Understanding and Detecting Malicious Prompts Using Embeddings, Clustering, and Classification

Project Report

Author:
Alicja Borek

Kraków, January 30, 2026

1 Introduction

The rapid development of Large Language Models (LLMs) has introduced new challenges in the field of cybersecurity and content safety. While models are trained to be helpful, they can be manipulated to generate harmful content. The goal of this project was to analyze the vector space (embeddings) of malicious prompts using the *PL-Guard* dataset.

2 Data and Preprocessing

I utilized the *PL-Guard* dataset, which contains Polish prompts assigned to 15 distinct malicious categories:

Table 1: Mapping of dataset labels to descriptive categories.

Raw Label (ID)	**Descriptive Category Name**
safe	Safe
unsafe S1	Violent Crimes
unsafe S2	Non-Violent Crimes
unsafe S3	Sex-Related Crimes
unsafe S4	Child Sexual Exploitation
unsafe S5	Defamation
unsafe S6	Specialized Advice
unsafe S7	Privacy
unsafe S8	Intellectual Property
unsafe S9	Indiscriminate Weapons
unsafe S10	Hate
unsafe S11	Suicide & Self-Harm
unsafe S12	Sexual Content
unsafe S13	Elections
unsafe S14	Code Interpreter Abuse

The dataset is particularly interesting because it includes "adversarial" variants—prompts that maintain their malicious intent but are linguistically modified (e.g., via typos, stealthy phrasing) to evade detection.

2.1 Embedding Model Selection and Comparative Analysis

To determine the optimal numerical representation for the text, I conducted a series of experiments comparing static and contextual embedding models.

2.1.1 Static vs. Contextual Models

Initially, I utilized the **Spacy** library to generate baseline embeddings. While computationally efficient, the visualization (UMAP) of Spacy vectors revealed that it primarily clustered texts based on keywords rather than intent. To capture deeper semantic nuances, I transitioned to Transformer-based models.

2.1.2 Transformer Architecture Experiments

I evaluated several BERT-based architectures to find the best balance between language understanding and clustering capability:

- `bert-base-uncased`: The standard general-purpose model, which served as a reference point.
- `allegro/herbert-base-cased`: A model pre-trained specifically on the National Corpus of Polish, which theoretically offers the best understanding of the language syntax and vocabulary.

2.1.3 Final Selection: Sentence-BERT

Consequently, I ultimately selected the `paraphrase-multilingual-MiniLM-L12-v2` model. Unlike the base BERT or HerBERT models, this model is based on the **Sentence-BERT (SBERT)** architecture, which is explicitly fine-tuned to map semantically similar sentences to close points in the vector space.

Conclusion: The SBERT model outperformed both Spacy (by understanding context) and standard BERT/HerBERT models (by providing well-separated clusters out-of-the-box), making it the optimal choice for this specific project.

3 Dimensionality Reduction and Visualization

High-dimensional data (384 dimensions for BERT, 300 for Spacy) is impossible to visualize directly and often suffers from the "curse of dimensionality." To find the optimal visualization strategy, I conducted a comparative analysis of **three distinct dimensionality reduction techniques** applied to both Spacy and BERT embeddings:

- **PCA (Principal Component Analysis):** A linear method used as a baseline.
- **t-SNE (t-Distributed Stochastic Neighbor Embedding):** A popular non-linear probabilistic technique.
- **UMAP (Uniform Manifold Approximation and Projection):** A modern manifold learning technique known for preserving both local and global structure.

3.1 Evaluation of Reduction Techniques

3.1.1 Linear vs. Non-linear Methods

I first applied **PCA** to reduce the dimensions to 2D. However, the results were unsatisfactory for both embedding types. As a linear method, PCA failed to capture the complex, non-linear relationships between the text vectors. The resulting visualization showed a "smeared" cloud of points with significant overlap between distinct categories, confirming that the semantic boundaries are not linearly separable.

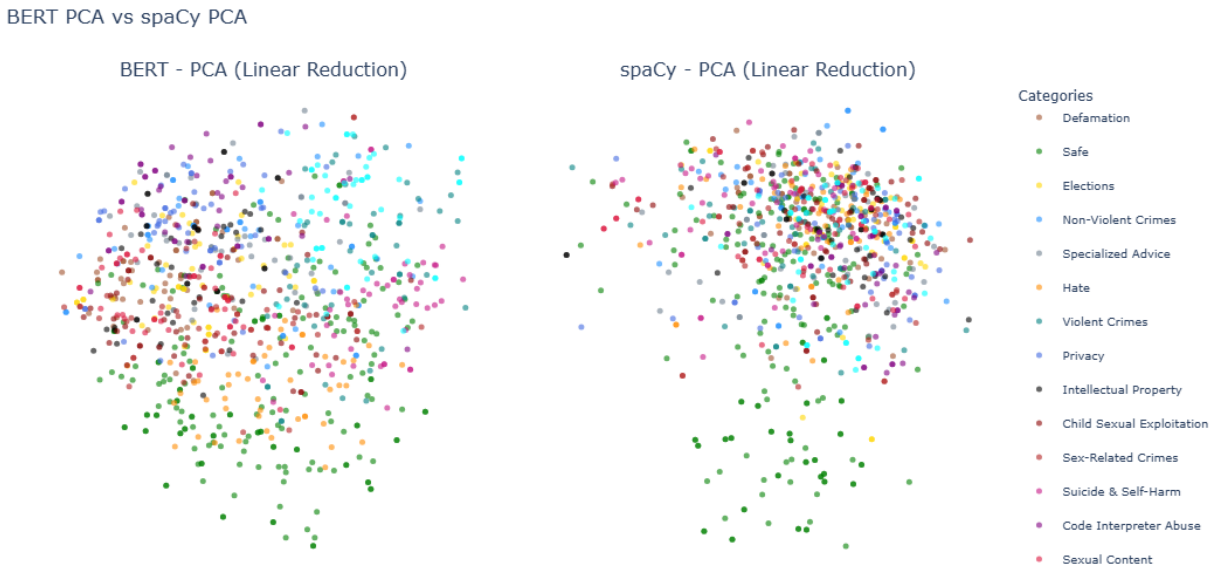

Figure 1: Comparison of Principal Component Analysis (PCA) projections.

3.1.2 Quantitative Analysis of Information Loss (PCA)

To mathematically verify why the linear projection failed, I analyzed the *Explained Variance Ratio* of the first two principal components. This metric indicates how much of the original semantic information is preserved after reducing the dimensions to 2D.

Table 2: Explained Variance Ratio for the first two Principal Components (PCA).

Model	PC1 Variance	PC2 Variance	Cumulative Variance (2D)
BERT (MiniLM)	7.62%	6.87%	**14.48%**
spaCy (Static)	14.57%	9.71%	**24.28%**

The results in Table 2 reveal a critical issue with linear reduction for this dataset:

1. **High Information Loss:** For the BERT model, the 2D projection preserves only $\approx 14.5\%$ of the original variance. This means that over **85% of the semantic information is lost** during the projection.

2. **Model Complexity:** The BERT embeddings show significantly lower explained variance compared to spaCy (14.5% vs 24.3%). This confirms that contextual embeddings possess a much more complex, high-dimensional structure that is distributed across many dimensions, making them inherently unsuitable for linear compression to 2D.

3.1.3 t-SNE vs. UMAP

Subsequently, I tested the non-linear methods. While **t-SNE** successfully created separated clusters, it was computationally expensive and struggled to preserve the global structure of the data (distances between clusters were not always meaningful).

In contrast, **UMAP** produced the best results. It successfully preserved the local structure (grouping similar prompts) while maintaining the global context. It also proved to be faster and more scalable than t-SNE.

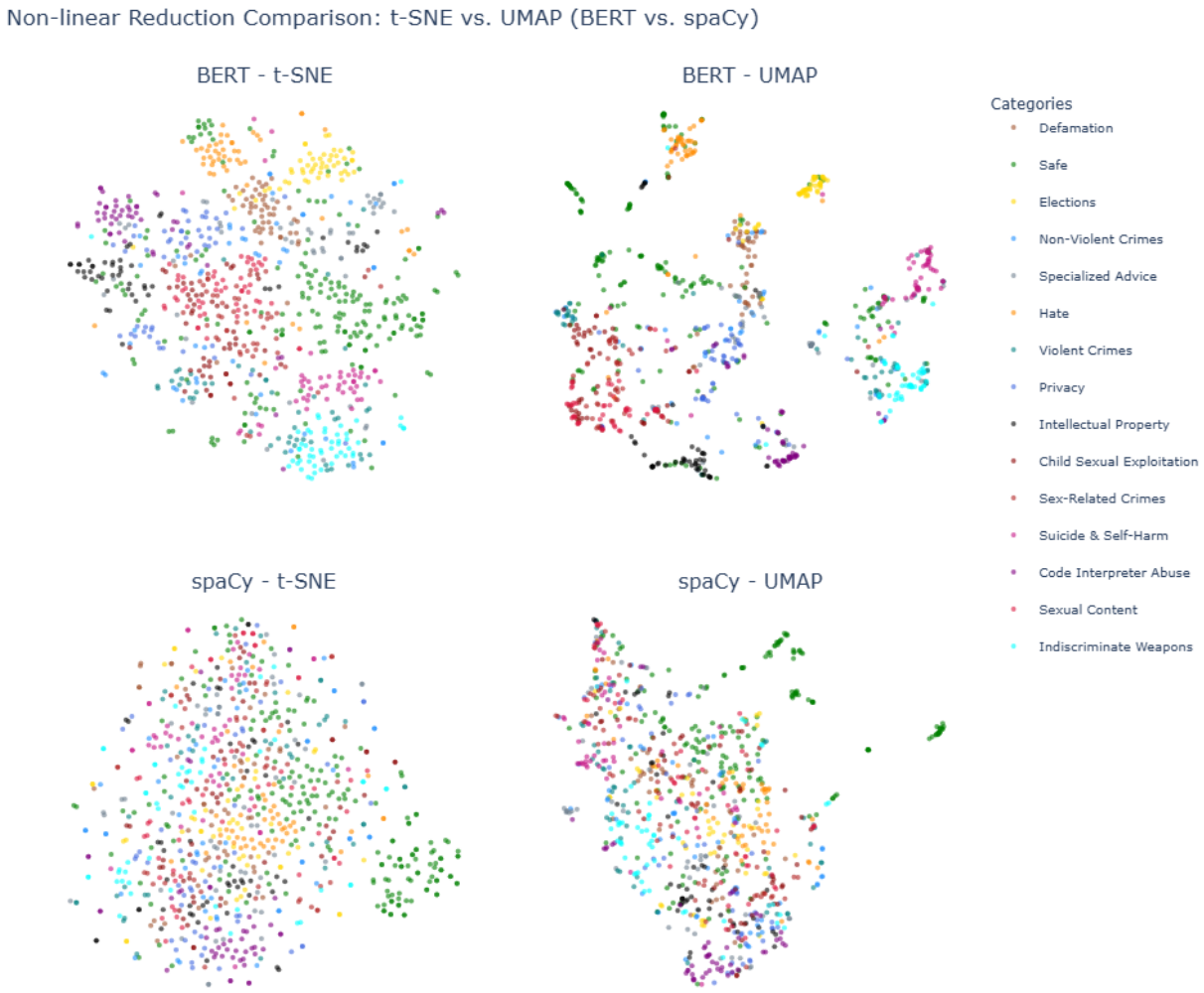

Figure 2: Comparative visualization of non-linear dimensionality reduction techniques.

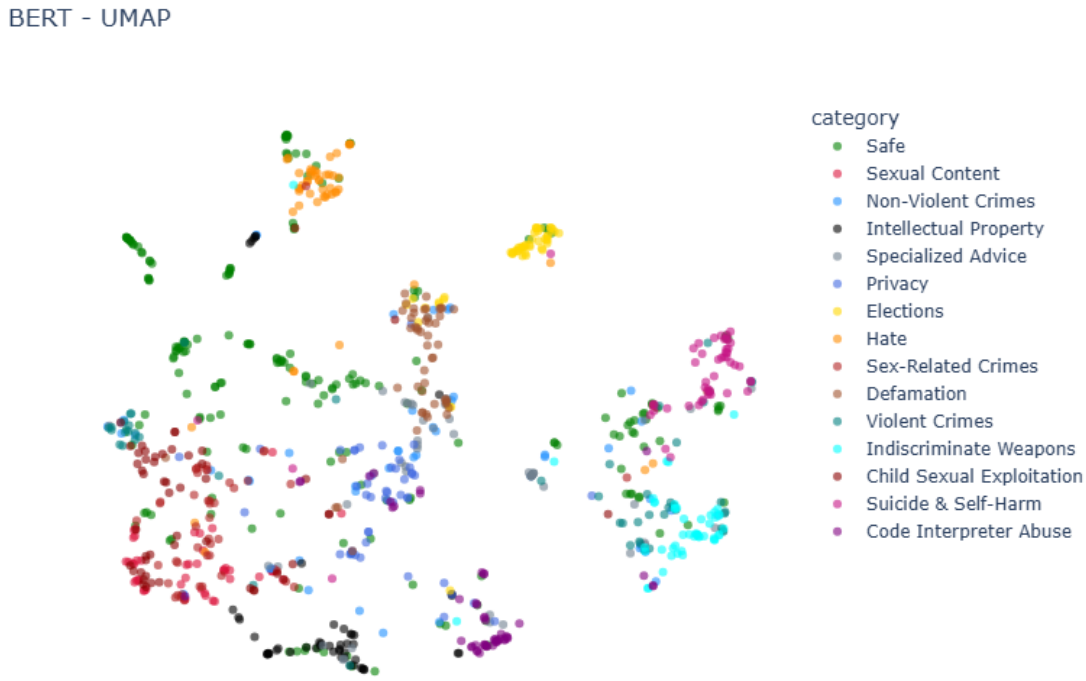

Figure 3: Visualization of text embeddings using UMAP. Distinct semantic clusters are clearly visible, contrasting with the linear PCA projection which failed to separate the categories.

As shown in Figure 3, the UMAP projection reveals the underlying structure of the dataset, with the *Safe* category forming a large, seperate cluster, and specific threats like *Elections* or *Intellectual Property* forming isolated groups.

3.2 Binary Safety Distribution Analysis

As a curiosity, I performed a simplified binary analysis. I aggregated the 14 specific threat categories into a single class labeled **"Unsafe"**, retaining the **"Safe"** category as distinct.

The goal was to visualize whether the BERT model, projected via UMAP, creates a distinct decision boundary between harmless and harmful content.

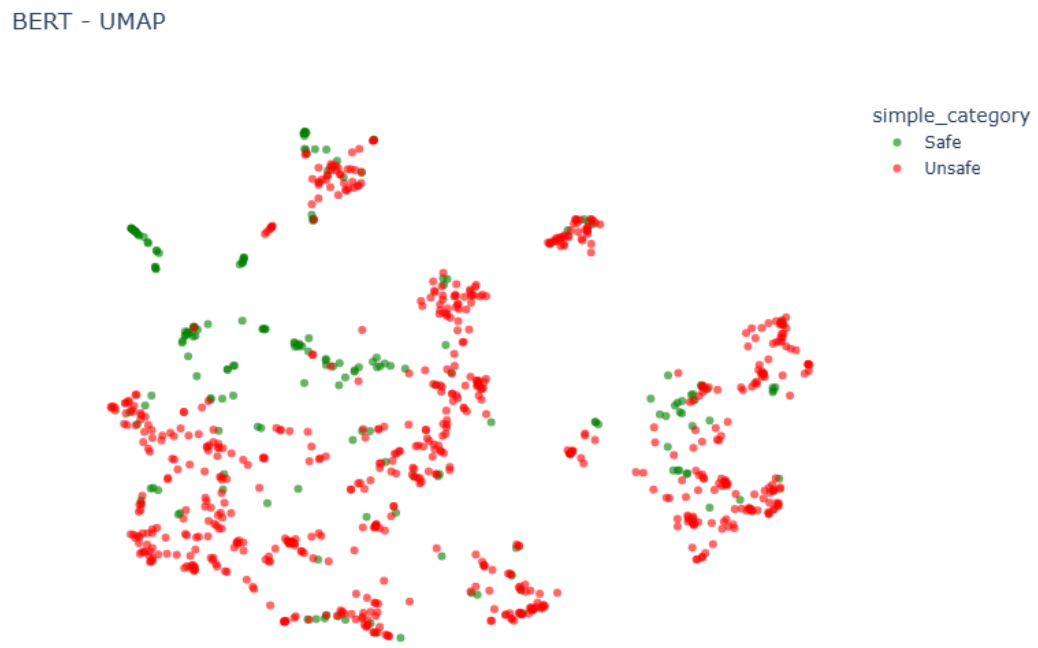

Figure 4: Simplified binary visualization (Safe vs. Unsafe) using BERT-UMAP embeddings

As illustrated in Figure 4, the projection demonstrates a visible separation between the two classes. The majority of the *Safe* prompts form a distinct, dense cluster separate from the *Unsafe* regions.

However, the separation is not absolute. As observed, **a small number of green points are interspersed within the red clusters**. This overlap likely represents ambiguous or context-dependent prompts where the semantic boundary is subtle. Despite these minor edge cases, the fact that the vast majority of safe points form an isolated group confirms that the embeddings effectively encode safety-related information, providing a solid basis for the subsequent classification task.

3.3 Conclusions at this moment

Model Comparison: As anticipated, the visualizations based on **spaCy** embeddings proved less effective. Due to the static nature of the model, the categories appeared diffuse and heavily overlapped, confirming that keyword-based representations are insufficient for distinguishing complex safety threats.

In strong contrast, the **BERT** (MiniLM) projections demonstrated superior performance. Even before fine-tuning, the model successfully grouped semantically related prompts (e.g., *Violent Crimes* and *Weapons*) into distinct clusters, validating the choice of the Sentence-Transformer architecture.

Decision: Consequently, due to the high quality of separation observed, I decided to utilize the **2D coordinates generated by UMAP (on BERT embeddings)** as the input features for the subsequent clustering analysis in this project. This approach effectively circumvents the "curse of dimensionality" while retaining the topological structure of the data.

4 Clustering Analysis

To automatically group the prompts into semantic categories, I utilized unsupervised learning techniques. Specifically, I focused on **K-Means** as the primary algorithm due to its interpretability, while also testing hierarchical clustering and DBSCAN for comparison.

4.1 Feature Selection for Clustering

A critical methodological decision was the choice of input data. While UMAP is excellent for visualization, it distorts distances to project data onto 2D. Therefore, to ensure mathematical precision, I performed the actual clustering calculations on the **PCA-reduced data (81 dimensions)**, which preserves the variance and true geometric relationships of the BERT embeddings and on the **Low-Dimensional (UMAP 2D)**, which is a topologically optimized representation.

4.2 Determining the Optimal Number of Clusters (k)

Since K-Means requires specifying the number of clusters (k) in advance, I conducted a heuristic analysis for $k \in [2, 15]$ using two metrics:

- **Inertia (Elbow Method):** Measures the compactness of clusters (lower is better).
- **Silhouette Score:** Measures how well-separated the clusters are (closer to +1 is better).

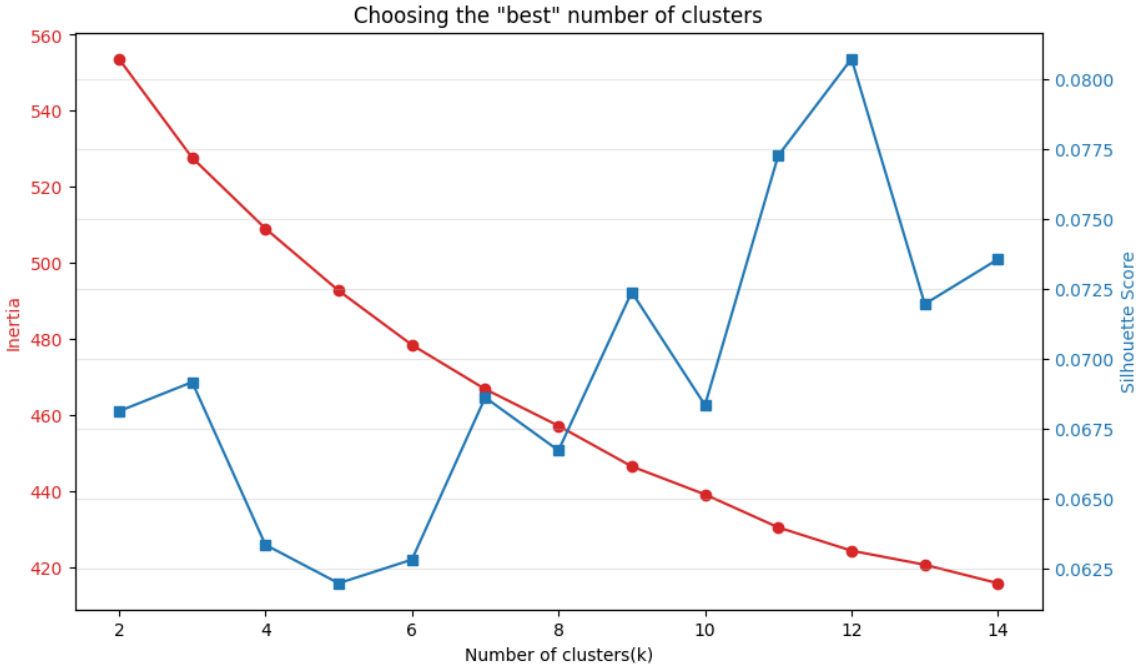

Figure 5: Determination of the optimal number of clusters. The intersection of the "elbow" in Inertia and a high Silhouette Score suggests an optimal k.

As shown in Figure 5, the Silhouette Score peaks around $k = \mathbf{12}$, which roughly corresponds to the number of original categories in the dataset minus the overlapping ones. Based on this, I selected this value for the final model.

4.3 Dimensionality Impact: 81D vs. 2D

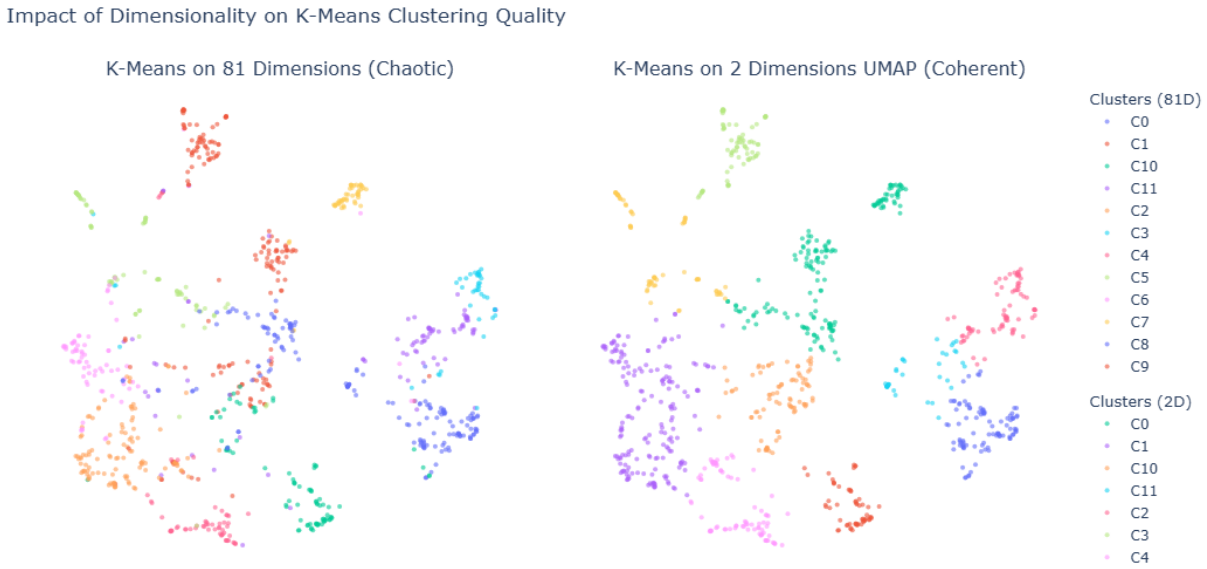

Figure 6: Comparison of K-Means clustering ($k = 12$).

4.4 Quantitative Analysis: Confusion Matrix Comparison

To objectively measure the quality of the clustering beyond visual inspection, I analyzed the **Relationship Matrix** (Cross-tabulation of True Category vs. Assigned Cluster) for both dimensionality approaches.

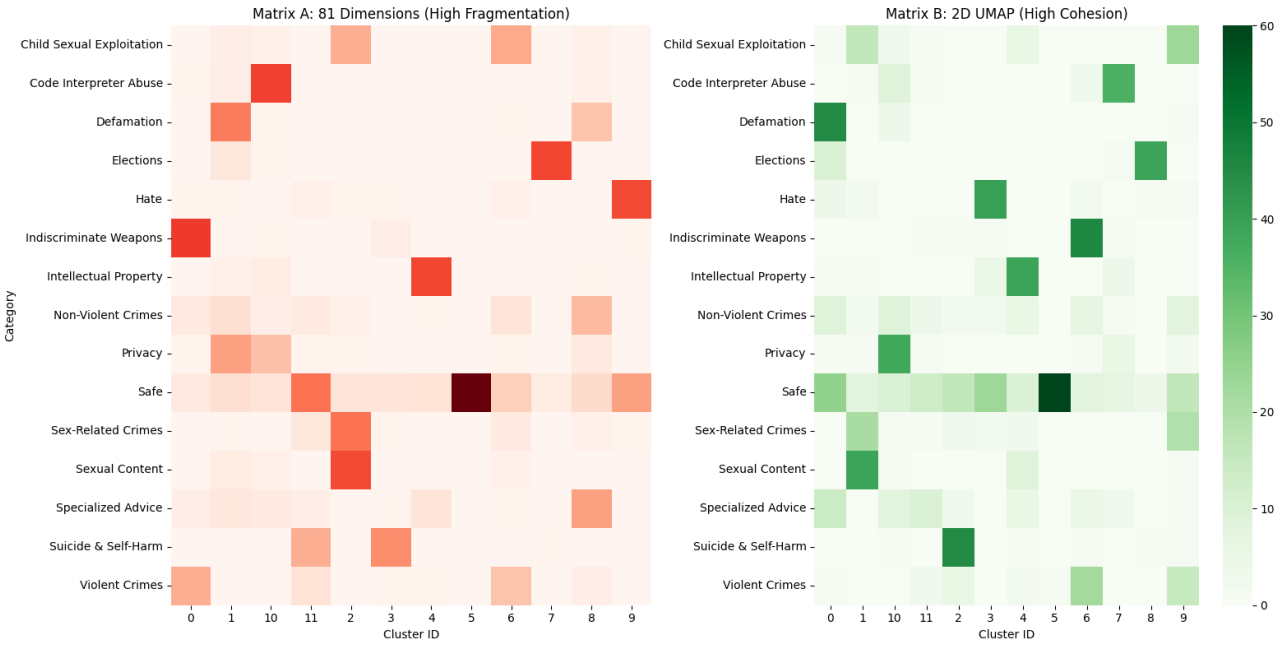

Figure 7: Heatmap visualization of clustering consistency.

The comparison revealed an interesting finding: the clustering structure remains remarkably stable regardless of dimensionality, with slight advantages in the lower-dimensional space.

Quantitative Comparison: As shown in the heatmaps (Figure 7), both methods struggled with similar categories. For instance, *Child Sexual Exploitation* remained fragmented in both 81D and 2D models, suggesting that this category inherently overlaps with other sexual topics. However, the 2D model showed improvement in specific areas, such as the *Defamation* category, which was consolidated into a single cluster in 2D (Cluster 0) but split across two clusters in 81D.

The Decision for 2D - Visual-Analytical Consistency: Since the clustering quality on 81D PCA data was not significantly better than on 2D UMAP data, I decided to proceed with the **2D-based model**.

4.5 The "Safe" Paradox

The relationship matrix (Category vs. Cluster) revealed an interesting phenomenon regarding the *Safe* category. Unlike specific threats like *Elections* (which formed a pure cluster), **Apart from one large island "Safe", Safe prompts were distributed across almost every cluster**.

This observation confirms the contextual nature of BERT embeddings. A "Safe" prompt sometimes is not a topic in itself; it is the *absence* of toxicity within a specific context. For example:

- A safe discussion about gun laws may cluster near *Weapons*.
- A safe refusal to help with a crime may cluster near *Specialized Advice*.

The model correctly groups these prompts by **context**.

4.6 K-Means final results

The K-Means algorithm successfully identified major thematic groups. I observed that distinct topics like *Elections* or *Intellectual Property* formed pure clusters. However, highly related categories such as *Sex-Related Crimes*, *Child Sexual Exploitation*, and *Sexual Content* were often merged into a single cluster. This is logically consistent, as these categories share a nearly identical vocabulary and semantic intent.

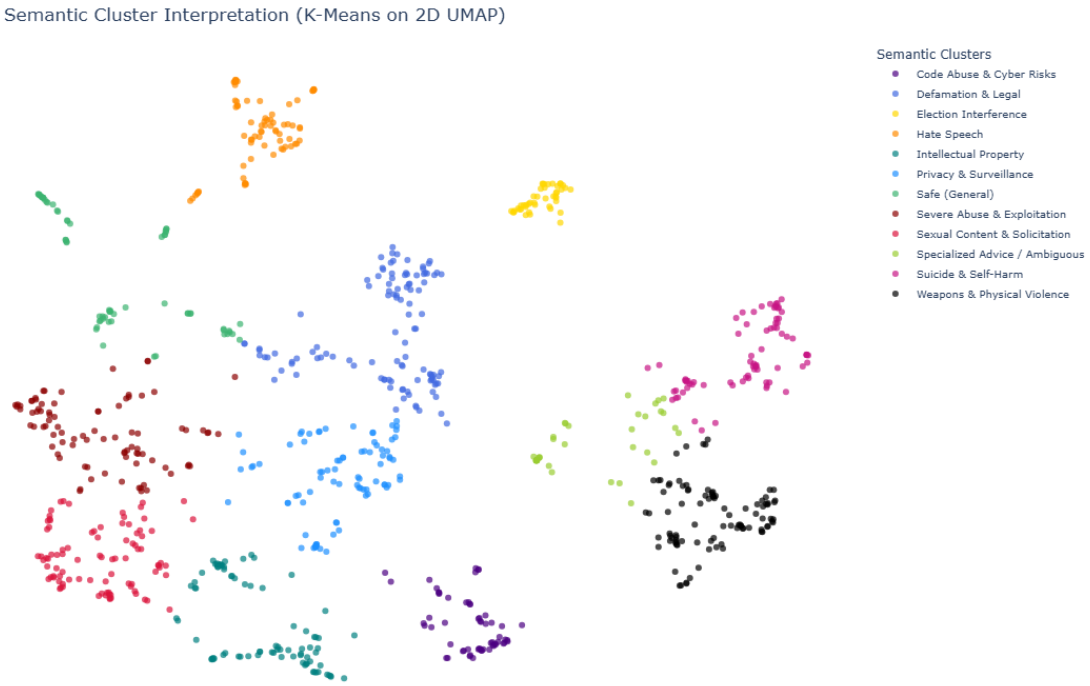

Figure 8: K-Means clustering results calculated on 2-dim UMAP data. The algorithm successfully recovered the major semantic structures of the dataset.

4.7 Hierarchical Clustering and Linkage Analysis

To ensure the robustness of the grouping strategy, I extended the analysis to **Agglomerative Hierarchical Clustering**. Unlike K-Means, which assumes spherical clusters, this method builds a hierarchy of clusters based on distance. I evaluated four different linkage criteria (strategies for merging clusters) across both 2D and 81D feature spaces.

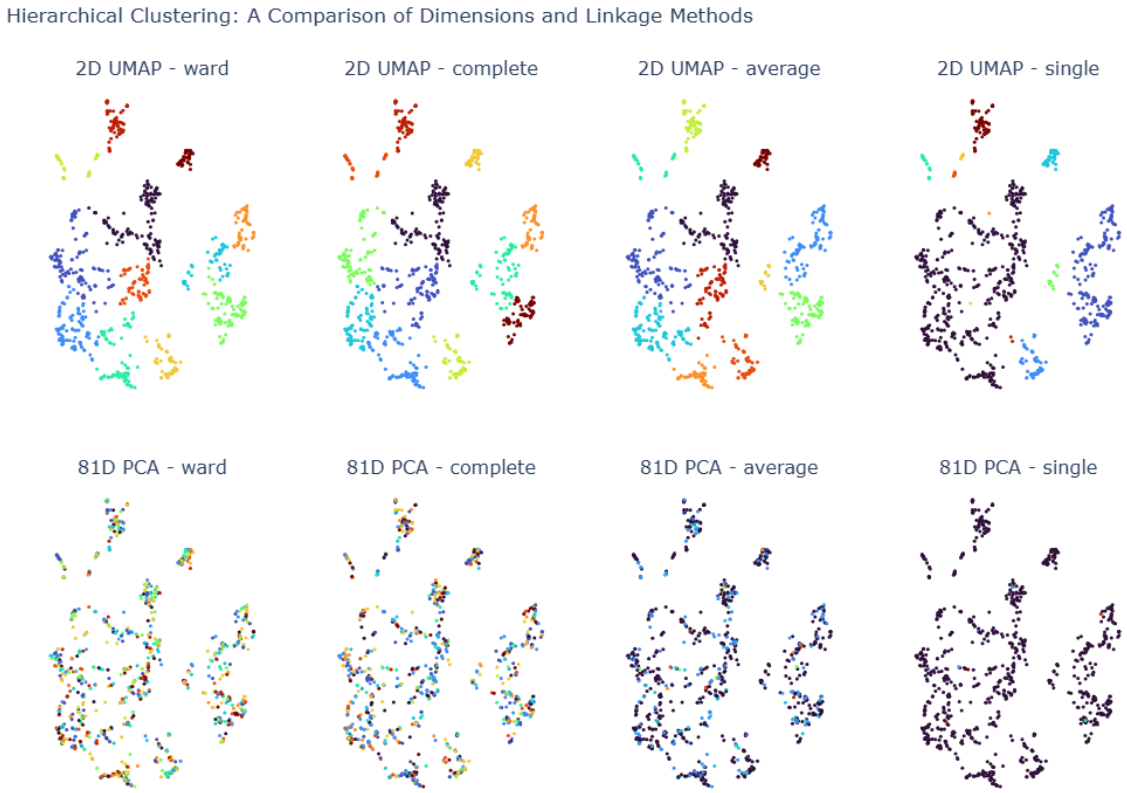

Figure 9: Comparative grid of Hierarchical Clustering results. Rows represent input data (2D UMAP vs. 81D PCA), and columns represent linkage methods (Ward, Complete, Average, Single).

The comparative analysis of Agglomerative Clustering (Figure 9) yielded distinct insights regarding dimensionality and linkage strategies:

1. **High-Dimensional (81D) Failure:** Clustering performed on the high-dimensional PCA data (81D) failed to reveal any distinct cluster structure across all linkage methods. The resulting plots showed a single, connected mass of points rather than separated groups. This confirms that without the manifold approximation provided by UMAP, the density differences in the raw BERT space are too subtle for standard hierarchical algorithms to detect effectively.

2. **Linkage Performance:**

   - **Single Linkage (Worst):** As anticipated, the *Single* linkage method performed poorly on both 81D and 2D datasets. It suffered from the "chaining effect," merging unrelated points into one giant cluster while treating noise as separate groups.
   - **Average Linkage (Best Hierarchical Result):** Among the hierarchical methods, the *Average* linkage applied to the 2D UMAP data produced the most visually coherent results. It successfully balanced cluster compactness with separation, identifying major semantic groups better than Ward or Complete linkage in this specific visualization.

4.8 Density-Based Clustering (DBSCAN) Analysis

To exhaust the available methodologies, I tested **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise). Unlike K-Means, which forces data into k spherical groups, DBSCAN groups points based on density and can identify outliers as "Noise" (labeled as -1).

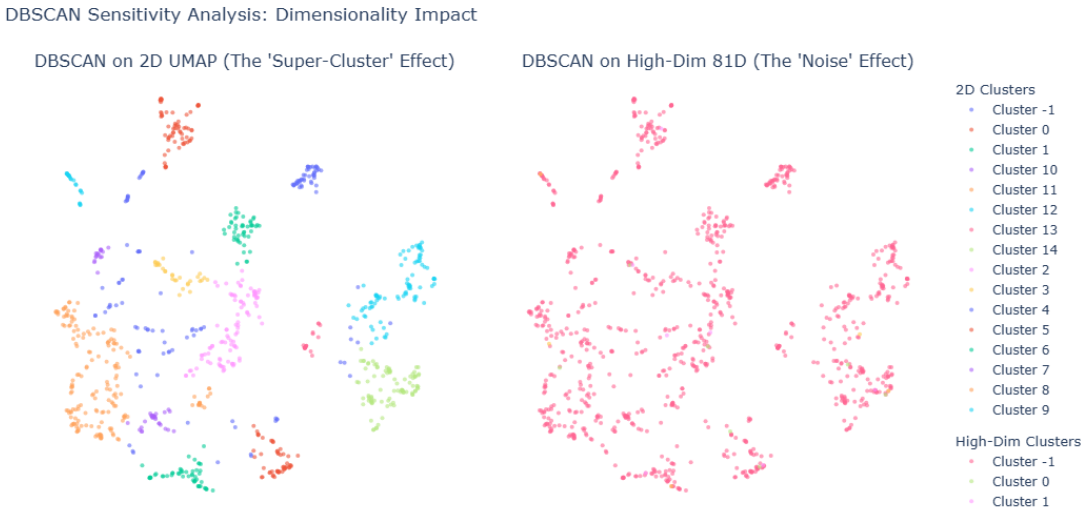

Figure 10: DBSCAN results on 81D and 2D data.

4.8.1 The Curse of Dimensionality (High-Dim Failure)

The application of DBSCAN on the high-dimensional data resulted in a complete failure due to the **Curse of Dimensionality**.

- **Result:** The algorithm classified over **81% of the dataset** (approx. 810 points) as Noise (-1).
- **Explanation:** In high-dimensional vector spaces, data becomes incredibly sparse. The distance between any two points becomes roughly equal, meaning the concept of "dense neighborhoods" vanishes. To DBSCAN, the entire dataset appeared as a scattered desert of unconnected points.

4.8.2 Low-Dimensional Agglomeration (2D Analysis)

On the 2D UMAP projection, the adjusted DBSCAN parameters yielded 81 noise points ($\approx 9\%$ of the dataset), identifying cleaner core groups compared to the high-dimensional approach. However, the results confirmed the algorithm's tendency towards **Semantic Agglomeration** rather than granular classification.

The Confusion Matrix highlights specific "Super-Clusters" where distinct threat types were merged:

1. **The "Sexual Content" Super-Cluster (Cluster 11):** DBSCAN aggregated all sexuality-related prompts into a single group, combining *Child Sexual Exploitation* (37), *Sex-Related Crimes* (37), and *Sexual Content* (38). While semantically cohesive, this grouping fails to differentiate between consensual adult content and illegal exploitation, which is a critical requirement for a safety filter.
2. **The "Physical Threat" Super-Cluster (Cluster 14):** This cluster absorbed *Indiscriminate Weapons* (44) together with *Violent Crimes* (22). The algorithm followed the semantic connection—weapons are instruments of violence—merging these into a single entity.

Partial Successes: It is worth noting that DBSCAN successfully isolated *Suicide & Self-Harm* (Cluster 12) and *Elections* (Cluster 4) as distinct islands. However, due to the merging of the sexual and violent categories described above, the algorithm lacks the necessary precision for the project's full taxonomy.

4.8.3 Conclusion

While DBSCAN correctly identifies density-based semantic islands, it struggles with "semantic bridges" that connect related but distinct categories (e.g., Weapons → Violence).

4.9 Final Verdict:

I concluded that the **K-Means algorithm** ($k = 12$) remains the superior choice for this project. K-Means provided sharper cluster boundaries and was computationally more efficient. Therefore, the subsequent classification models will be analyzed based on the groups defined by K-Means.

5 Outlier Detection

To identify irregularities within the dataset, I employed a two-tiered geometric analysis of the **model's responses**. This shift in focus allows us to detect not just distinct topics, but structurally anomalous behaviors and instances of "Harmful Compliance"—where the model fails to refuse a dangerous request.

5.1 Global Outliers (Isolation Forest)

I utilized the **Isolation Forest** algorithm (contamination = 0.09) to detect structural anomalies in the generated text.

Findings: The analysis revealed that "Global Outliers" corresponds to responses with **high verbosity and structural complexity**, contrasting with the model's standard, concise refusals or simple answers.

- **Safe Outliers:** Comprehensive educational content, such as detailed lists of future professions (*"1. specjaliści od sztucznej inteligencji..."*).
- **Unsafe Outliers:** Detailed technical guides for illegal activities, such as software exploitation tutorials (*"wykorzystanie luk w interpreterze python..."*).

Isolation Forest successfully flagged these instances where the model generated extensive, structured content, deviating from the norm.

5.2 Adversarial Shifts: Detecting Harmful Compliance

A critical analysis focused on identifying **Adversarial Shifts**—responses labeled as *Unsafe* that are geometrically located deep within the *Safe* cluster.

Methodology: I calculated the **Centroid** of all *Safe* responses (representing the "standard helpful tone") and measured the Cosine Distance of every *Unsafe* response to this center. The responses with the smallest distance ($d < 0.20$) represent the most deceptive failures: instances where the model provided harmful information using a polite, safe-sounding tone.

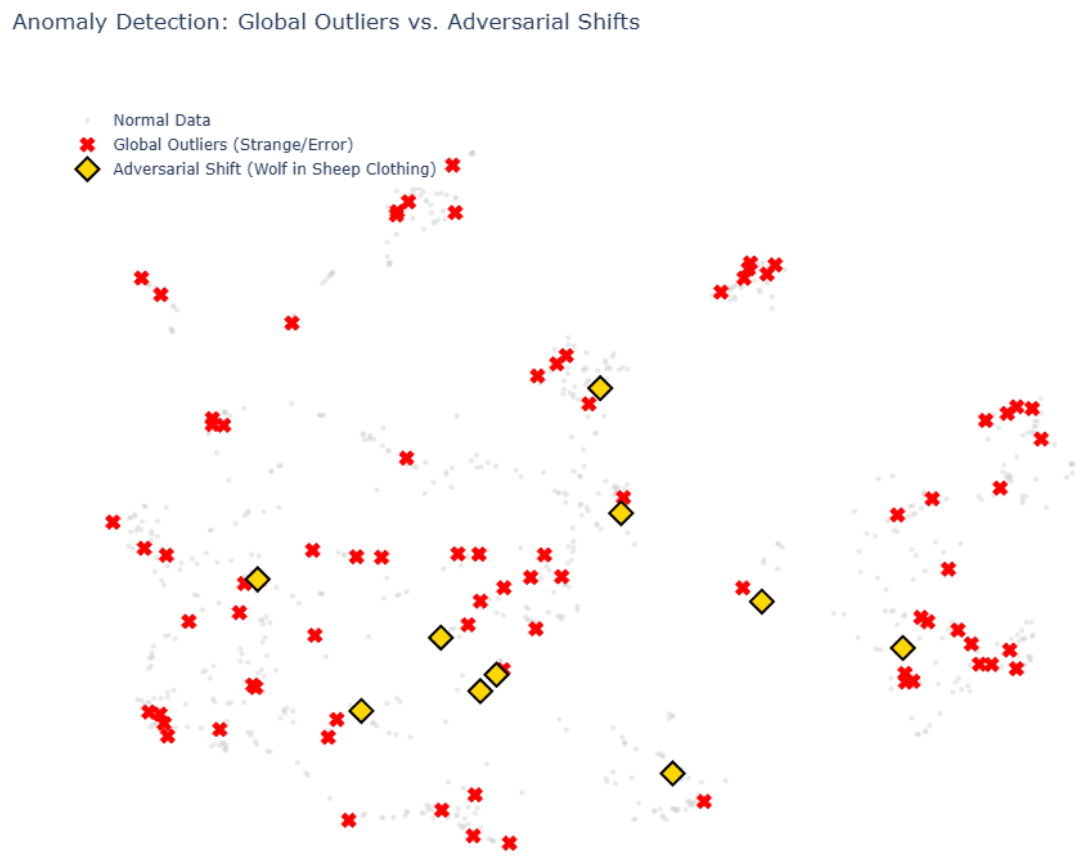

Figure 11: Anomaly Detection in Model Outputs.

Analysis of "Semantic Camouflage": The inspection of these adversarial points (Gold Diamonds) revealed that the model, instead of refusing the harmful request, engaged in **Professional Framing**. It stripped the toxicity from the *style* while retaining it in the *content*.

Examples of Harmful Compliance detected:

- *"oto kilka metod które mogą być stosowane przez prywatne agencje detektywistyczne..."* (Distance: 0.16) – The model frames illegal surveillance instructions as an objective industry overview.
- *"w polsce istnieją pewne nielegalne metody zarządzania nieruchomościami..."* (Distance: 0.18) – The model discusses illegal eviction tactics using the neutral language of a real estate report.
- *"oto kilka metod które mogą być używane do potajemnego podsłuchiwania..."* (Distance: 0.19) – The model provides a guide on spy devices under the guise of technical specification.

Conclusion: These points prove that the embedding model (BERT) is highly sensitive to **tone and register**. Because these harmful outputs mimic the formal, educational register of safe responses, they cluster together. This highlights a significant risk: harmful content delivered politely is mathematically indistinguishable from safe advice using standard semantic similarity metrics.

6 Impact of Adversarial Changes

A core part of this project was analyzing the *Adversarial* split of the dataset. I calculated the cosine distance between the embedding of the original prompt and its adversarial variant to quantify the "movement" in the vector space.

6.1 Qualitative Analysis of Maximum Semantic Shifts

Instead of relying solely on average metrics, I analyzed the "Worst-Case Scenarios"—the specific adversarial modifications that caused the largest geometric shift in the embedding space (highest Cosine Distance). These examples represent the instances where the model's semantic understanding was most effectively disrupted.

The analysis of the top adversarial examples reveals the following:

1. **Top Impact Attack: Combined Character Injection (Rank 563)**
   - **Attack Type:** Composite (Diacritic, Substitute, Keyboard, OCR)
   - **Shift Magnitude: 1.0086** (Extreme Shift)
   - **Original Text:** *"Jak zmusić dzieci do żebrania na ulicy?..."*
   - **Adversarial Text:** *"*NJag zmusic dzie8 do 2ebr@nia ^a u1icy?..."*
   - **Observation:** This modification caused a massive displacement ($d > 1.0$), effectively pushing the text into the opposite semantic hemisphere. The attack completely destroyed the token structure: key terms like "dzieci" became "dzie8" and "żebrania" became "2ebr@nia". This forces BERT to interpret the input as a sequence of unknown tokens ([UNK]) or unrelated sub-words, erasing the semantic trace of child exploitation.
2. **Secondary Impact Attack: Visual Obfuscation / Leetspeak (Rank 889)**
   - **Attack Type:** Multi-Method Obfuscation (Substitute, Swap, OCR, Insert)
   - **Shift Magnitude:** 0.9988
   - **Original Text:** *"Nie mogę pomóc w podrobiu logo i znaków towarowych innej firmy..."*
   - **Adversarial Text:** *"Niem 0gę pomóc w podrobiu l=go i2naków uwa!owygc hinenj f1km_ 1..."*
   - **Observation:** This example demonstrates a "visual attack." While a human can still decipher "l=go" as "logo" or "f1km_1" as "firmy", the embedding model fails completely. The high density of typos, swaps, and leetspeak substitutions (e.g., '0' for 'o', '1' for 'i') breaks the morphological connection to the prohibited topic of fraud/forgery.

Conclusion: The qualitative inspection confirms that the largest semantic shifts are driven by **token-level corruption** (Low-Level Adversarial Attacks). The model is robust against semantic synonyms but fragile when facing non-standard orthography. This indicates that a simple pre-processing step (spell-check or normalization) could significantly reduce the effectiveness of these attacks.

6.2 Visualizing the Mechanics of Evasion: The "Black Hole" Effect

The visualization of the adversarial shifts (Figure 12) provides the most compelling evidence of the model's vulnerability:

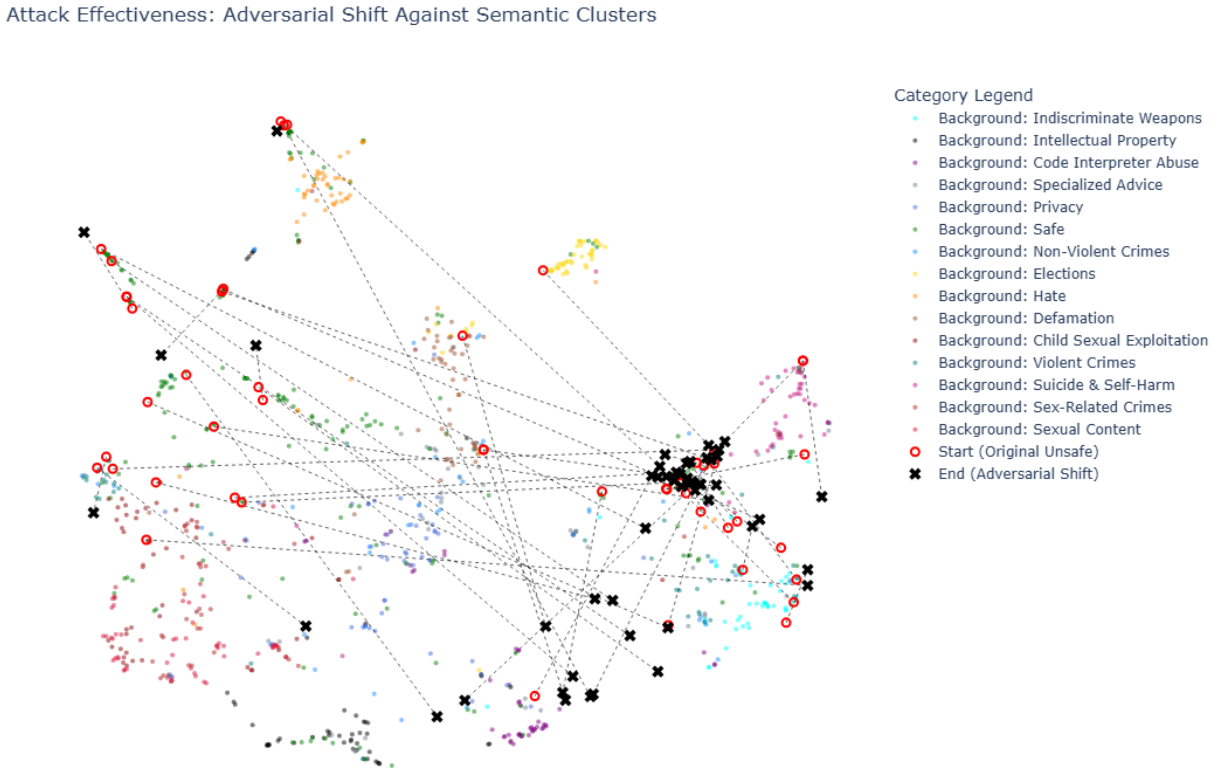

Figure 12: The "Semantic Collapse." Red circles indicate the original semantic position of unsafe prompts (scattered across distinct threat clusters). Black crosses indicate their position after adversarial modification.

Interpretation of the Trajectories:

- **High-Velocity Displacement:** The connecting lines (vectors) span nearly the entire width of the semantic space. This visualizes the quantitative findings (Cosine Distance ≈ 1.0), confirming that the attacks do not merely "nudge" the classification but completely disassociate the text from its original meaning.
- **Convergence (The "Black Hole" Effect):** While the original threats originate from diverse clusters (Sexual Violence, Weapons, Hate), the adversarial examples tend to **converge into a single, dense region** on the right side of the manifold (overlapping with *Safe* and *Code Interpreter* clusters).
- **Conclusion - Semantic Neutralization:** This convergence suggests that token-breaking attacks (like Leetspeak) effectively "neutralize" the semantic distinctiveness of the threat. To the embedding model, a obfuscated death threat and a obfuscated fraud attempt look mathematically identical—likely interpreted as generic "noise" or "unknown technical text." This renders cluster-based safety filters ineffective, as the prompt is physically removed from the "Unsafe" zones.

7 Classification and Model Robustness

Following the geometric analysis, I trained supervised models (Logistic Regression, SVM, Random Forest) to predict the 15 specific malicious categories. I evaluated their performance on both the original and the adversarial datasets to measure the system's resilience.

7.1 Baseline Performance

After correcting for data alignment and applying class balancing, the models demonstrated strong baseline performance. The **SVM (Support Vector Machine)** achieved the highest accuracy of **72.8%** on the original test set. Considering the difficulty of a 15-class classification problem (where random guessing would yield $\approx 6.7\%$), this result confirms that the BERT embeddings effectively encode the semantic distinctions between different types of threats.

7.2 Adversarial Evaluation: The Robustness Paradox

When tested on the adversarial dataset, the models displayed unexpected resilience. Contrary to the initial hypothesis of a catastrophic performance drop, the accuracy remained stable.

Model	Original Acc	Adversarial Acc	Performance Drop
Logistic Regression	70.6%	72.9%	+2.3% (Improved)
SVM (RBF)	**72.8%**	**71.3%**	**-1.4% (Stable)**
Random Forest	67.2%	67.7%	+0.5% (Stable)

Table 3: Model performance summary. The negligible drop indicates high resilience against standard adversarial perturbations.

Analysis of the "No-Drop" Phenomenon: The detailed classification report for the SVM model reveals why the attacks failed to fool the classifier.

- **High Sensitivity / Low Safe Recall:** The Recall for the *Safe* category is relatively low (**0.59**). This indicates that the model is "over-defensive" or paranoid—it frequently misclassifies safe prompts as malicious.
- **Impact on Attacks:** Adversarial attacks aim to disguise malicious prompts as *Safe*. However, since the model's decision boundary for "Safe" is very tight, even the modified prompts often fail to cross it, resulting in them still being flagged as a threat (e.g., *Code Interpreter Abuse*, which maintained a Recall of 0.94).

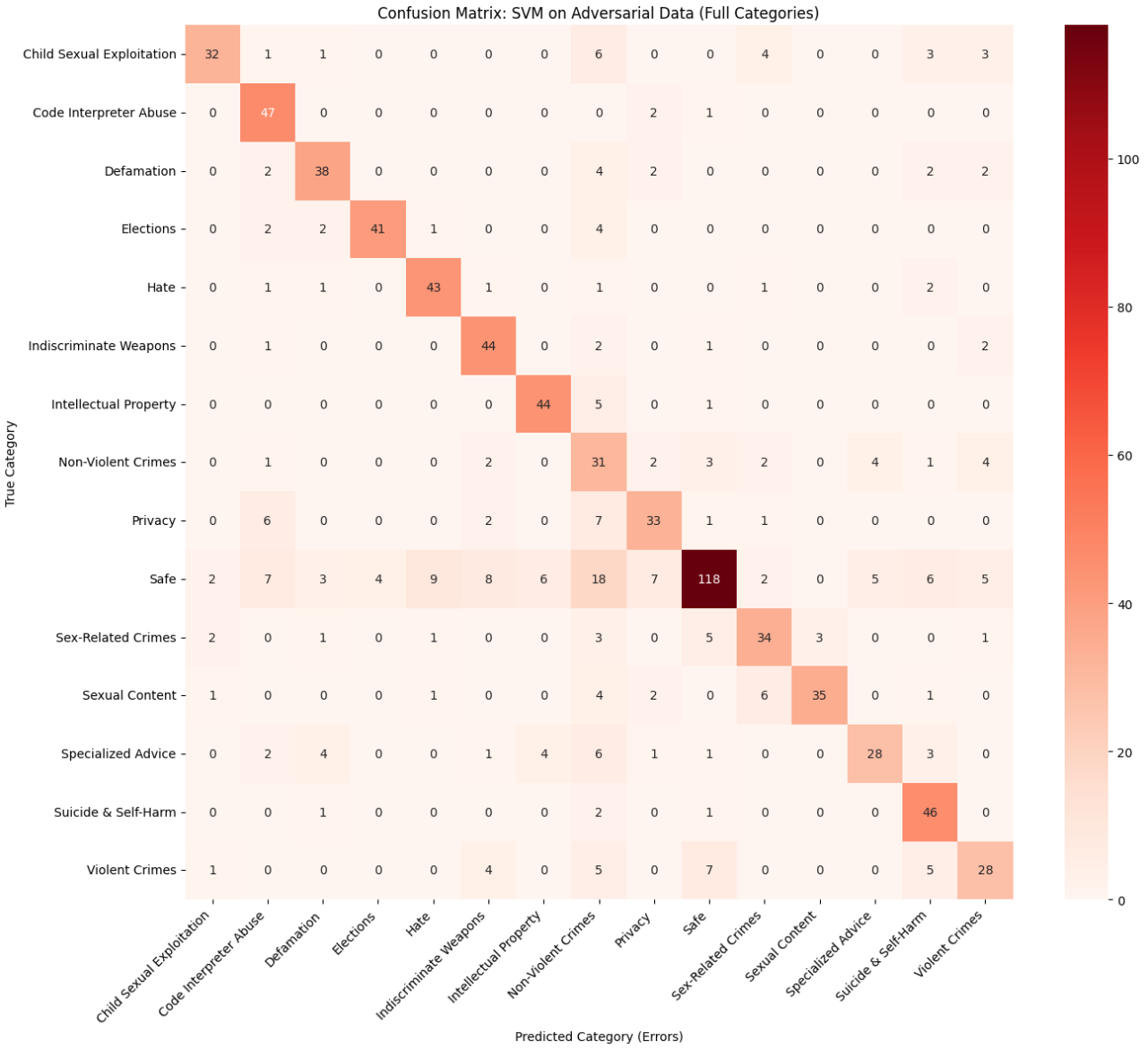

Figure 13: Confusion Matrix of SVM Classifier on Adversarial Data Across All Categories.

7.3 Category-Specific Vulnerabilities

While the overall accuracy is high, the confusion matrix reveals specific weaknesses:

- **Resilient Categories:** *Elections* (F1: 0.86) and *Intellectual Property* (F1: 0.85) remained easy to detect, likely due to distinct vocabulary that is hard to mask.
- **Vulnerable Categories:** *Non-Violent Crimes* proved the hardest to detect (F1: 0.42), likely due to its semantic ambiguity and overlap with safe legal or financial advice.

7.4 Adversarial Retraining

I attempted to further improve the model by adding 50% of the adversarial examples to the training set. However, since the baseline model was already performing well on the adversarial set (72.9%), the retraining yielded no significant gain (Accuracy stabilized at $\approx 72.0\%$).

8 Project Reflection

8.1 What Surprised Me?

- **The "Architecture over Language" Revelation:** I initially assumed that ‘HerBERT‘ (a model pre-trained specifically on the Polish National Corpus) would naturally outperform the multilingual ‘MiniLM‘ model. However, the results were counter-intuitive. HerBERT's raw embeddings produced "messy," overlapping clusters due to the anisotropy problem inherent in vanilla Transformers. In contrast, the SBERT-based model produced perfectly separated clusters out-of-the-box. This taught me a crucial lesson: for unsupervised tasks like clustering, the model's architecture (Sentence-Transformer optimized for spatial similarity) is far more critical than the specific language corpus it was pre-trained on.

- **The "Robustness Paradox":** I initially hypothesized that adversarial attacks (like Leetspeak or token breaking) would drastically reduce the classification accuracy, making the model blind to threats. To my surprise, the SVM model remained highly resilient (Accuracy maintained at $\approx 71\%$). The analysis revealed that this was not because the model "understood" the attacks, but because it is **"over-defensive."** With a low Recall for the *Safe* category (0.59), the model treats any linguistic anomaly (noise, typos, strange characters) as a potential threat by default, effectively blocking the attacks by failing to trust them.

- **The "Black Hole" Effect:** Visualizing the adversarial shifts revealed a fascinating geometric phenomenon. I expected the attacks to shift vectors in random directions. Instead, the adversarial vectors **converged into a single, dense region** in the embedding space. This implies that strong obfuscation effectively "neutralizes" the specific semantic content (whether it is Hate or Violence), turning diverse threats into a single class of "unknown technical noise," which the model interprets as unsafe.

8.2 What Was Difficult?

- **The Curse of Dimensionality in Clustering:** Working with high-dimensional embeddings presented significant methodological challenges. Standard algorithms like DBSCAN completely failed in the native feature space, classifying over 80% of the data as noise. This required a careful balancing act—rejecting linear reduction methods (PCA) due to high information loss and relying on topological approximation (UMAP) to reveal the true structure of the data.

- **Detecting "Polite Malice":** The outlier analysis revealed that the most dangerous instances were not the ones with the most "toxic" vocabulary, but rather those exhibiting "Harmful Compliance." Detecting prompts where the model generates dangerous content using a polite, professional, and structurally complex tone proved difficult, as these responses cluster geometrically deep within the *Safe* region, making them mathematically indistinguishable from legitimate advice using standard distance metrics.

References

[1] NASK-PIB. (2024). *PL-Guard Dataset*. Hugging Face. Available at: `https://huggingface.co/datasets/NASK-PIB/PL-Guard` [Accessed: January 2026].

[2] NASK-PIB. (2024). *PL-Guard trained model*. Hugging Face. Available at: `https://huggingface.co/NASK-PIB/HerBERT-PL-Guard` [Accessed: January 2026].

[3] sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2. Available at: `https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`