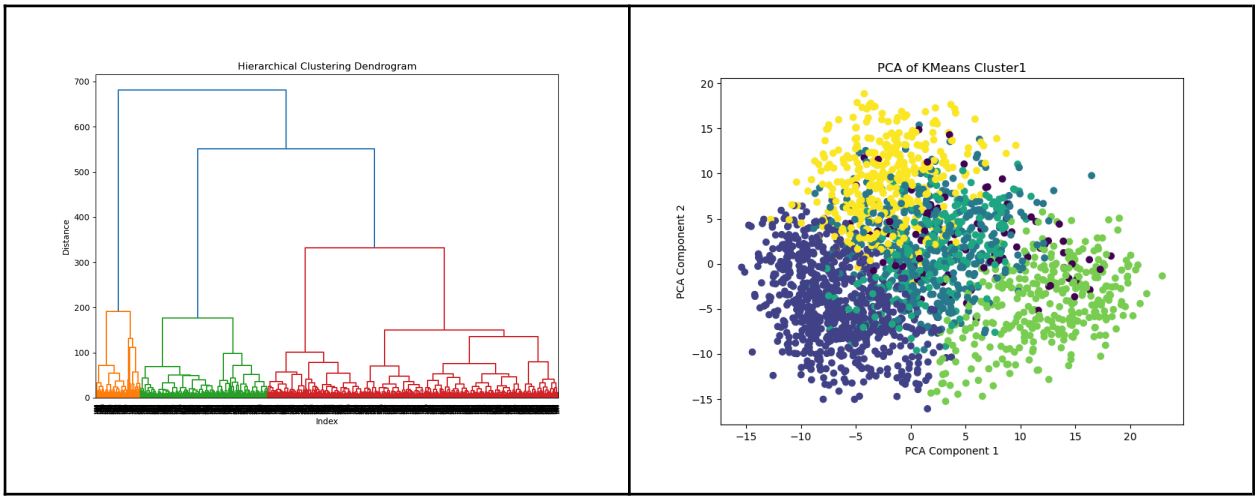
BERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

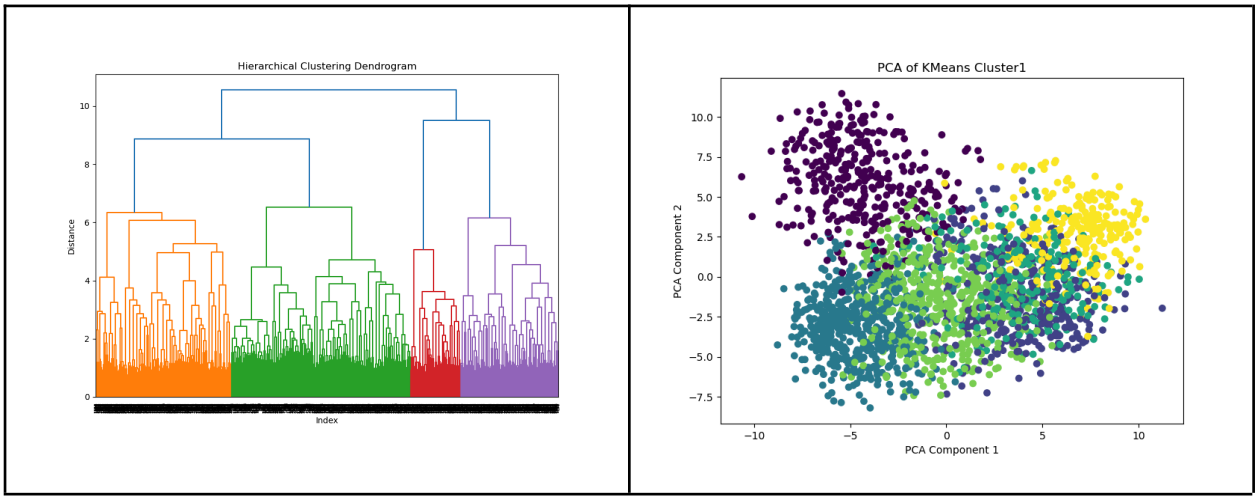
BGE



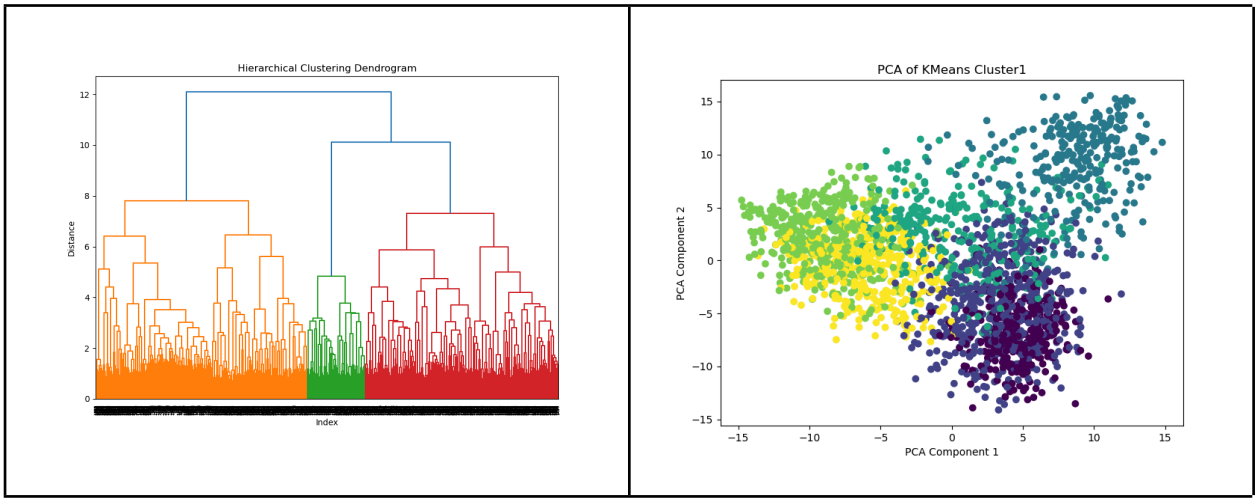
DEEP



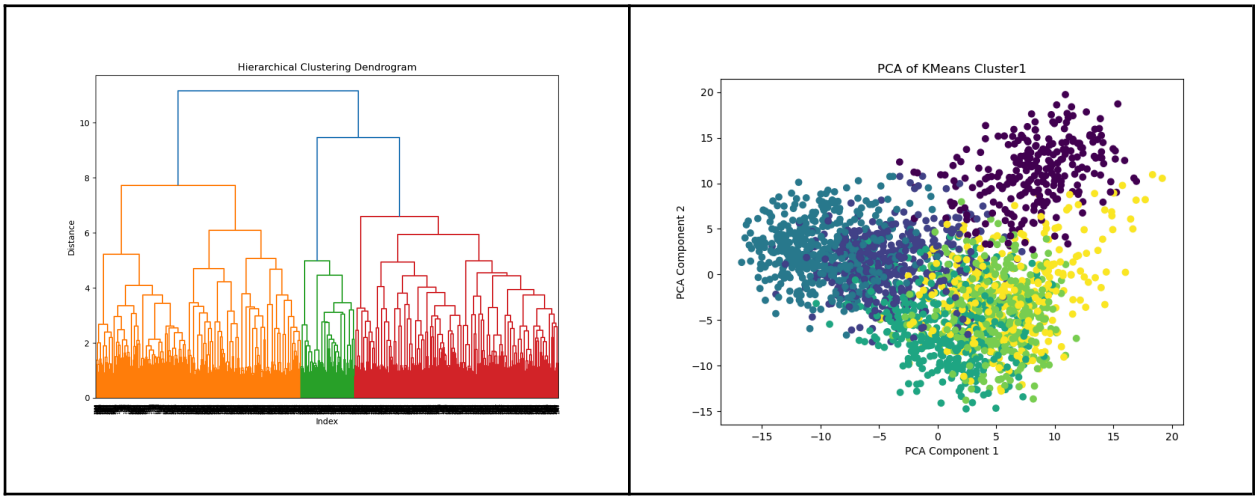
GPT2


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

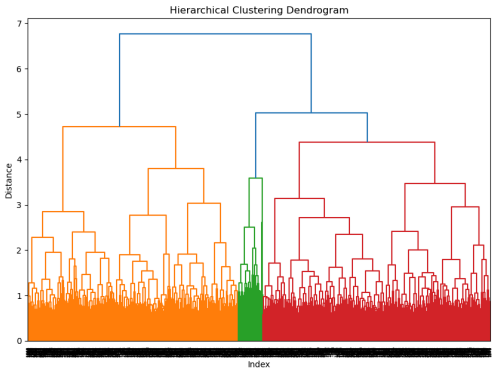
MINI

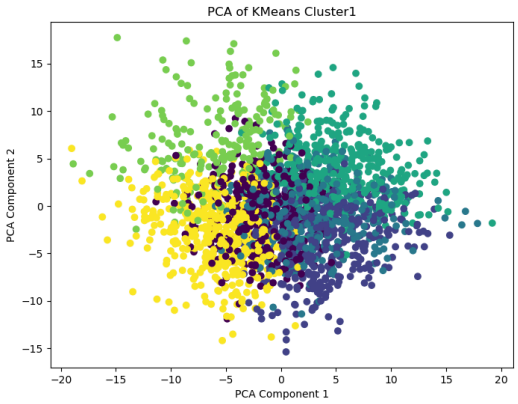

Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

MPNET

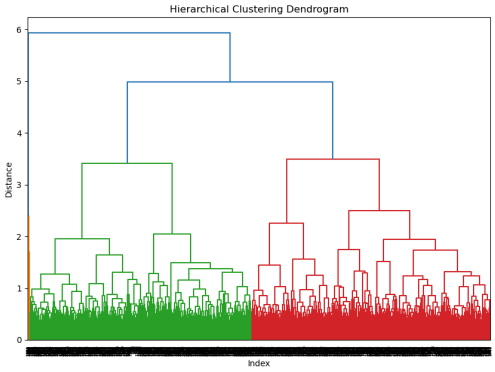

Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

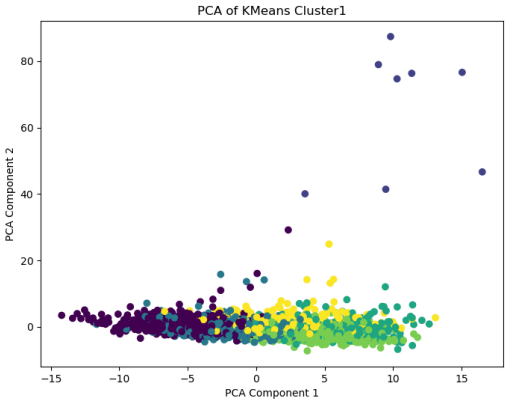
ROBERTA



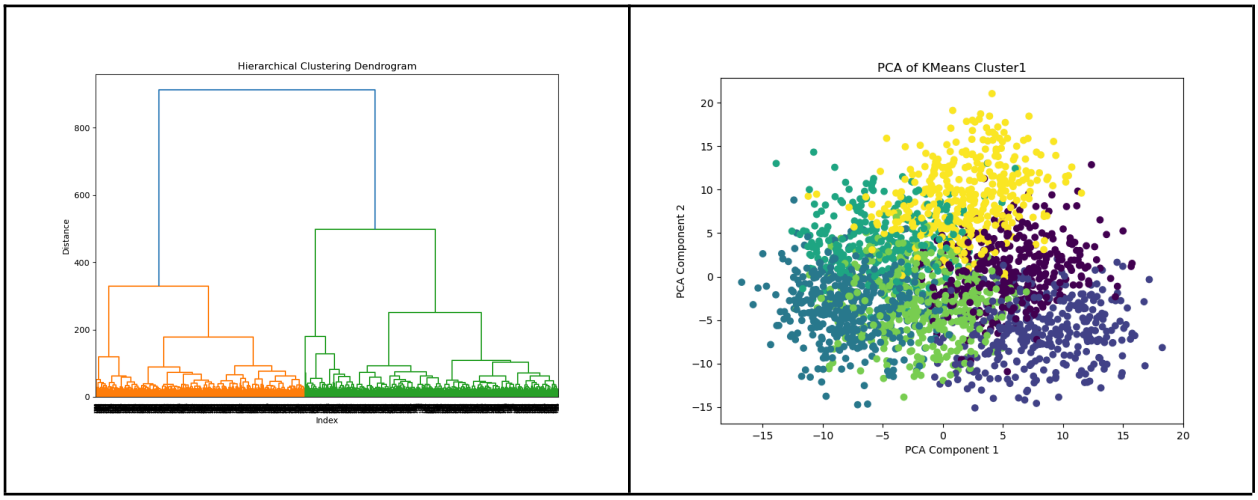


T5



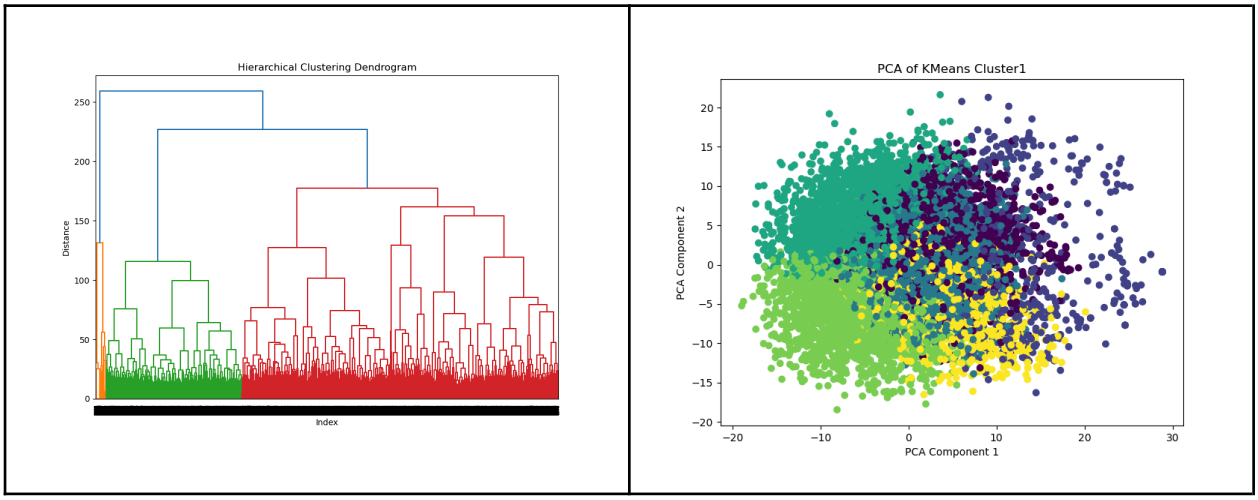


WORD2VEC

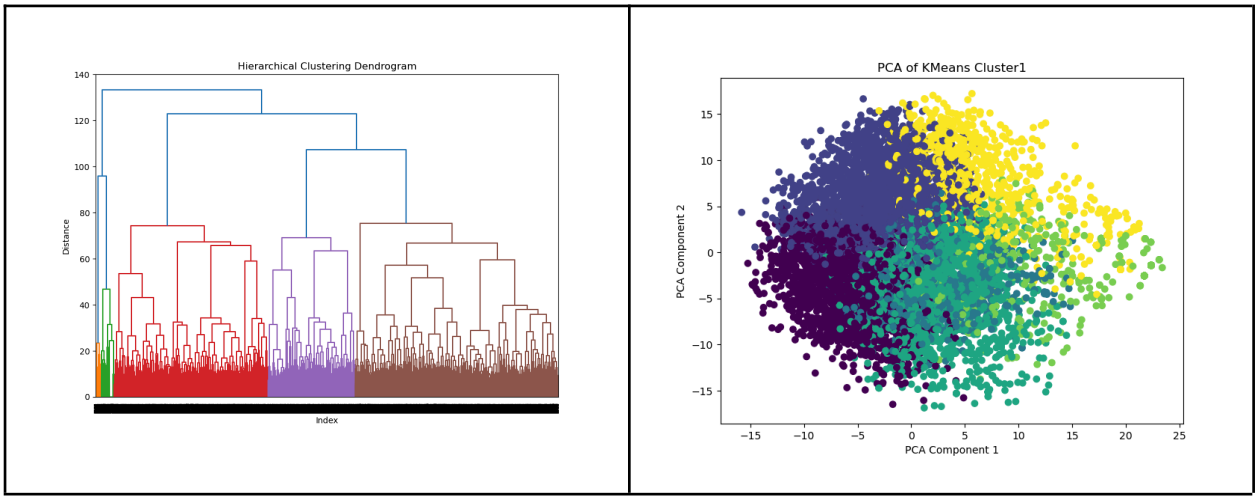


XLNET

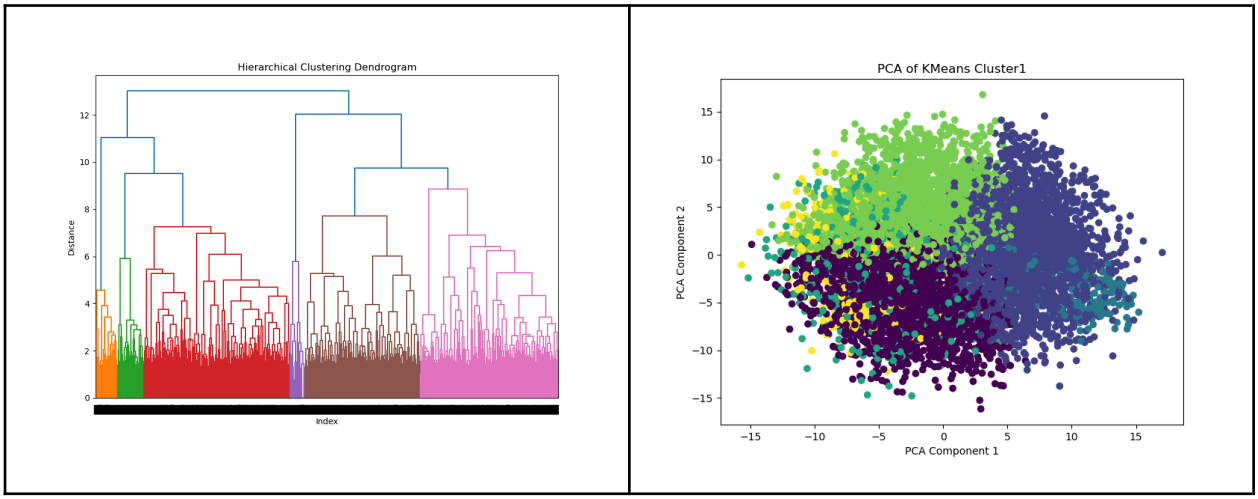
PRISM DATASET



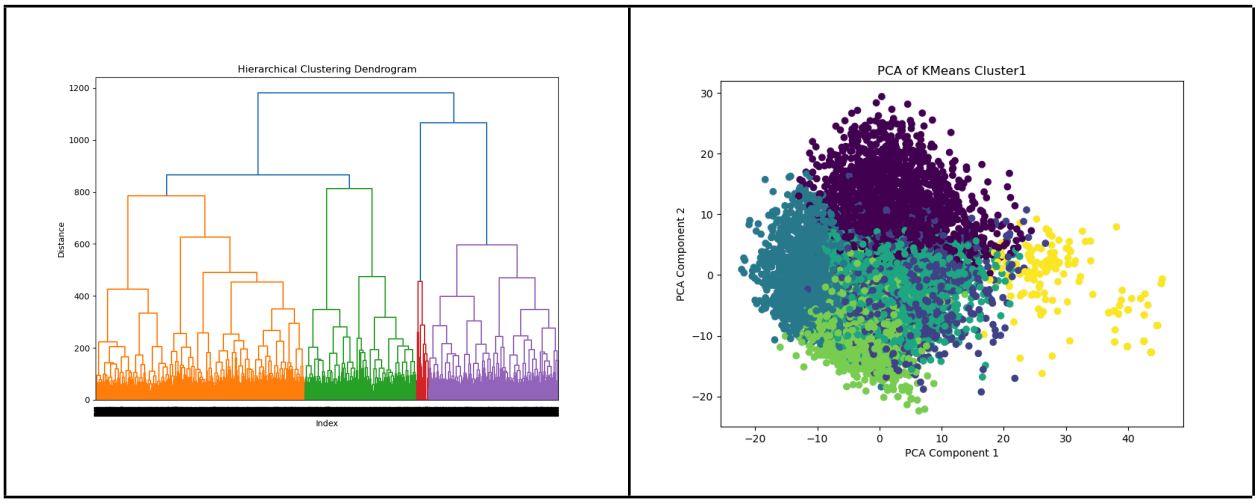
ALBERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

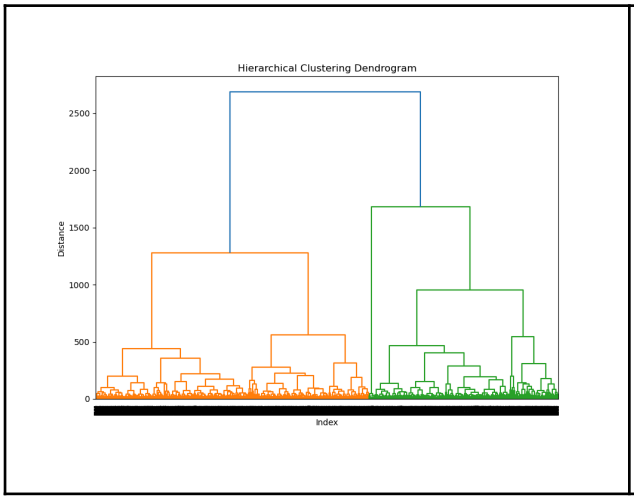
BERT

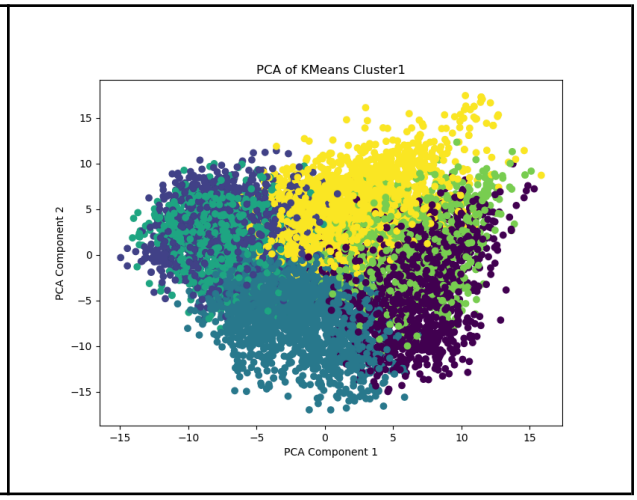

Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

BGE

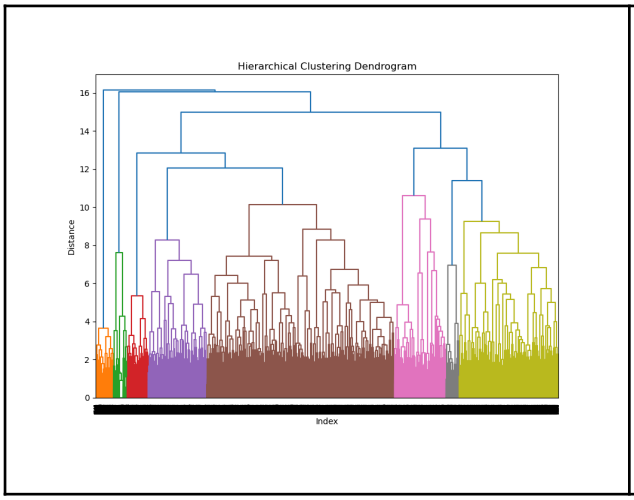

Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

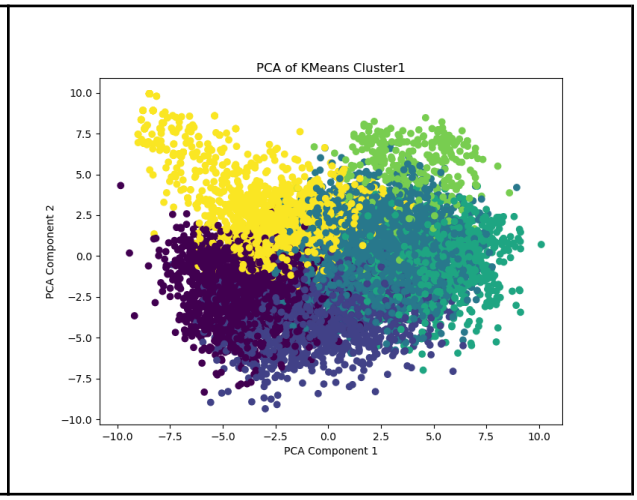
DEEP



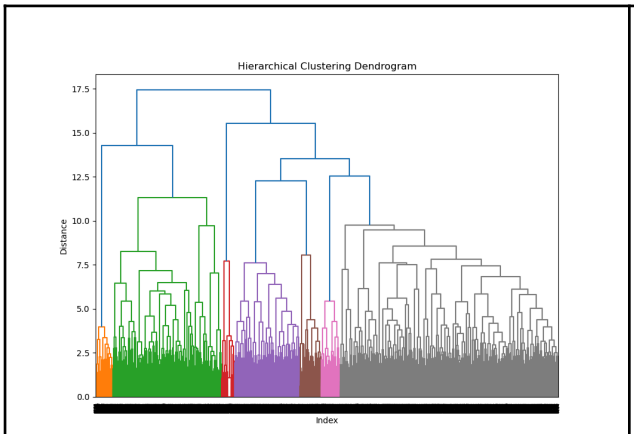


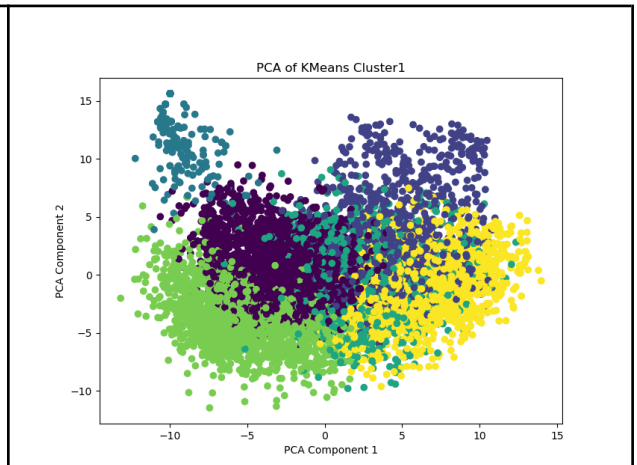
GPT2



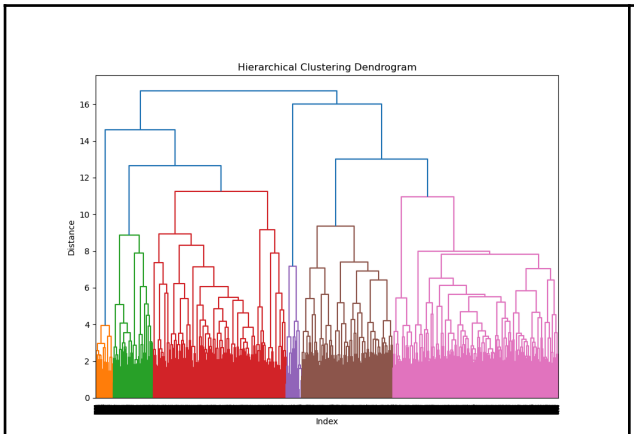


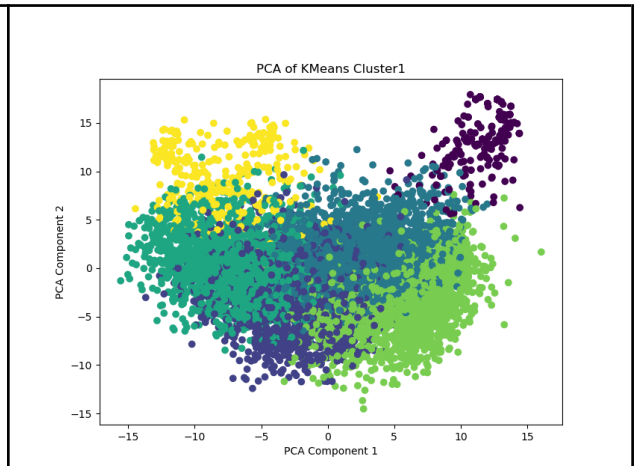
MINI



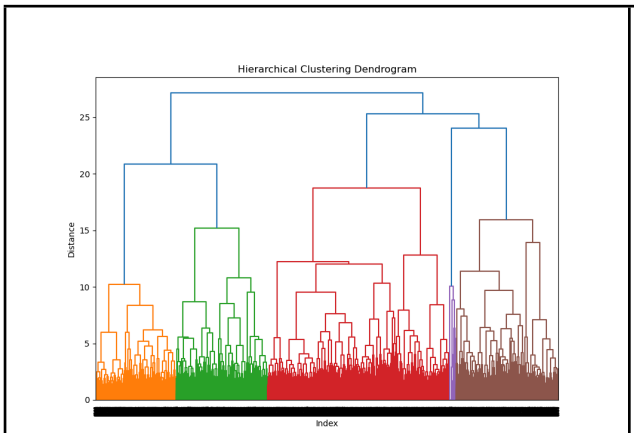


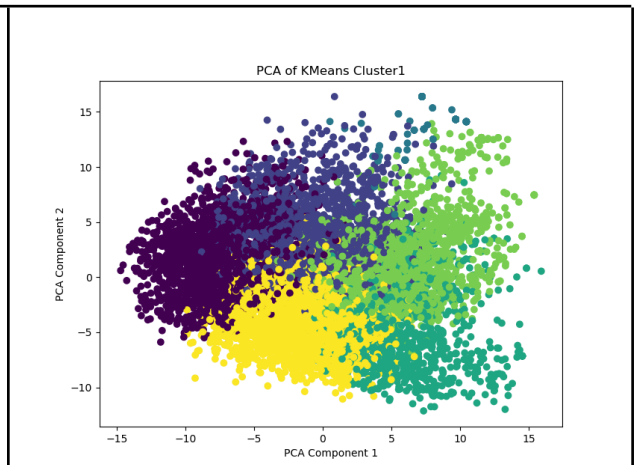
MPNET



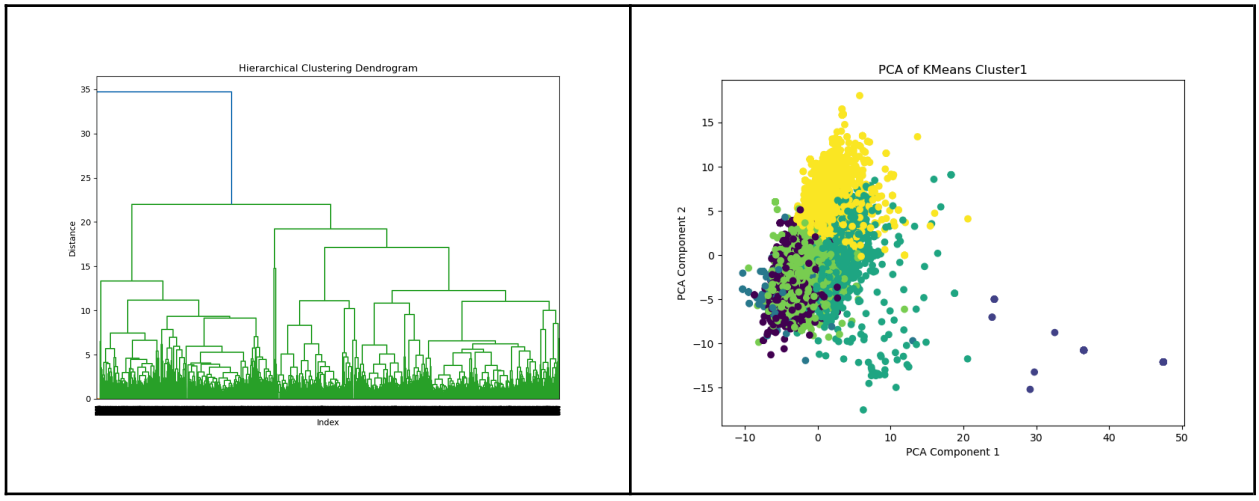


ROBERTA

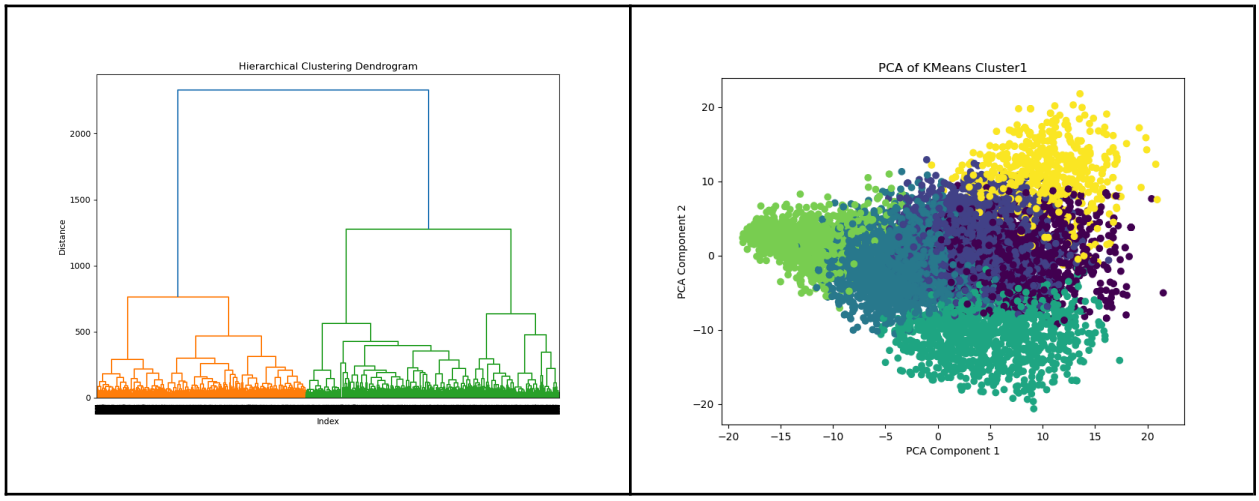




T5

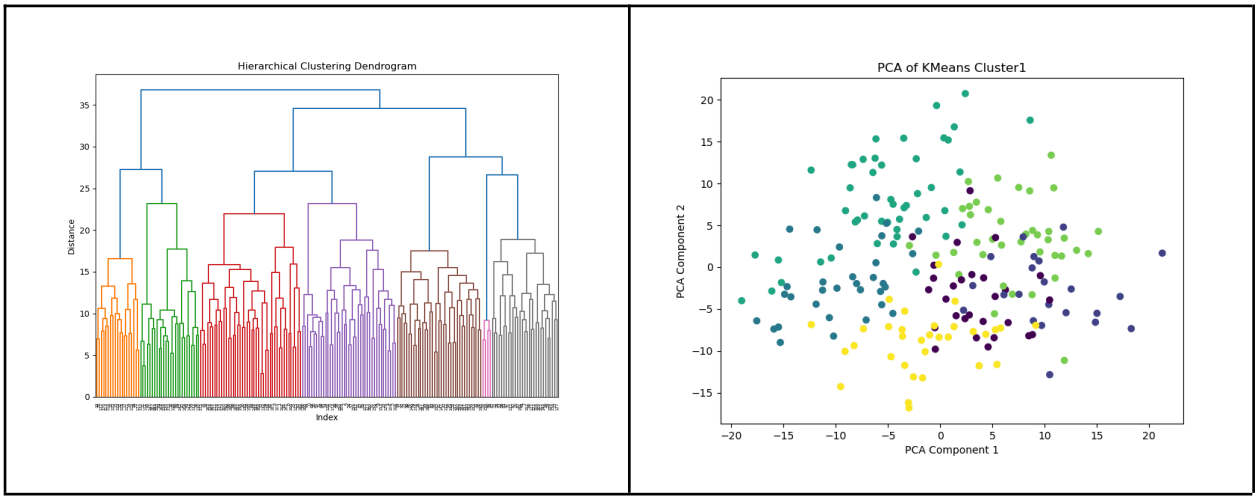


WORD2VEC

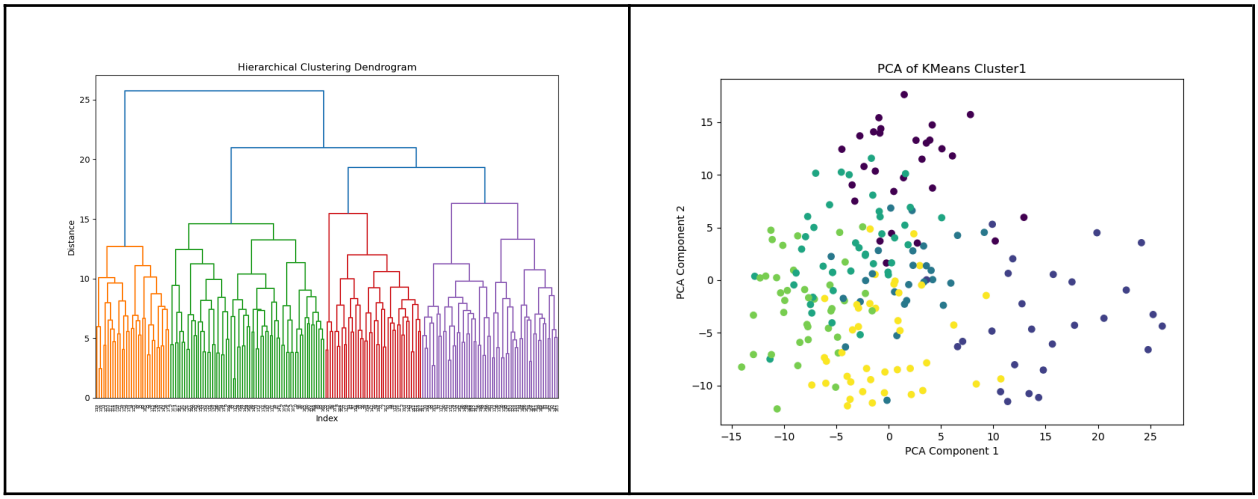


XLNET

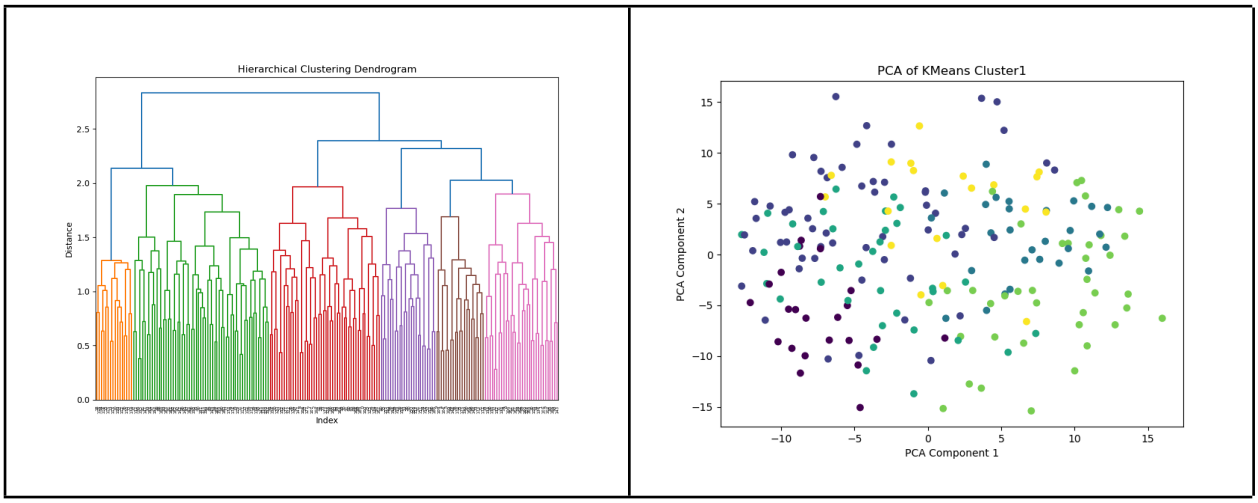
JBBBehavior DATASET


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

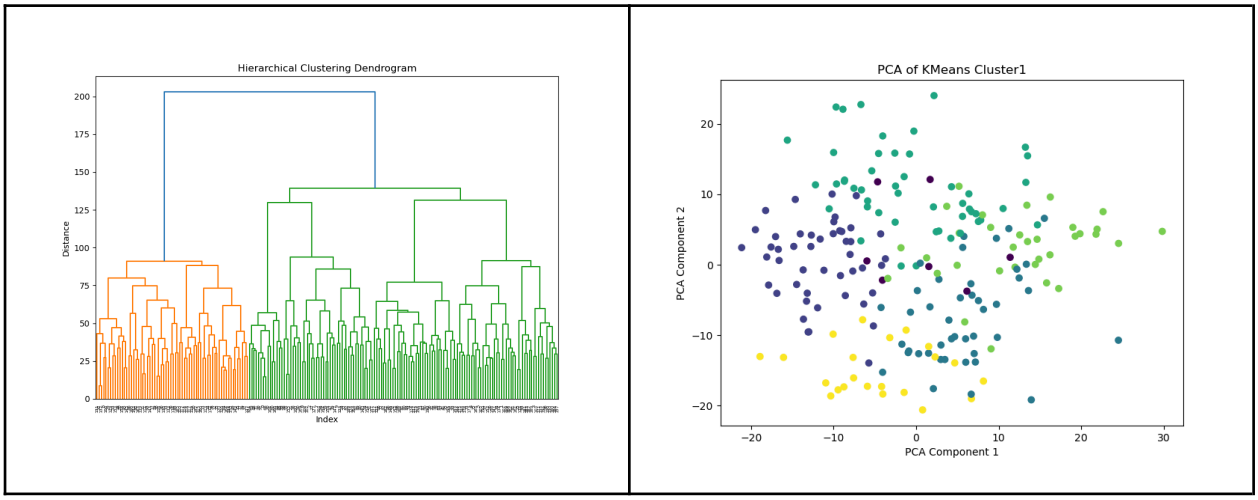
ALBERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

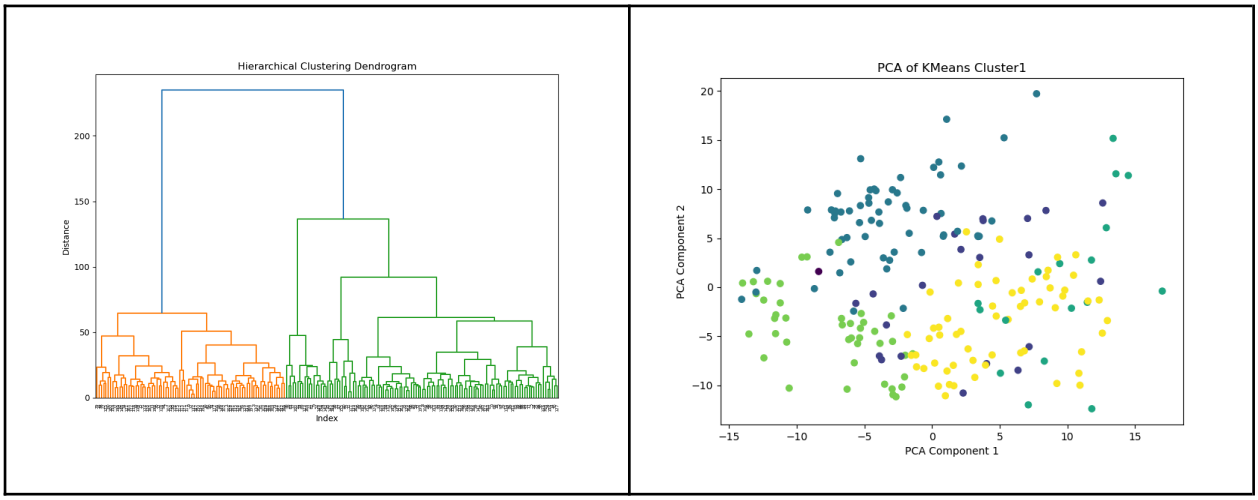
BERT


Hierarchical Clustering Dendrogram
Distance
Index
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

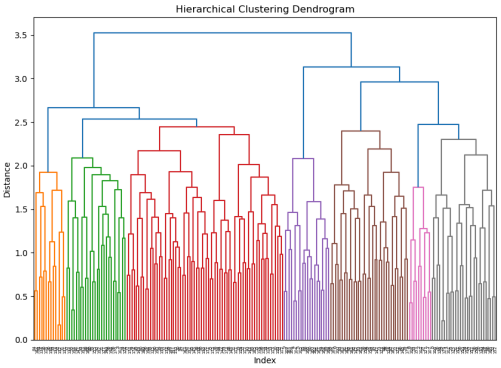
BGE

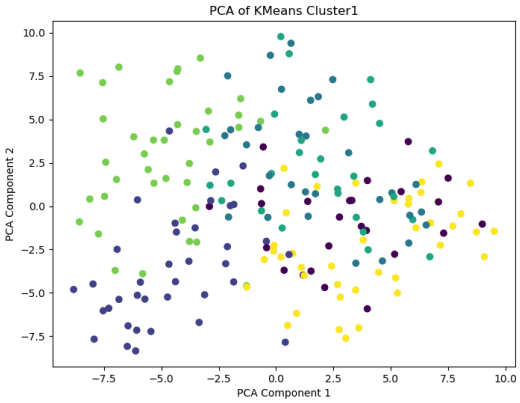


DEEP

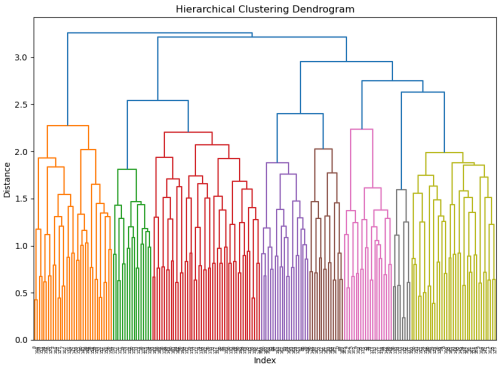


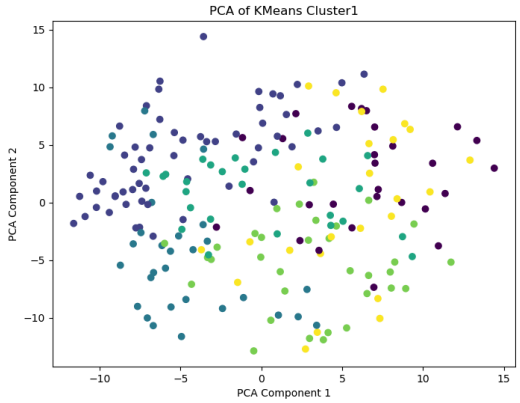
GPT2


Hierarchical Clustering Dendrogram
Distance
Index

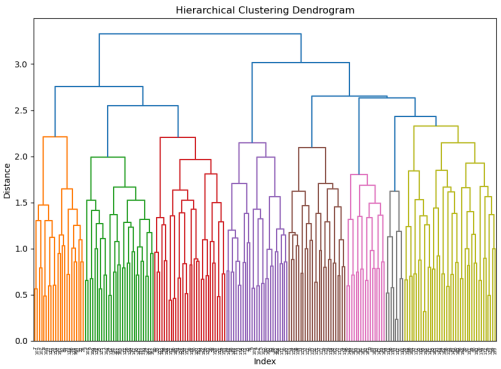

PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

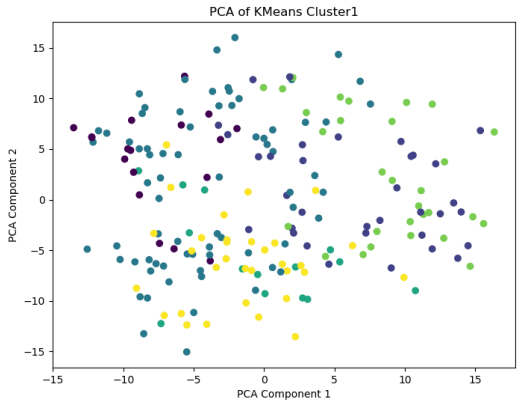
MINI


Hierarchical Clustering Dendrogram
Distance
Index

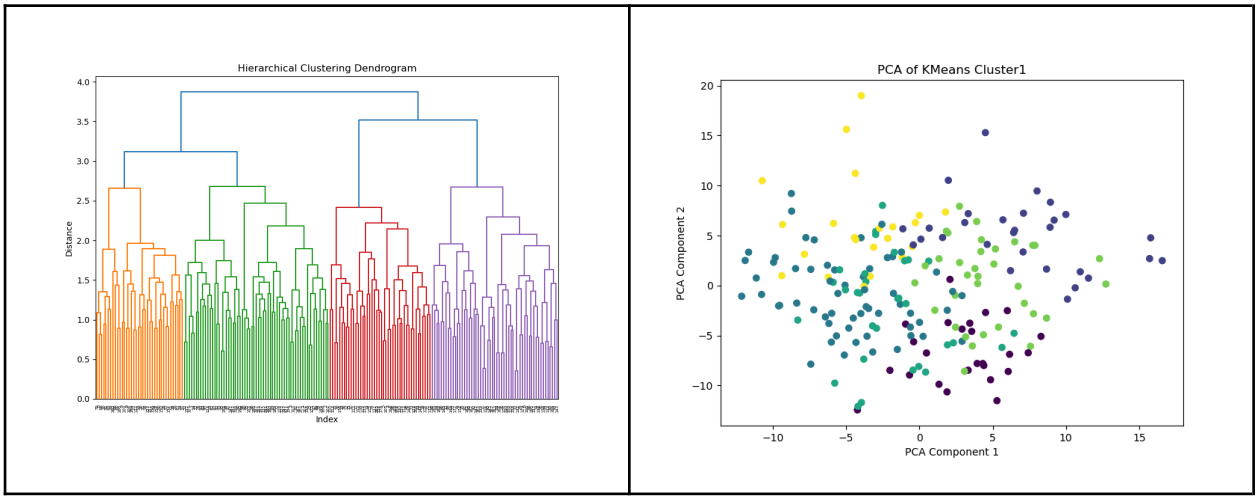

PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

MPNET

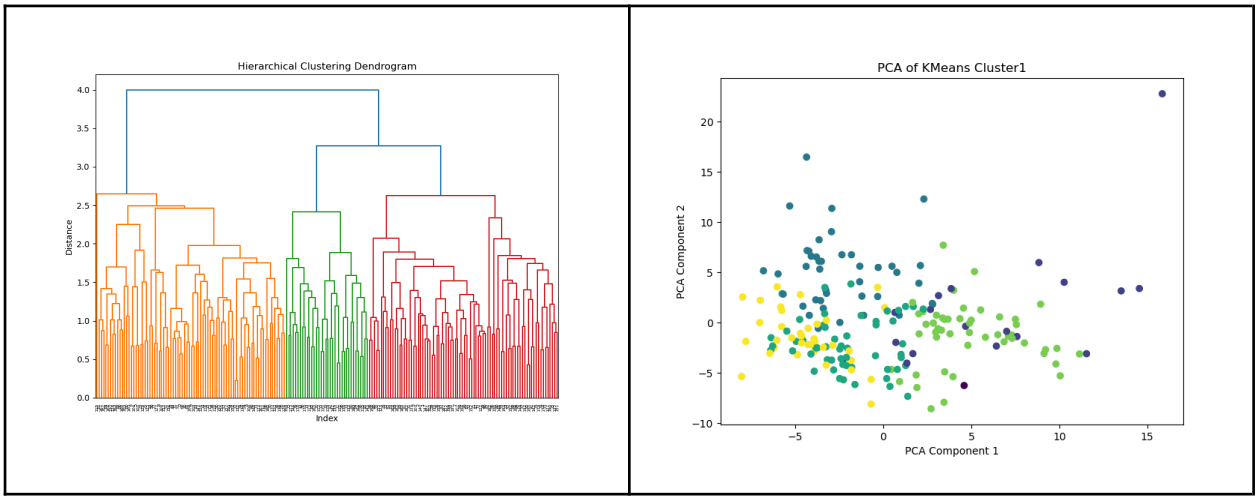

Hierarchical Clustering Dendrogram
Distance
Index


PCA of KMeans Cluster1
PCA Component 2
PCA Component 1

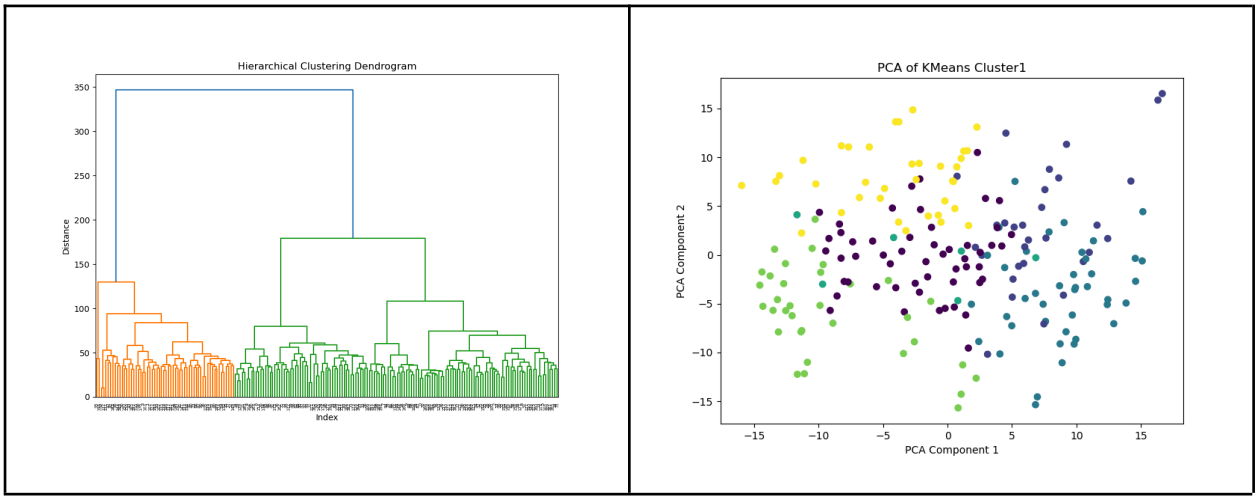
ROBERTA



T5



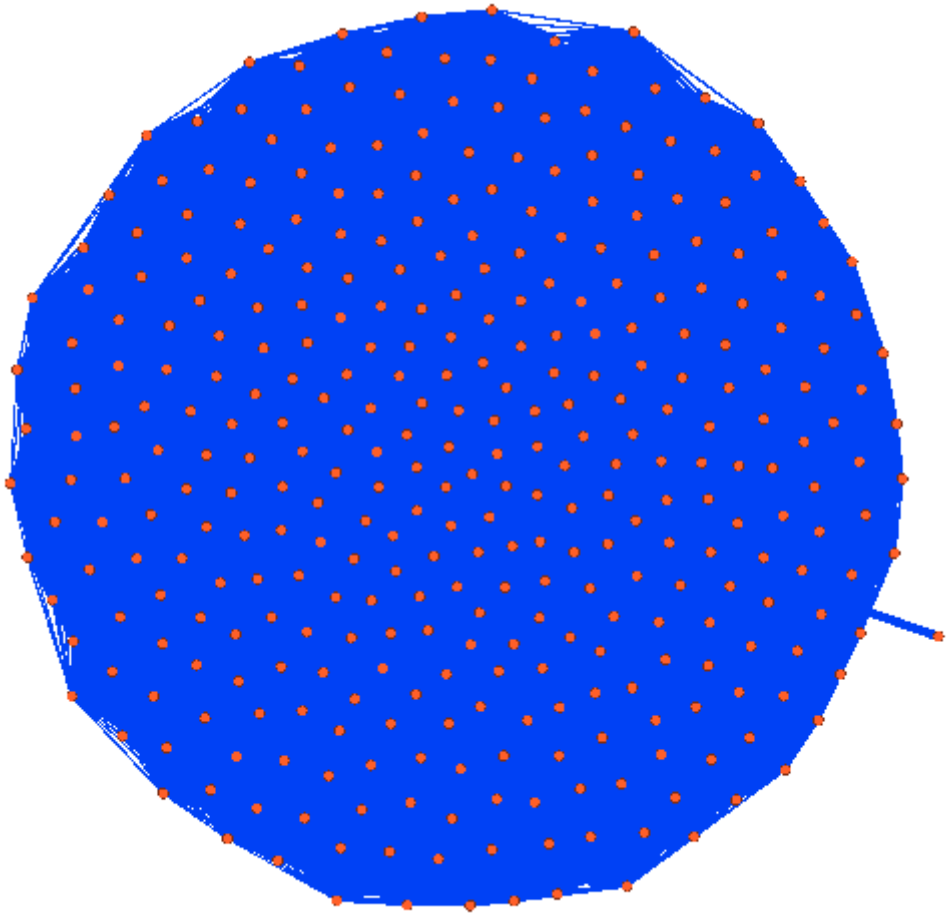
WORD2VEC


Hierarchical Clustering Dendrogram
Distance
Index
0
50
100
150
200
250
300
350
PCA of KMeans Cluster1
PCA Component 2
PCA Component 1
−15
−10
−5
0
5
10
15

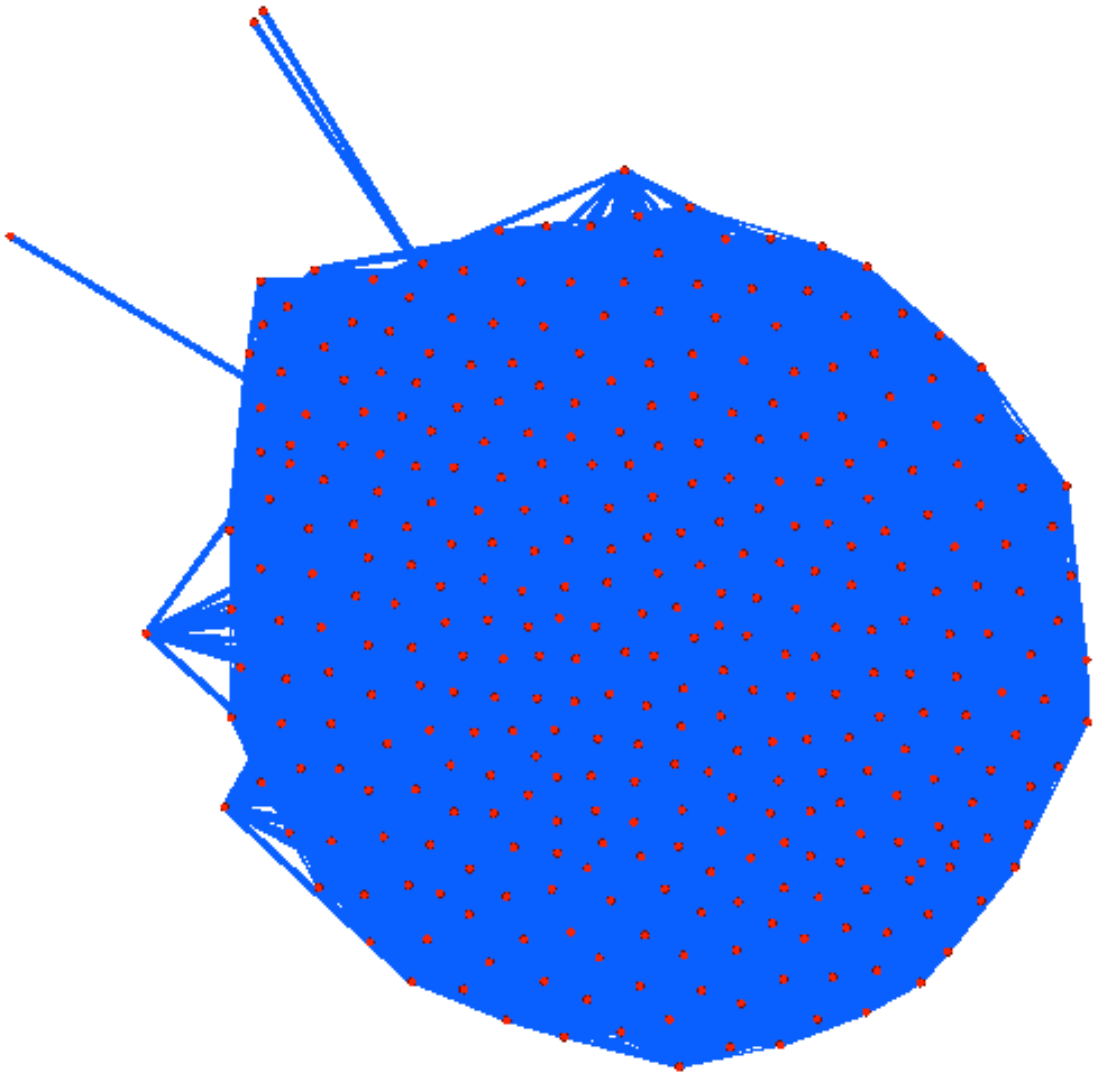
XLNET

DICES [Force Atlas 2, 20 speed, 50 scaling, prevent overlap]

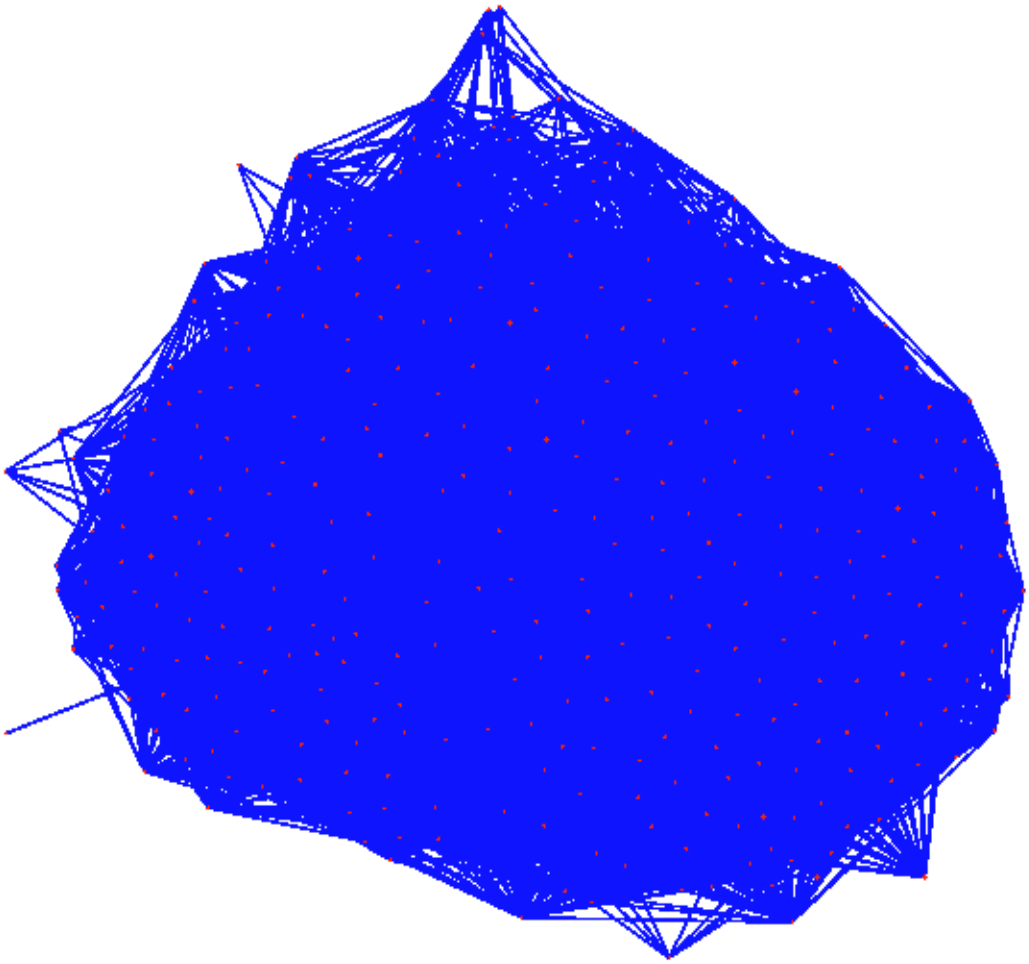
ALBERT: Nodes - 350, Edges - 60618



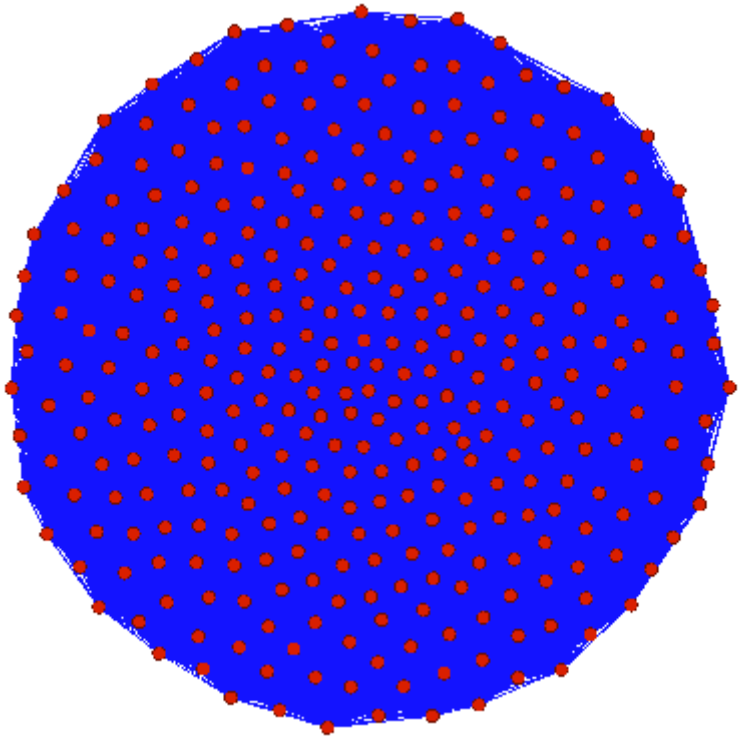
BERT: Nodes - 350, Edges - 48783



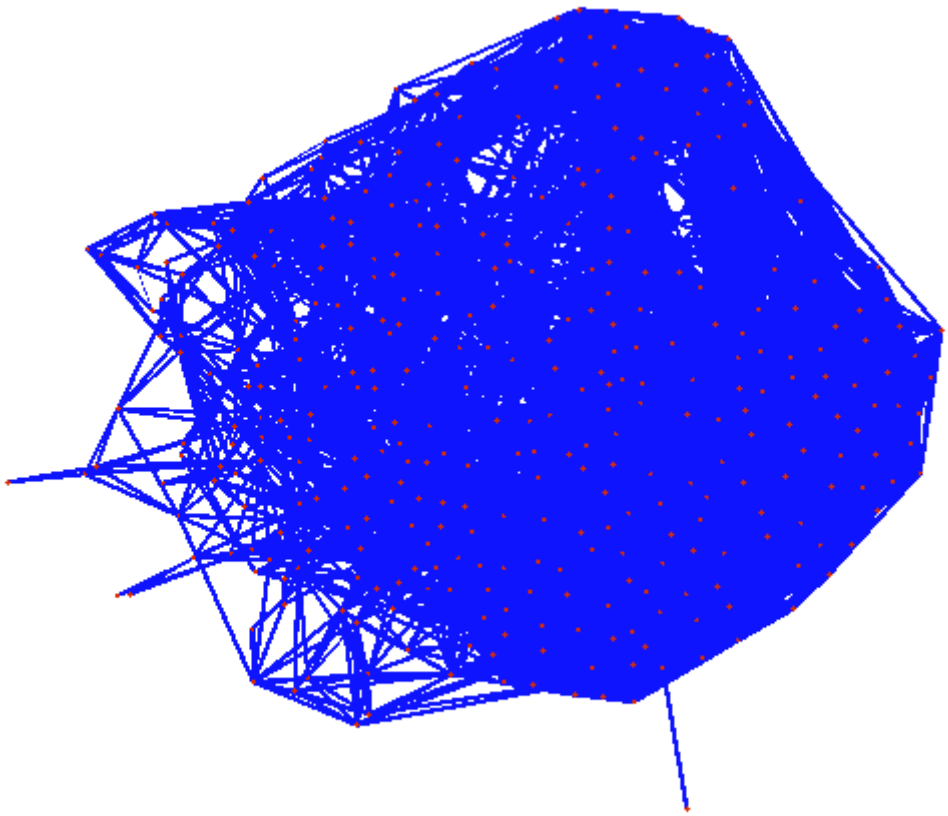
BGE: Nodes - 350, Edges - 18553



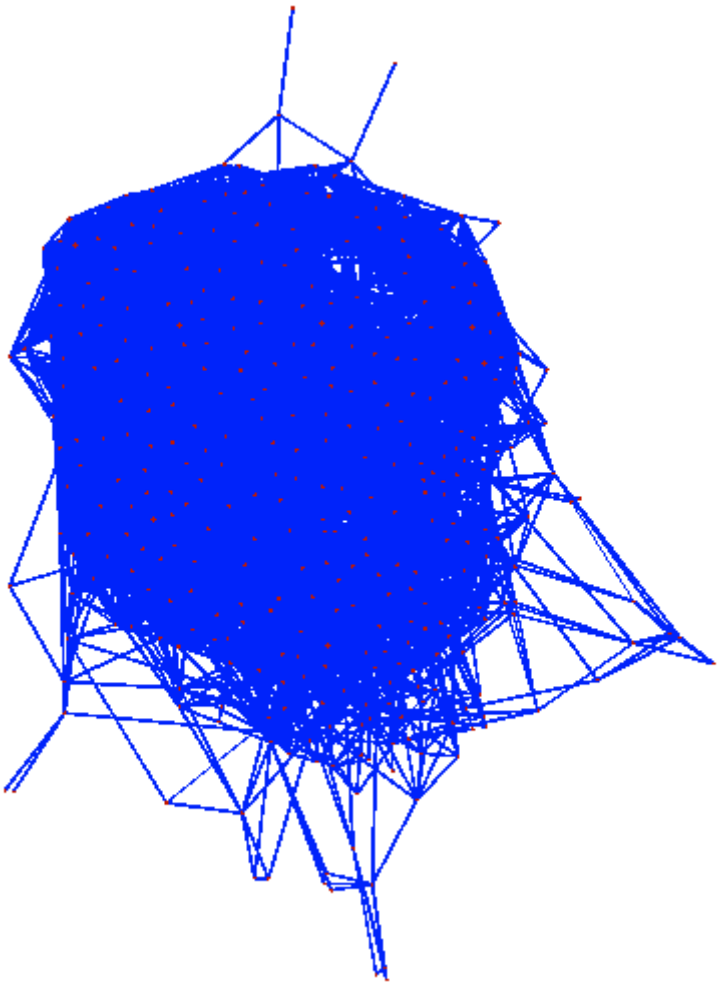
GPT: Nodes - 350, Edges - 60595



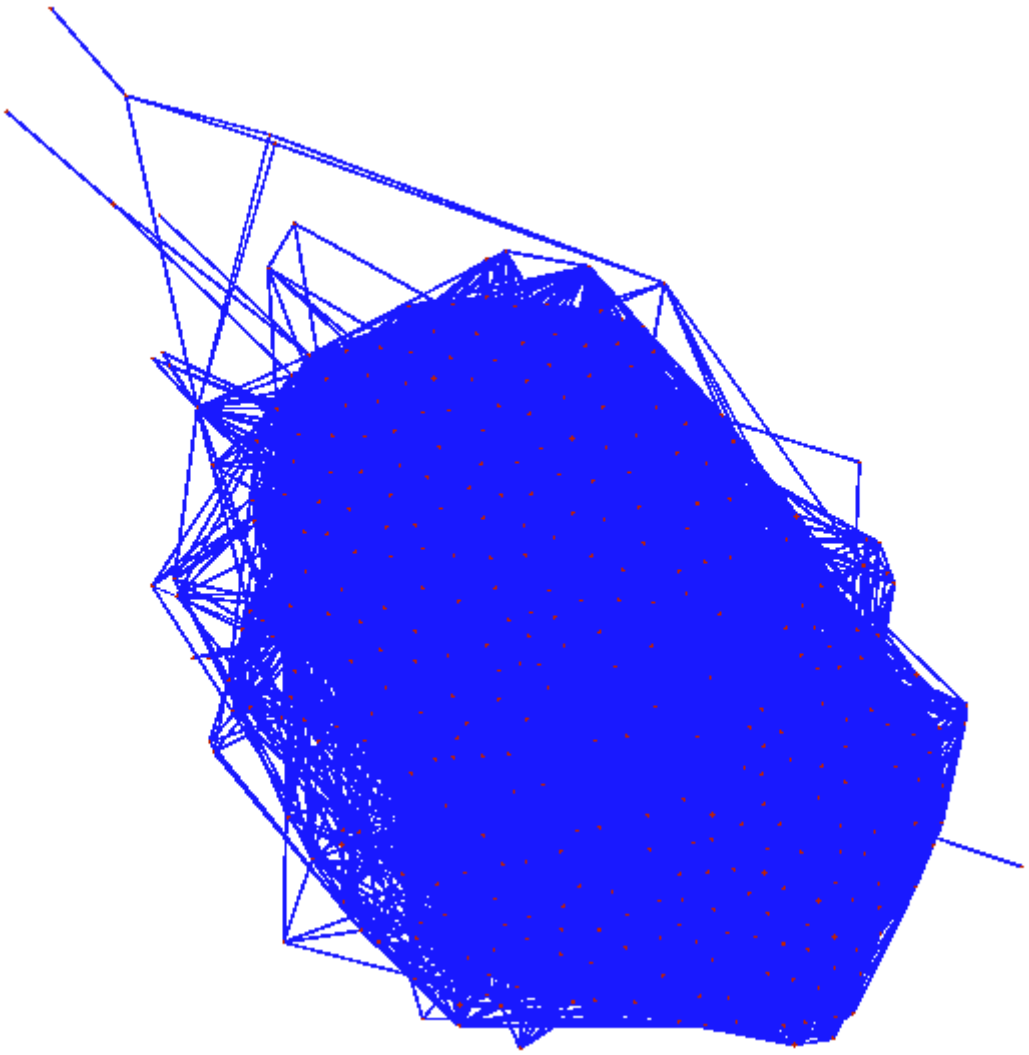
MINI: Nodes - 350, Edges - 4622



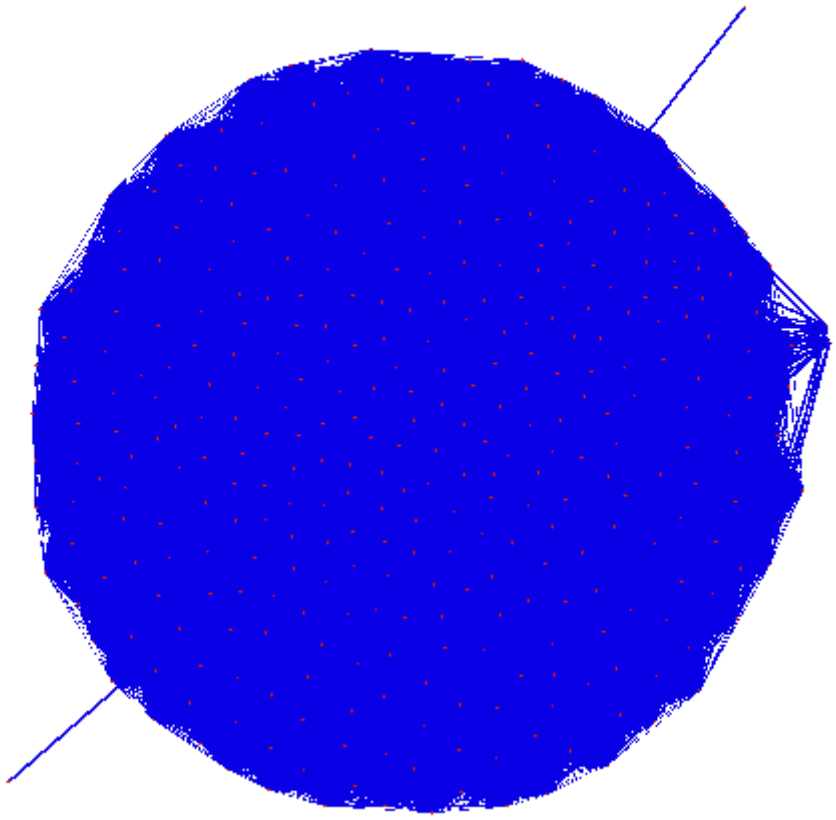
MPNET: Nodes - 350, Edges - 5210



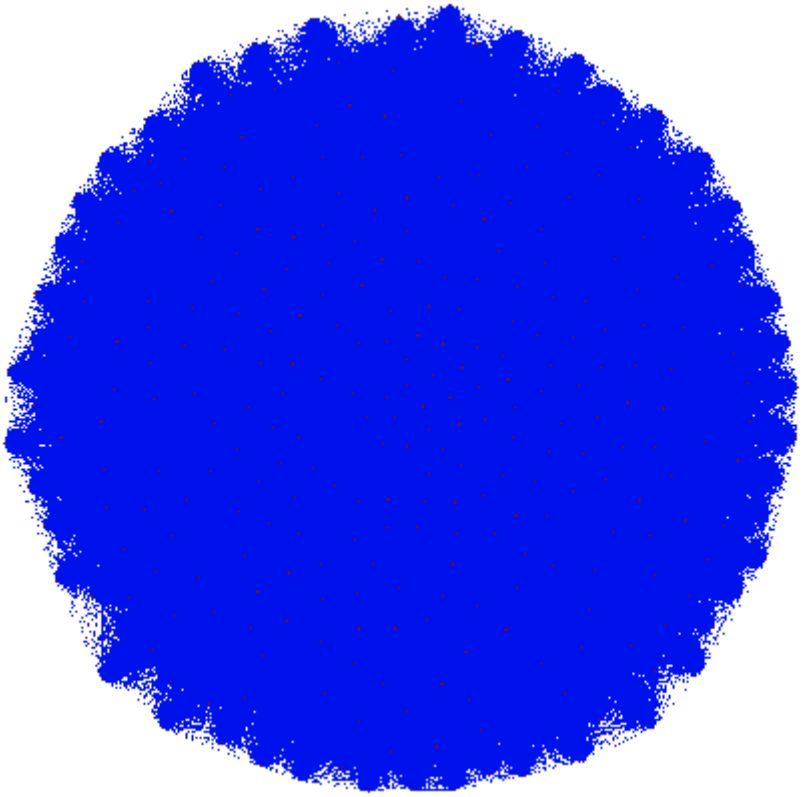
ROBERTA: Nodes - 350, Edges - 12027



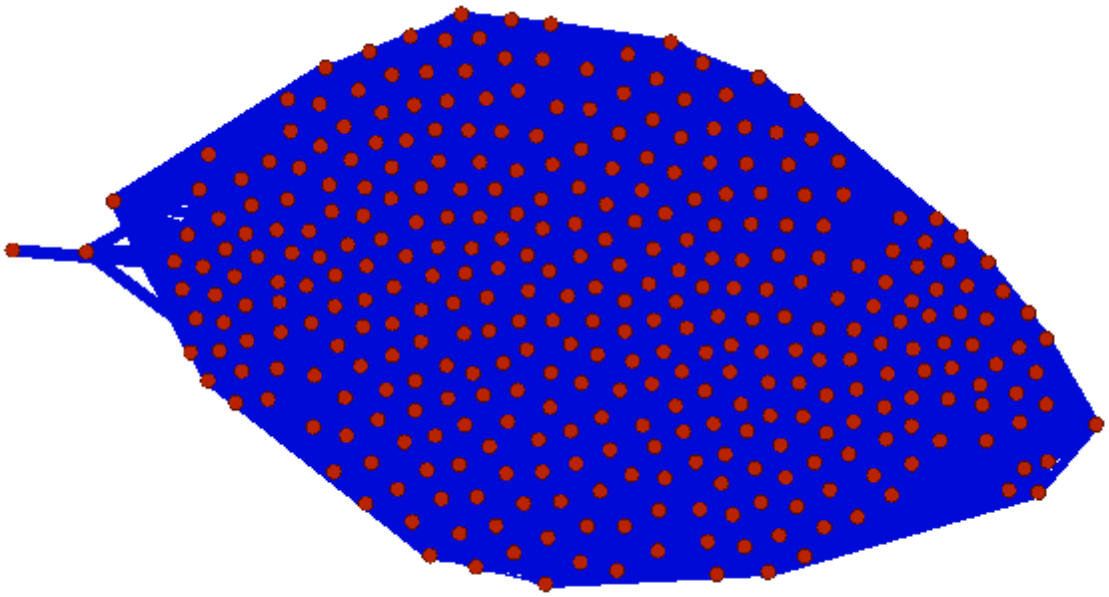
T5: Nodes - 350, Edges - 52188



Word2vec: Nodes - 350, Edges - 60624

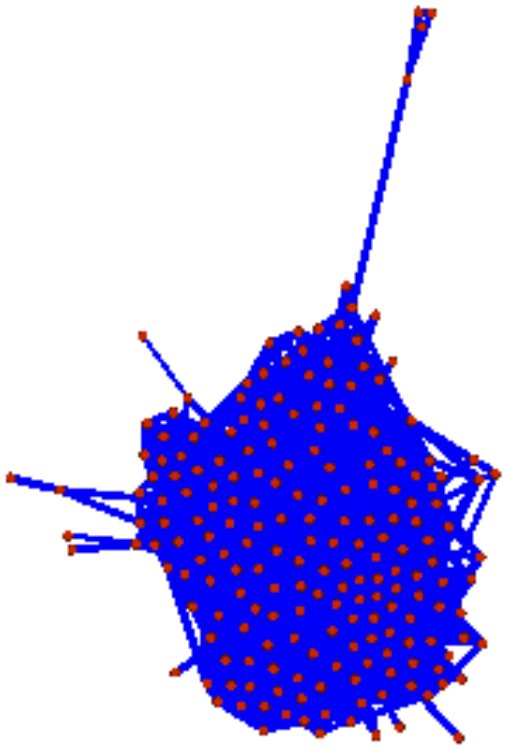


XLNET: Nodes - 350, Edges - 41964

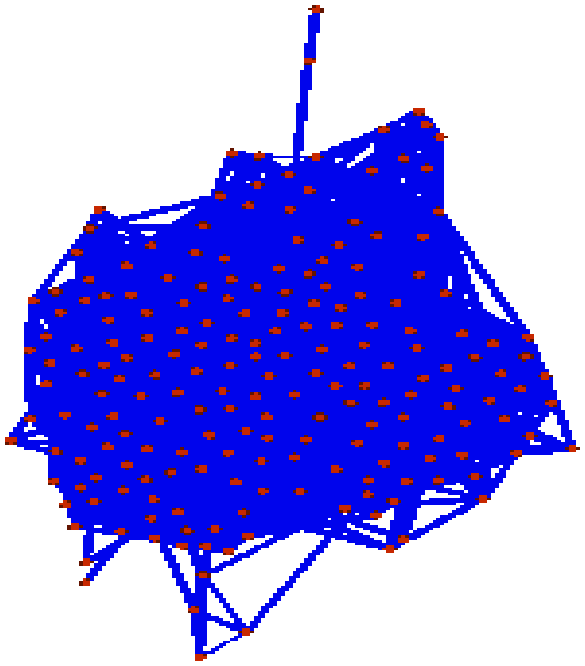


JBB [Force Atlas 2, 20 speed, 50 scaling, prevent overlap]

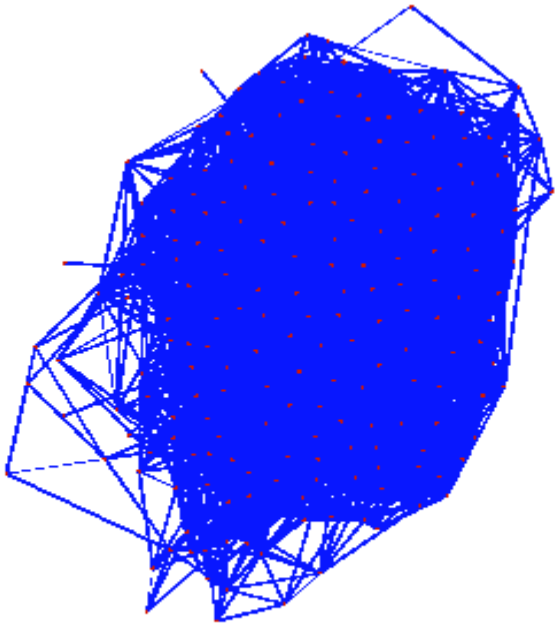
Albert: Nodes - 200, Edges - 2490



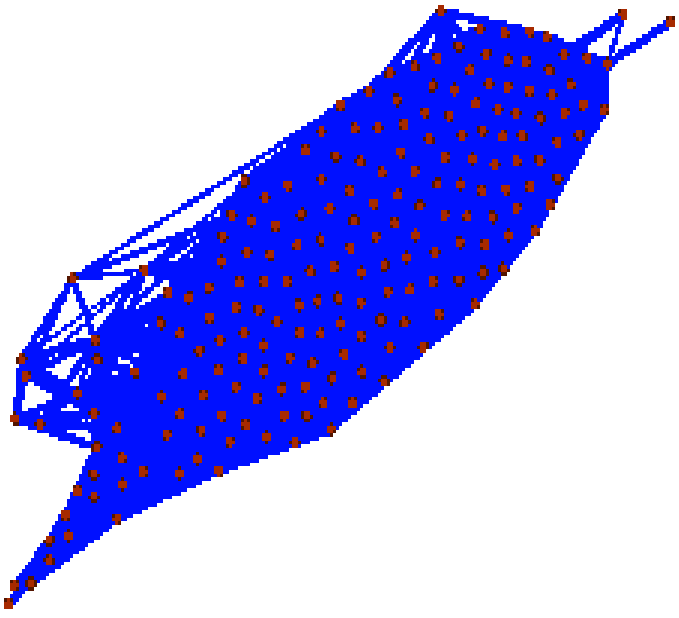
Bert: Nodes - 200, Edges - 3479



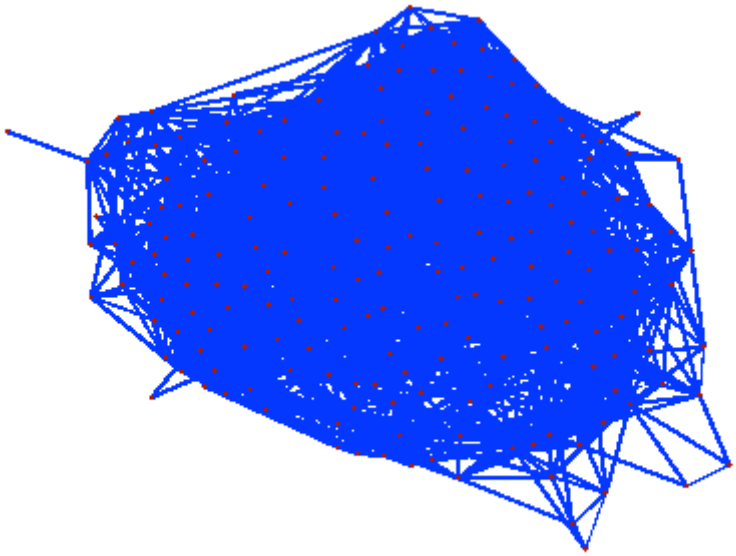
Bge: Nodes - 200, Edges - 3821



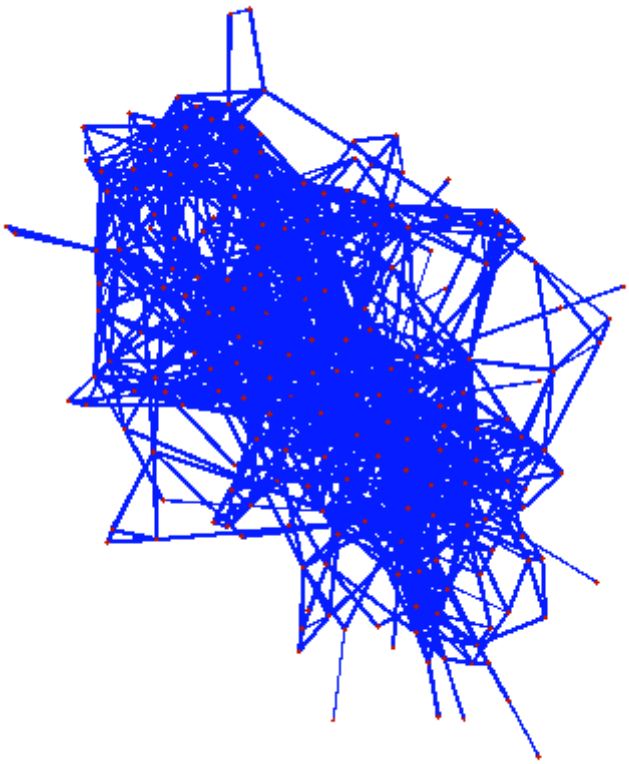
Gpt2: Nodes - 200, Edges -8235



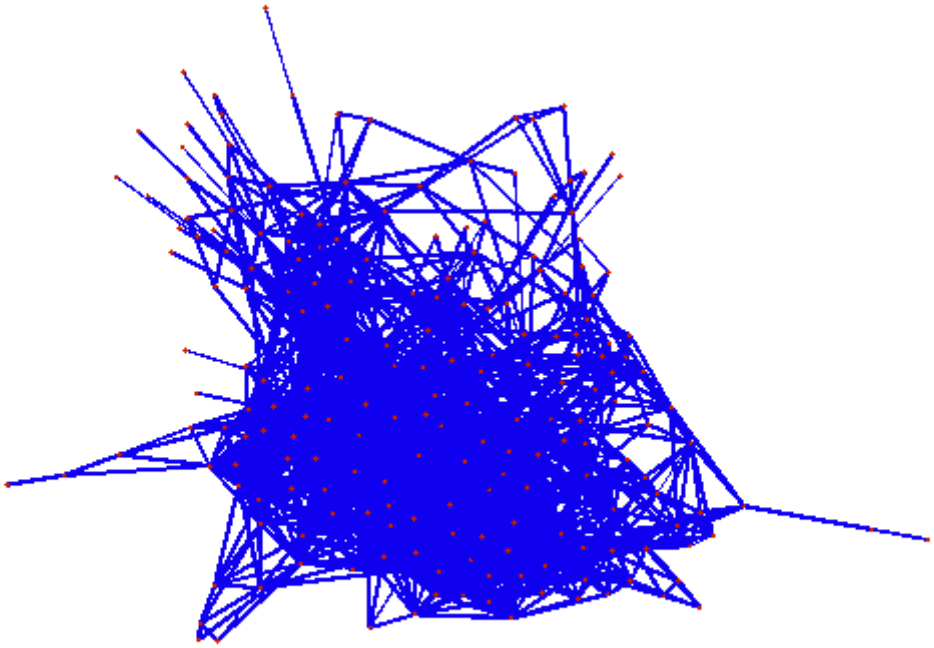
Mini: Nodes - 200, Edges - 2322



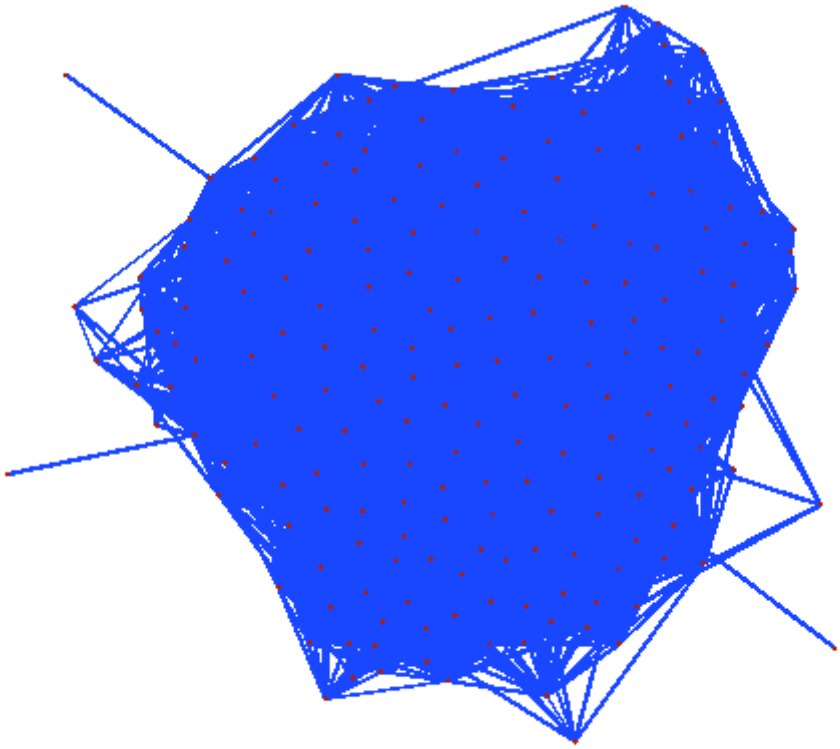
Mpnet: Nodes - 200, Edges - 1095



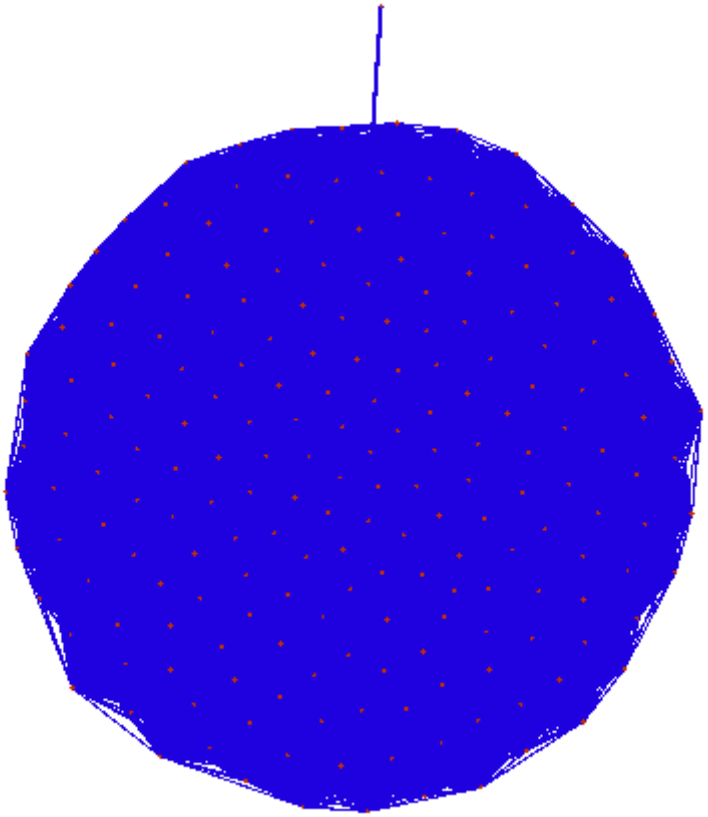
Roberta: Nodes - 200, Edges - 1413



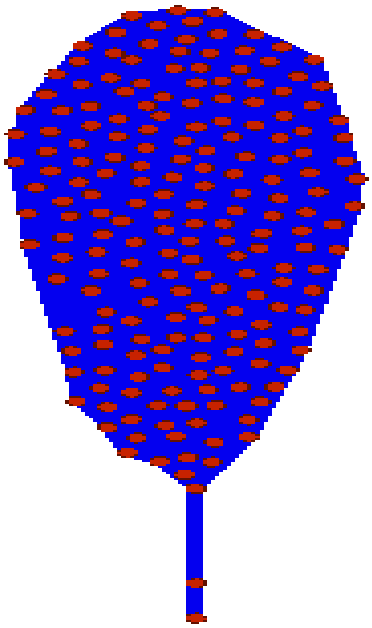
T5: Nodes - 200, Edges - 7060



Word2vec: Nodes - 200, Edges - 19305

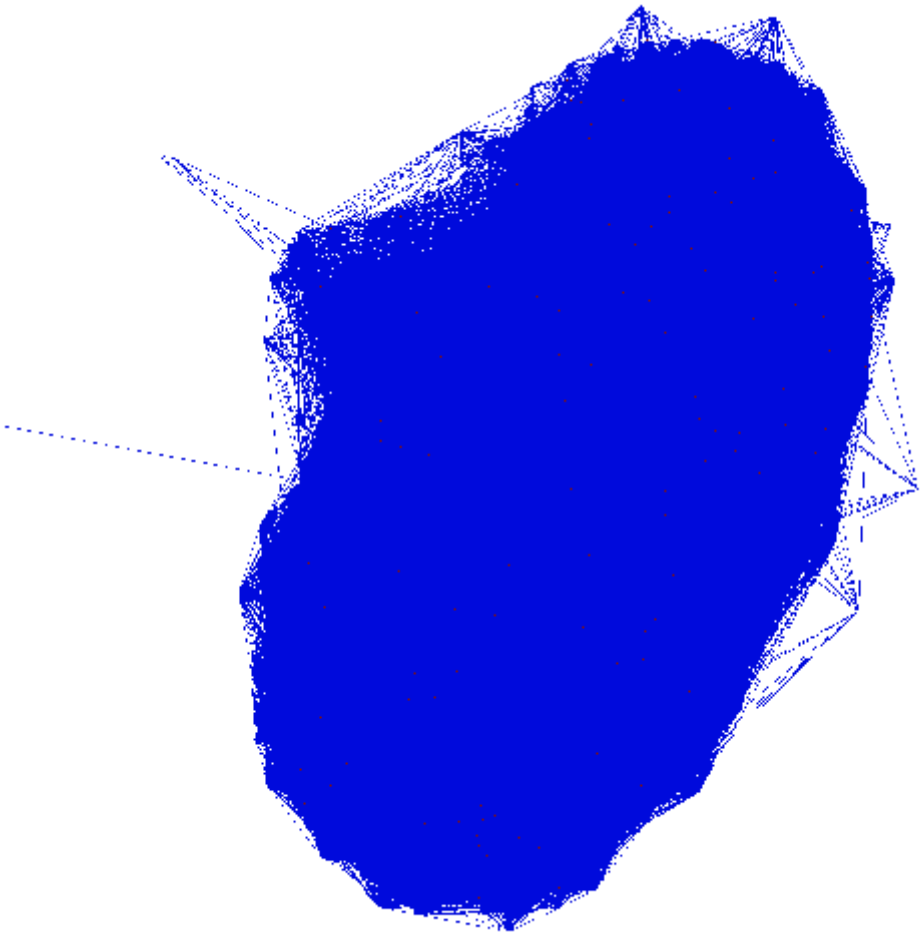


Xlnet: Nodes - 200, Edges -10192

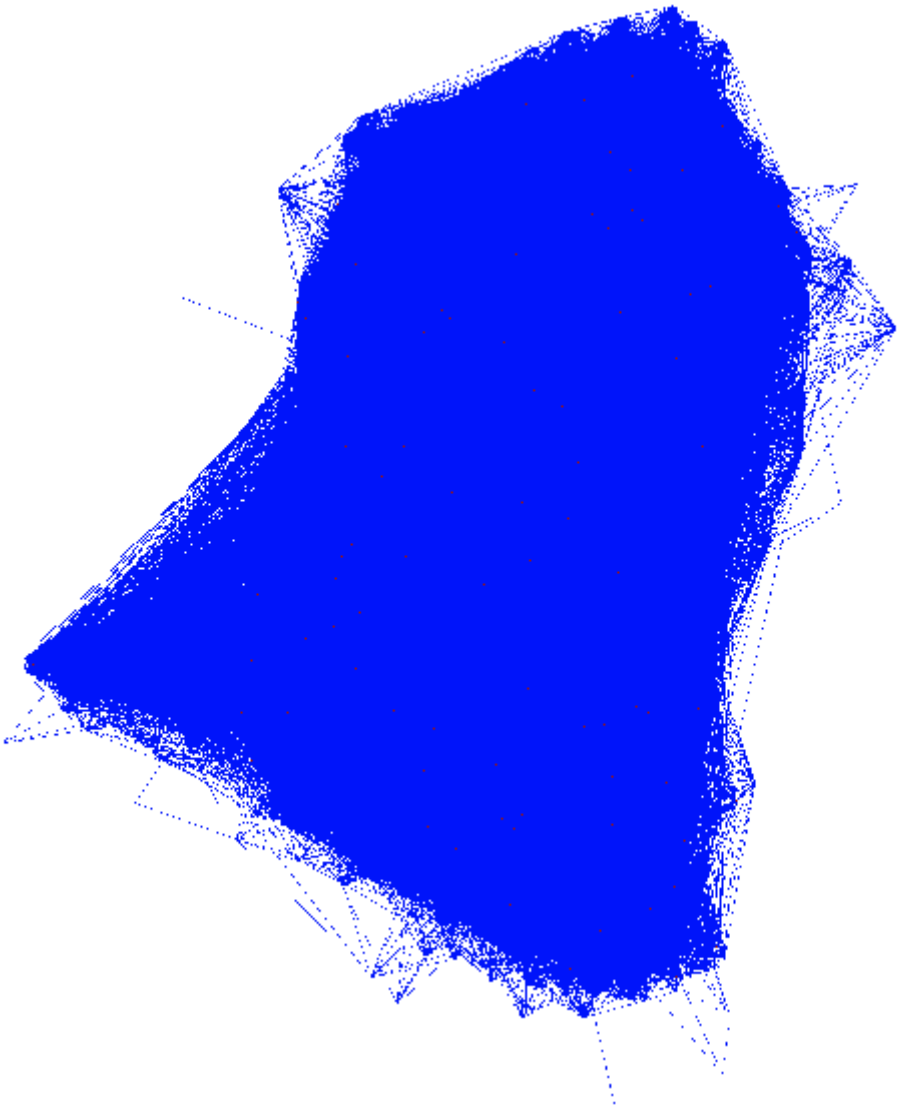


SG [Force Atlas 2, 50 speed, 500 scaling, prevent overlap]

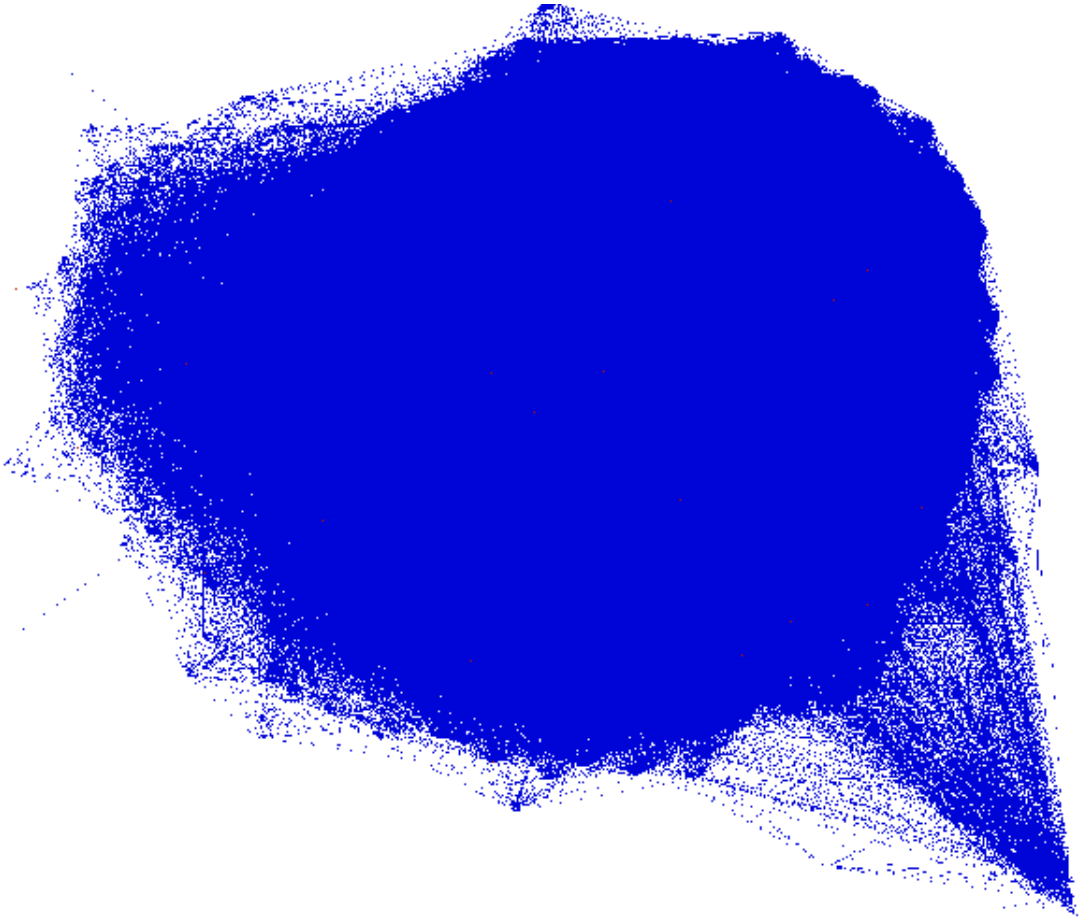
Albert: Node-1442 Edges-365244



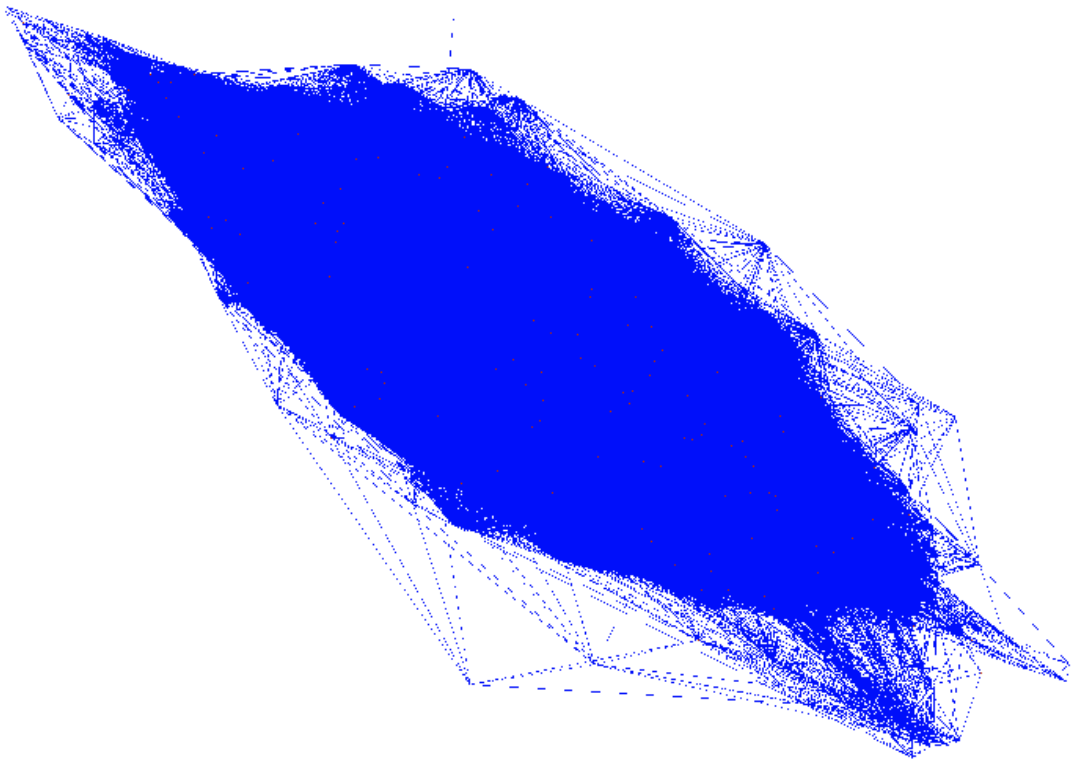
Bert: Node-1442 Edges- 273658



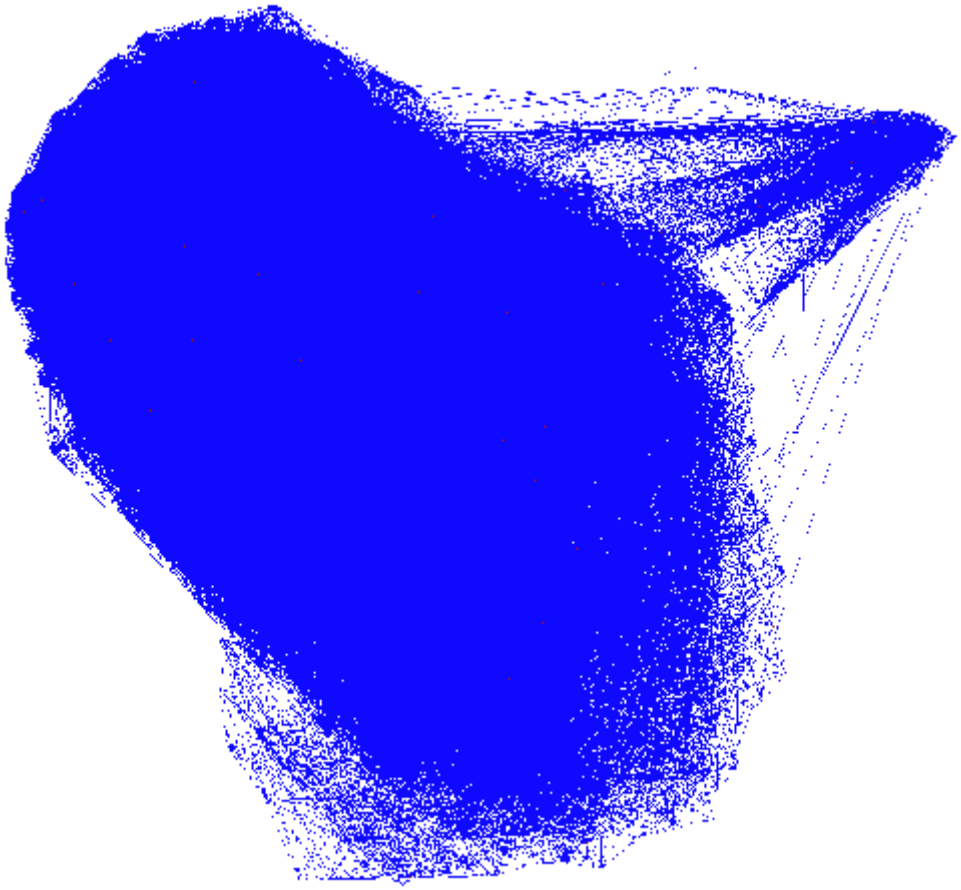
Bge: Node-1442 Edges- 358516



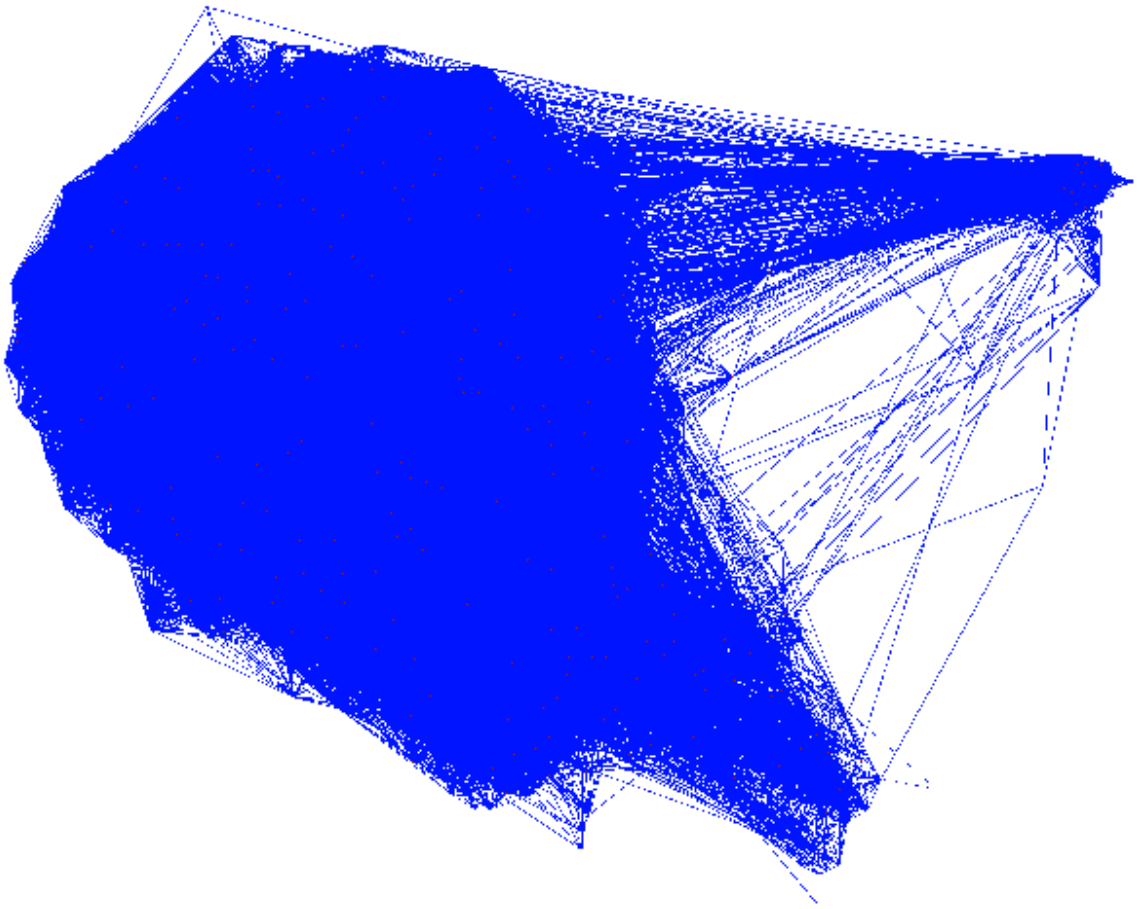
Gpt: Node-1442 Edges- 401524



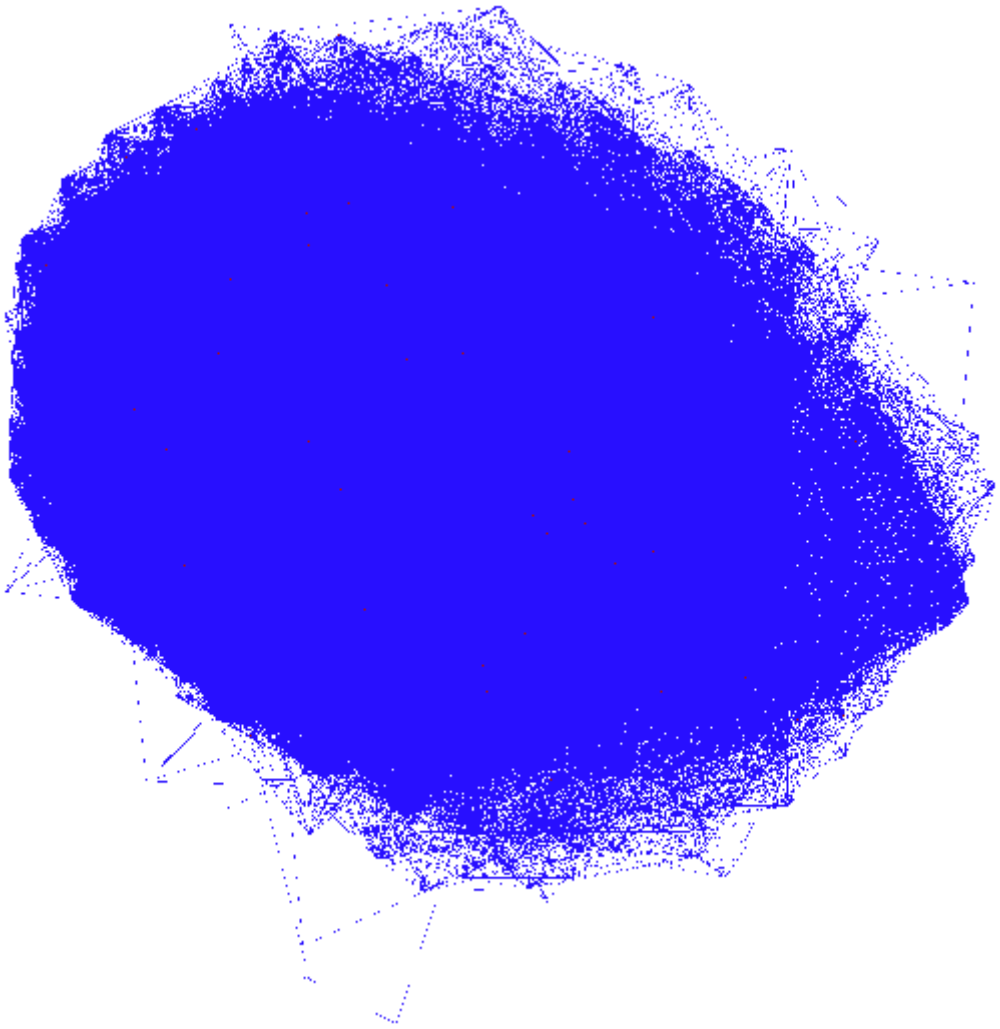
Mini: Node-1442 Edges- 104350



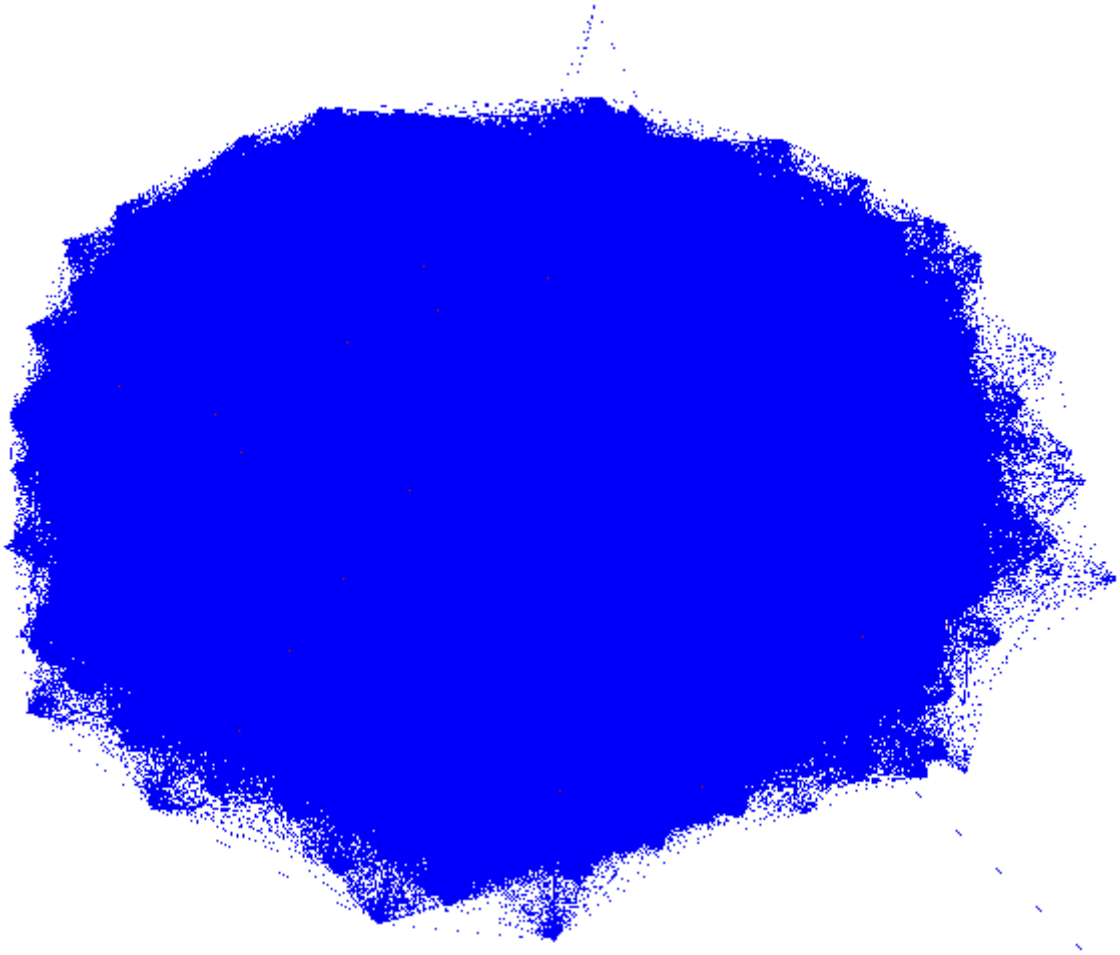
Mpnet: Node-1442 Edges- 195288



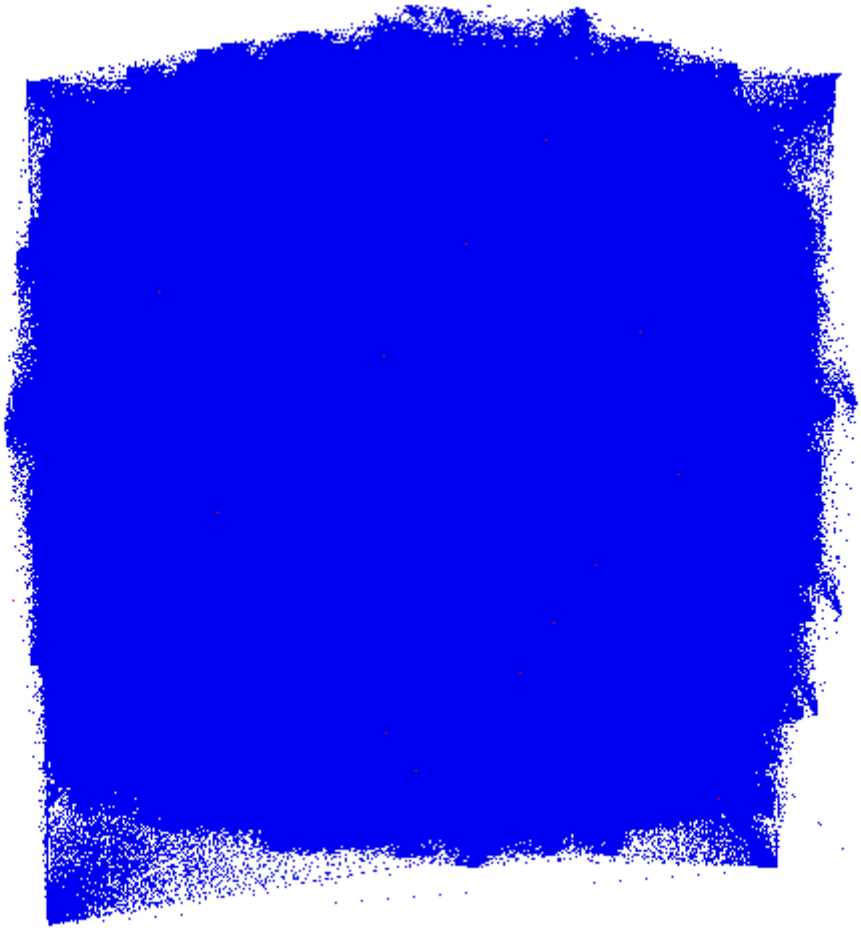
Roberta: Node-1442 Edges- 175163



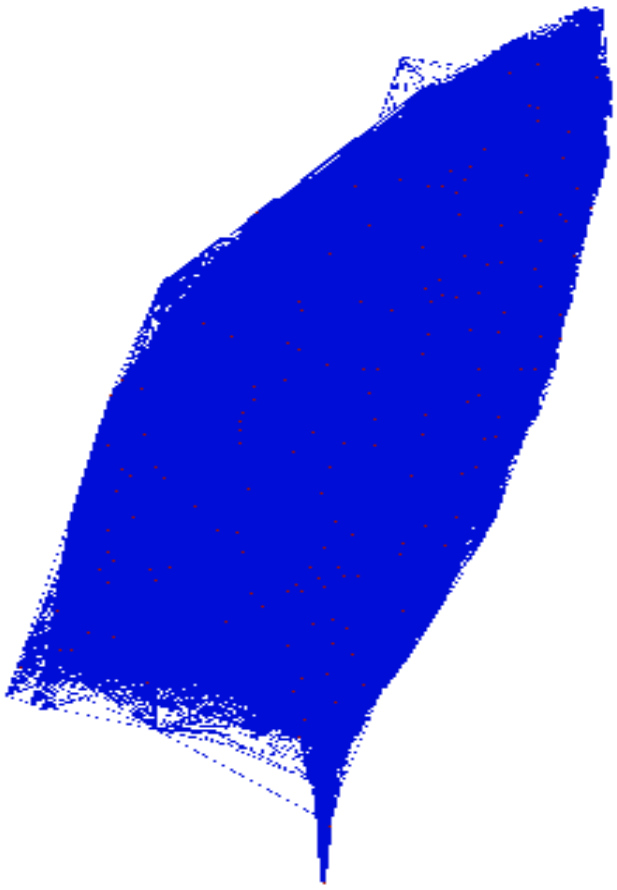
T5: Node-1442 Edges- 680146



Word2vec: Node-1442 Edges- 1031046



Xlnet: Node-1442 Edges- 349850



Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.252	0.217	0.2	0.3	0.079	0.231	0.207	0.285	0.129
BERT	0.252	0.0	0.539	0.531	0.683	0.334	0.554	0.521	0.686	0.291
MPNET	0.217	0.539	0.0	0.827	0.488	0.028	0.826	0.899	0.444	0.239
MINI	0.2	0.531	0.827	0.0	0.474	0.09	0.776	0.786	0.456	0.22
T5	0.3	0.683	0.488	0.474	0.0	0.431	0.536	0.476	0.714	0.306
word2vec	0.079	0.334	0.028	0.09	0.431	0.0	0.063	0.022	0.425	0.122
BGE	0.231	0.554	0.826	0.776	0.536	0.063	0.0	0.806	0.494	0.245
ROBERTA	0.207	0.521	0.899	0.786	0.476	0.022	0.806	0.0	0.429	0.219
ALBERT	0.285	0.686	0.444	0.456	0.714	0.425	0.494	0.429	0.0	0.279
XLNet	0.129	0.291	0.239	0.22	0.306	0.122	0.245	0.219	0.279	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 4: Spearman Distortion SG Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.433	0.124	0.105	0.521	0.432	0.31	0.205	0.532	0.238
BERT	0.433	0.0	0.258	0.181	0.849	0.815	0.495	0.467	0.832	0.439
MPNET	0.124	0.258	0.0	0.719	0.184	0.16	0.578	0.67	0.253	0.128
MINI	0.105	0.181	0.719	0.0	0.111	0.077	0.539	0.545	0.191	0.052
T5	0.521	0.849	0.184	0.111	0.0	0.868	0.444	0.388	0.911	0.463
word2vec	0.432	0.815	0.16	0.077	0.868	0.0	0.399	0.36	0.829	0.444
BGE	0.31	0.495	0.578	0.539	0.444	0.399	0.0	0.616	0.496	0.225
ROBERTA	0.205	0.467	0.67	0.545	0.388	0.36	0.616	0.0	0.421	0.285
ALBERT	0.532	0.832	0.253	0.191	0.911	0.829	0.496	0.421	0.0	0.435
XLNet	0.238	0.439	0.128	0.052	0.463	0.444	0.225	0.285	0.435	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 5: Spearman Distortion DICEs Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.424	0.397	0.342	0.458	0.376	0.35	0.382	0.419	0.267
BERT	0.424	0.0	0.742	0.68	0.87	0.741	0.584	0.731	0.836	0.513
MPNET	0.397	0.742	0.0	0.882	0.738	0.617	0.729	0.915	0.706	0.428
MINI	0.342	0.68	0.882	0.0	0.68	0.584	0.745	0.854	0.654	0.379
T5	0.458	0.87	0.738	0.68	0.0	0.799	0.592	0.722	0.895	0.532
word2vec	0.376	0.741	0.617	0.584	0.799	0.0	0.494	0.599	0.756	0.465
BGE	0.35	0.584	0.729	0.745	0.592	0.494	0.0	0.707	0.577	0.338
ROBERTA	0.382	0.731	0.915	0.854	0.722	0.599	0.707	0.0	0.7	0.417
ALBERT	0.419	0.836	0.706	0.654	0.895	0.756	0.577	0.7	0.0	0.511
XLNet	0.267	0.513	0.428	0.379	0.532	0.465	0.338	0.417	0.511	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 6: Spearman Distortion GEN Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.106	0.071	0.061	0.171	0.149	0.09	0.062	0.169	0.075
BERT	0.106	0.0	0.288	0.292	0.492	0.31	0.22	0.29	0.563	0.153
MPNET	0.071	0.288	0.0	0.695	0.288	0.038	0.607	0.794	0.26	0.083
MINI	0.061	0.292	0.695	0.0	0.323	0.081	0.521	0.651	0.3	0.091
T5	0.171	0.492	0.288	0.323	0.0	0.506	0.21	0.28	0.636	0.171
word2vec	0.149	0.31	0.038	0.081	0.506	0.0	-0.066	0.046	0.481	0.139
BGE	0.09	0.22	0.607	0.521	0.21	-0.066	0.0	0.581	0.205	0.101
ROBERTA	0.062	0.29	0.794	0.651	0.28	0.046	0.581	0.0	0.255	0.073
ALBERT	0.169	0.563	0.26	0.3	0.636	0.481	0.205	0.255	0.0	0.217
XLNet	0.075	0.153	0.083	0.091	0.171	0.139	0.101	0.073	0.217	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 7: Spearman Distortion GEST Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.22	0.095	0.115	0.252	0.18	0.167	0.064	0.266	0.05
BERT	0.22	0.0	0.506	0.478	0.527	0.381	0.408	0.513	0.633	0.092
MPNET	0.095	0.506	0.0	0.754	0.482	0.302	0.601	0.812	0.446	0.122
MINI	0.115	0.478	0.754	0.0	0.466	0.265	0.605	0.659	0.453	0.142
T5	0.252	0.527	0.482	0.466	0.0	0.424	0.45	0.484	0.65	0.13
word2vec	0.18	0.381	0.302	0.265	0.424	0.0	0.215	0.335	0.424	0.031
BGE	0.167	0.408	0.601	0.605	0.45	0.215	0.0	0.512	0.449	0.098
ROBERTA	0.064	0.513	0.812	0.659	0.484	0.335	0.512	0.0	0.427	0.092
ALBERT	0.266	0.633	0.446	0.453	0.65	0.424	0.449	0.427	0.0	0.125
XLNet	0.05	0.092	0.122	0.142	0.13	0.031	0.098	0.092	0.125	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 8: Spearman Distortion JBB Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.233	0.135	0.13	0.247	0.114	0.031	0.156	0.307	0.103
BERT	0.233	0.0	0.255	0.26	0.595	0.518	0.259	0.32	0.79	0.086
MPNET	0.135	0.255	0.0	0.649	0.24	0.084	0.329	0.686	0.257	0.03
MINI	0.13	0.26	0.649	0.0	0.231	0.089	0.327	0.581	0.264	0.034
T5	0.247	0.595	0.24	0.231	0.0	0.465	0.173	0.285	0.64	0.089
word2vec	0.114	0.518	0.084	0.089	0.465	0.0	0.198	0.134	0.575	-0.007
BGE	0.031	0.259	0.329	0.327	0.173	0.198	0.0	0.359	0.286	0.013
ROBERTA	0.156	0.32	0.686	0.581	0.285	0.134	0.359	0.0	0.326	0.036
ALBERT	0.307	0.79	0.257	0.264	0.64	0.575	0.286	0.326	0.0	0.101
XLNet	0.103	0.086	0.03	0.034	0.089	-0.007	0.013	0.036	0.101	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 9: Spearman Distortion MTEB Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.154	0.113	0.107	0.212	0.081	0.132	0.11	0.258	0.088
BERT	0.154	0.0	0.4	0.413	0.517	0.192	0.35	0.405	0.591	0.164
MPNET	0.113	0.4	0.0	0.737	0.372	0.187	0.54	0.766	0.39	0.137
MINI	0.107	0.413	0.737	0.0	0.36	0.185	0.529	0.673	0.401	0.127
T5	0.212	0.517	0.372	0.36	0.0	0.321	0.381	0.379	0.656	0.262
word2vec	0.081	0.192	0.187	0.185	0.321	0.0	0.092	0.16	0.319	0.127
BGE	0.132	0.35	0.54	0.529	0.381	0.092	0.0	0.474	0.396	0.142
ROBERTA	0.11	0.405	0.766	0.673	0.379	0.16	0.474	0.0	0.374	0.138
ALBERT	0.258	0.591	0.39	0.401	0.656	0.319	0.396	0.374	0.0	0.227
XLNet	0.088	0.164	0.137	0.127	0.262	0.127	0.142	0.138	0.227	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 10: Spearman Distortion SAFE Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.261	0.071	0.077	0.303	0.201	0.077	0.092	0.347	0.158
BERT	0.261	0.0	0.239	0.259	0.665	0.5	0.224	0.253	0.703	0.2
MPNET	0.071	0.239	0.0	0.716	0.136	0.017	0.5	0.754	0.223	0.055
MINI	0.077	0.259	0.716	0.0	0.158	0.044	0.502	0.659	0.244	0.055
T5	0.303	0.665	0.136	0.158	0.0	0.684	0.176	0.153	0.714	0.242
word2vec	0.201	0.5	0.017	0.044	0.684	0.0	0.113	0.02	0.561	0.171
BGE	0.077	0.224	0.5	0.502	0.176	0.113	0.0	0.483	0.275	0.06
ROBERTA	0.092	0.253	0.754	0.659	0.153	0.02	0.483	0.0	0.234	0.061
ALBERT	0.347	0.703	0.223	0.244	0.714	0.561	0.275	0.234	0.0	0.236
XLNet	0.158	0.2	0.055	0.055	0.242	0.171	0.06	0.061	0.236	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 11: Spearman Distortion PRISM Dataset

Spearman Correlation

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.182	0.083	0.123	0.225	0.155	0.134	0.13	0.274	0.141
BERT	0.182	0.0	0.426	0.484	0.806	0.713	0.526	0.362	0.798	0.34
MPNET	0.083	0.426	0.0	0.748	0.344	0.311	0.64	0.757	0.387	0.151
MINI	0.123	0.484	0.748	0.0	0.407	0.371	0.684	0.649	0.467	0.191
T5	0.225	0.806	0.344	0.407	0.0	0.736	0.468	0.302	0.807	0.359
word2vec	0.155	0.713	0.311	0.371	0.736	0.0	0.408	0.245	0.701	0.276
BGE	0.134	0.526	0.64	0.684	0.468	0.408	0.0	0.578	0.499	0.219
ROBERTA	0.13	0.362	0.757	0.649	0.302	0.245	0.578	0.0	0.357	0.157
ALBERT	0.274	0.798	0.387	0.467	0.807	0.701	0.499	0.357	0.0	0.414
XLNet	0.141	0.34	0.151	0.191	0.359	0.276	0.219	0.157	0.414	0.0

P-Values

	GPT2	BERT	MPNET	MINI	T5	word2vec	BGE	ROBERTA	ALBERT	XLNet
GPT2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MPNET	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
MINI	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
T5	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
word2vec	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
BGE	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ROBERTA	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALBERT	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
XLNet	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 12: Spearman Distortion Wild Dataset

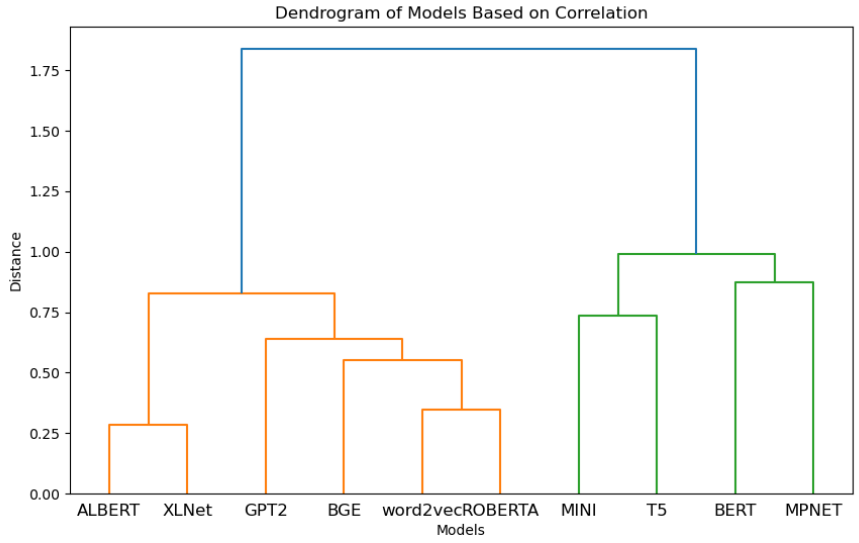


Figure 13: Spearman Dendrogram

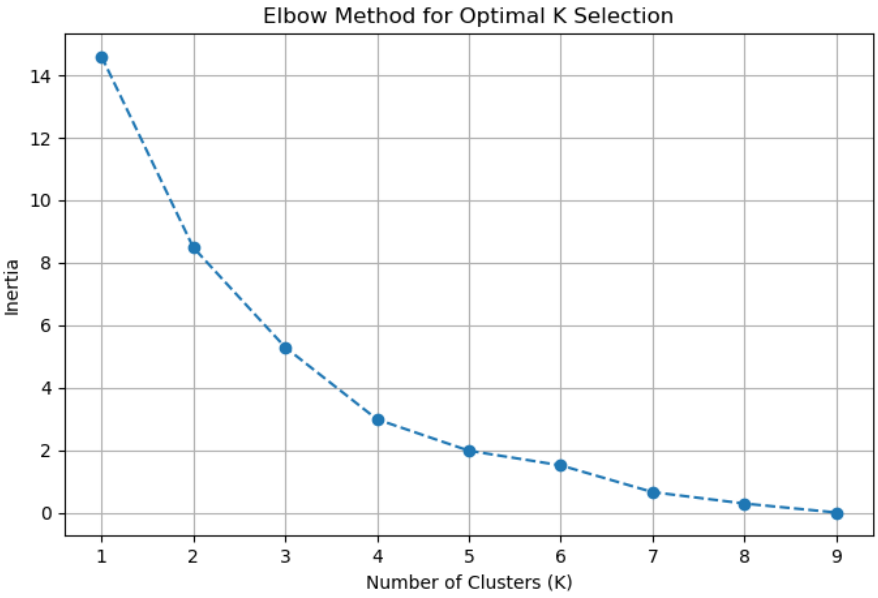


Figure 14: Optimal K clusters on dataset

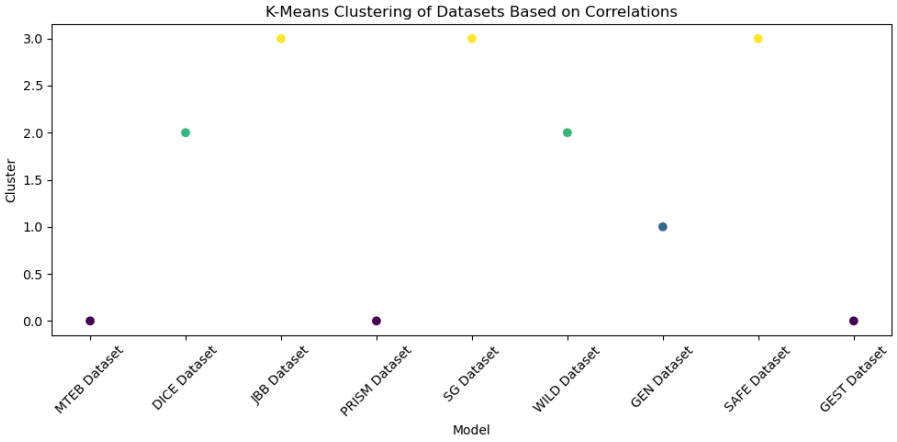


Figure 15: K-mean Clustering on spearman