Bridging Language and 3D Assets: Comparing Embedding-Based and Contrastive Approaches for Text-to-3D Retrieval

Adam Rolander **Pablo Calderon** **Juan Fernández de Navarrete**
Gayathri Kuniyil **Arjun Kohli** **Ethan Cota**

Abstract

3D asset libraries have grown tremendously, making effective text-to-3D retrieval a valuable tool in 3D content creation. However, most asset retrieval systems struggle with the semantic variability of natural language queries. We address the question of whether an unsupervised embedding-based approach or a supervised contrastive approach yields better semantic alignment between text descriptions and corresponding 3D assets. We implement a baseline unsupervised retrieval pipeline using a pre-trained sentence encoder (MiniLM L6 V2) and compare its performance against a supervised multimodal model fine-tuned on text-image pairs using a contrastive learning objective. To evaluate the role of linguistic diversity, the supervised model was trained on both base (1 caption : 1 asset) and paraphrase-augmented datasets (11 captions : 1 asset, synthesized using Gemini-2.5-Flash). All models achieved perfect Recall@10 on original captions. However, evaluation under linguistic perturbations revealed key differences: the Supervised Base model suffered from reduced stability (R@10=0.923) and poor discrimination (Discrimination=1.00—maximum false positives), indicating severe overfitting. The Supervised Augmented model recovered perfect stability (R@10=1.00) and significantly improved discrimination (Discrimination=0.50), demonstrating the necessity of augmentation. Notably, the Unsupervised Baseline showed the highest discrimination power (Discrimination=0.25). Our findings conclude that unsupervised embedding-based retrieval is a robust baseline in low-data regimes, but supervised multimodal models, when trained with paraphrastic augmentation, achieve superior semantic grounding and stability necessary for practical text-to-3D search systems.

Code: https://github.com/AdamRolander/3D-Asset-Retrieval

Model Weights: https://huggingface.co/Aerolandaz/text_to_3D_retrieval/tree/main

1 Introduction

With the rapid expansion of 3D asset libraries, effective retrieval has become crucial. Most existing systems still rely on keyword matching, which fails to capture semantic relationships between terms. As an example, a search for “modern armchair” using current methods of 3D asset indexing is likely to overlook an asset with a label such as “contemporary lounge chair.” While these two queries are similar, simple keyword matching and recognition algorithms do not properly account for these types of semantic variation.

It is still uncertain whether unsupervised embedding-based retrieval or supervised fine-tuning outputs will create more reliable mappings between collections of 3D assets and their textual labels. We seek to implement both of these methods to answer this question by implementing these models and evaluating their performance across several quantitative and qualitative evaluation metrics. This

development will help to assess the potential of embedding-based retrieval in AI-focused 3D graphics generation and search pipelines.

This problem led us to the following approaches to determine which method results in the best semantic alignment between natural language text descriptions and their corresponding 3D asset:

1. Using a pre-trained text encoder (like BERT) to create a vector space for text descriptions of 3D assets and inferring similarity of novel text descriptions using embedding-based retrieval (unsupervised).
2. Jointly fine-tuning text and asset encoders using a supervised contrastive objective on paraphrased 3D asset descriptions to create a multimodal embedding space for asset retrieval from novel text descriptions (supervised).
3. Does paraphrastic data augmentation meaningfully improve the supervised model's performance?

2 Background

2.1 BERTScore

Zhang (2020) introduced BERTScore, a metric created to evaluate the performance of text generation systems. Their goal was to fix the problems of traditional metrics like BLEU, METEOR or ROUGE, which are dependent on n-gram overlap and fail to capture semantic meaning and context. Using contextual embeddings from BERT (Devlin et al., 2018), the model is able to compare candidate and reference text by computing the cosine similarity between their token embeddings. This produces standard evaluation metrics which account for the meaning, as well as the wording. The paper states that after experiments done across translation and summarization, BERTScore correlates more with human evaluations and still works well when the text is paraphrased, showing that it understands meaning, as well as wording.

Our project uses this idea by applying embedding-based retrieval to match the text descriptions to the corresponding 3D assets. This eliminates the constraints of previous solutions, allowing our model to understand context and

2.2 Contrastive Language-Image Pre-training

In a 2021 whitepaper by Alec Radford of OpenAI, researchers introduce Contrastive Language-Image Pre-training (CLIP) which allows models to learn relationships between raw text and paired images. As opposed to contemporary methods in computer vision, which required pre-determined object categories, and had limited generalization to novel tasks, CLIP aimed to create a shared latent space between text and images to allow broader contextual learning (Radford et al., 2021).

The researchers trained both an image encoder (via. ResNet and Vision Transformer) and a text encoder (via Transformer) on roughly 400 million image-text pairs with the goal of maximizing cosine similarity of the image-text embeddings that did occur in the dataset, while minimizing the cosine similarity of the embeddings that did not occur (Radford et al., 2021). CLIP was found to match or outperform state of the art approaches in image classification and action recognition with a zero-shot transfer.

CLIP's underlying concept of using natural language as a prior for multimodal representations is valuable to our task of using language to retrieve 3D assets. We seek to develop a shared latent space of asset representations and paraphrased versions of their text descriptions that maximizes similarity between each asset and its descriptions, while minimizing similarity between unrelated pairings.

2.3 Data Augmentation & Paraphrasing

Okur et al. (2022) introduced a data augmentation framework using paraphrase generation and entity extraction to enhance low-resource multimodal dialogue systems. Their goal was to address the data scarcity problem in intent classification and entity recognition tasks, where limited annotated data often leads to poor generalization. Using both neural paraphrase generation models and pattern-based templates, the authors created multiple paraphrased words that preserved semantic meaning while varying surface form. The augmented data was then used to retrain, resulting in a substantial increase in accuracy and F1-scores (from ~90.6 to ~99.4 in some cases).

This study is relevant for us because it demonstrates how data augmentation with paraphrasing can expand linguistic diversity and improve model robustness. In a similar way, our work explores whether fine-tuning text–3D mappings with paraphrased descriptions can yield better semantic alignment than unsupervised embedding retrieval, strengthening contextual understanding in text-based 3D asset search.

3 Methods

3.1 Data Provenance

We sourced our parallel dataset of 3D assets and captions from the Cap3D archive by Luo Tiange et al. on Hugging Face. It consists of 3D assets (.glb), 20 rendered images of each asset (.png), and textual metadata containing a description (.json), improving upon the original Objaverse dataset by AllenAI. While Objaverse was a powerful foundation for its scale of 3D assets, it lacks accurate and consistent text descriptions and is limited to caption-asset pairs. The Cap3D dataset expands upon Objaverse by providing enhanced descriptive captions, rendered views, and additional metadata for over 1.5M assets, making it the ideal source of data for our project (Tiange et al., 2024).

3.2 Data Preprocessing

We downloaded a 2-column CSV that maps unique asset identifiers (UIDs) to their corresponding caption for all 1,547,626 assets. However, due to compute limitations, we could only download image renders for the first 10,000 assets (~43GB). Additionally, we decided to use only 1/20 image renders per asset for the contrastive fine-tuning task. Using this data, we created a 3-column CSV of the UIDs for assets we had renders of, their captions, and a local path to the render.

For the contrastive fine-tuning task, we sought to compare results between a model trained on one caption per asset and a model trained on several paraphrases of each caption. To do this, we interacted with Gemini-2.5-flash via API to create 10 paraphrases of each of 10,000 original captions, resulting in 110,000 caption-render pairs in the dataset. The API call employed the prompt as described in Appendix A.1.

3.3 Embedding-Based Similarity

For the unsupervised retrieval baseline, we implemented a text only search pipeline using MiniLM sentence transformer (MiniLM L6 V2). Each caption in the filtered dataset was mapped to a 384 dimensional embedding that captures its semantic meaning. We applied L2 normalization since embedding magnitude can have a range of values, and normalizing the vectors makes them have the same overall scale. With normalized vectors, FAISS inner product search is essentially equivalent to cosine similarity. This lets us measure the semantic similarity between any two captions based on their closeness in the embedding space.

After constructing the embedding matrix, we indexed the unit length vectors using FAISS, a library useful in vector similarity search, to extract the nearest neighbors for a user's input. After receiving the user's query, we encode the user's input using MiniLM and L2 normalization so that the input maps to the same space as the embeddings. Next, FAISS outputs the captions that are the most similar to the user's query in the embedding space through computing the inner product. This results in an indexed list of captions that match the semantic text of the user's query. This method does not require training or fine tuning, but it is still a strong unsupervised baseline because of its semantic understanding capabilities learned by MiniLM.

3.4 Contrastive Multi-Modal Fine Tuning

For the supervised retrieval task, we implemented a contrastive learning objective on a text and image encoder and compared results between the base and augmented datasets. The text encoder model was the same as in the unsupervised embedding task (MiniLM L6 V2), and the image encoder model was ViT-B/16 by OpenAI. The default image encoder had an output dimensionality of 768, so a linear layer was used to map its outputs to the 384-dimensional space of the text encoder. The text encoder output was aggregated with mean pooling.

Prior to training, the text captions were tokenized with the MiniLM L6 V2 tokenizer and were padded or truncated to a max_length of 128 tokens. The images were preprocessed using ClipImageProcessor. The models were initially loaded with their default weights, then trained over 3 epochs with a batch size of 32, AdamW optimizer with a learning rate of 5e-5, and a learnable temperature parameter to scale logit scores. Both the text and image embeddings were L2-normalized prior to computing similarity for the symmetric cross-entropy loss. All training was performed on a single Apple M3 Pro chip with the 18-core integrated GPU and 36GB of unified memory.

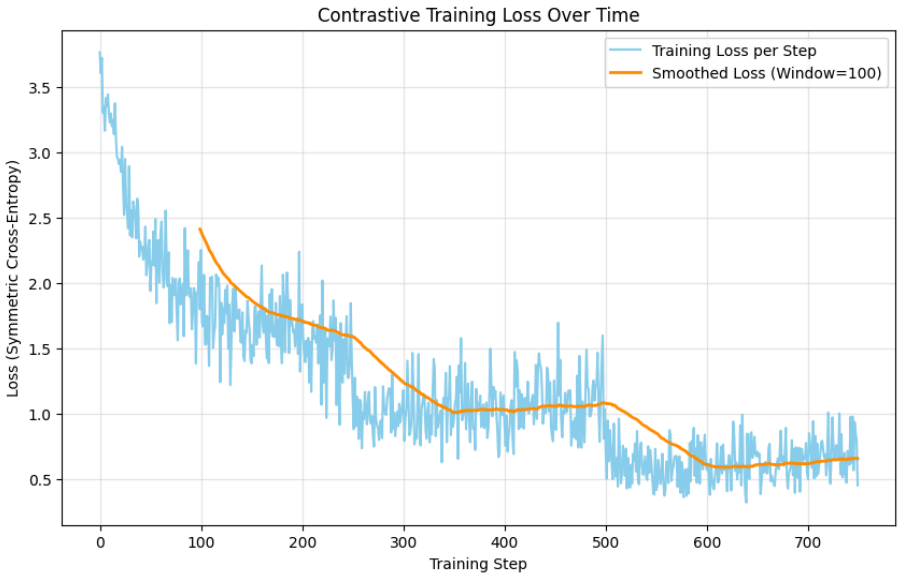

Figure 1: Loss Over Training Steps for the Non-Augmented Contrastive Model

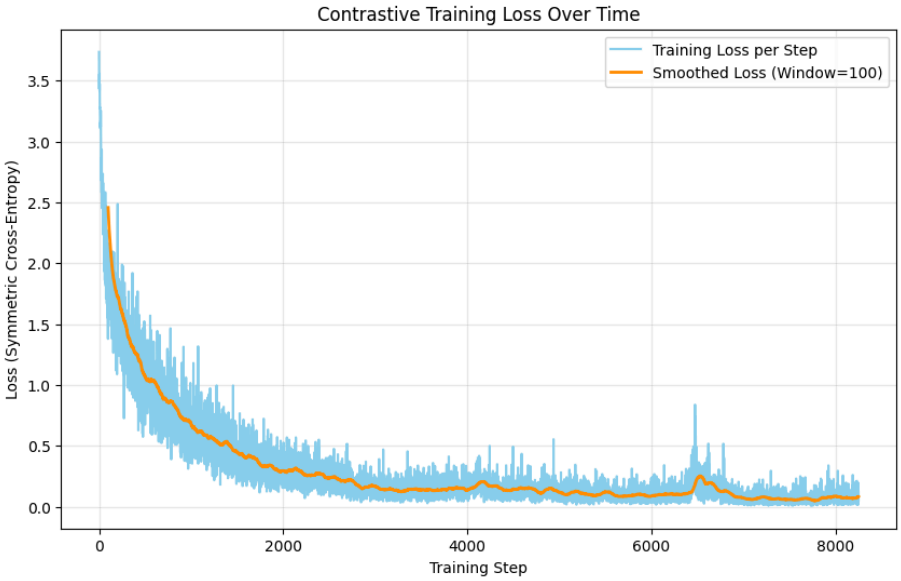

Figure 2: Loss Over Training Steps for the Augmented Contrastive Model

3.5 Evaluation

3.5.1 Evaluation of Embedding-based retrieval

We had two different tests/measures to evaluate the embedding based retrieval baseline. One of these measures was quantitative evaluation. For this, we sampled 100 captions from the dataset and then analyzed the caption as a query and compared this with the FAISS index.. A retrieval was correct if the actual caption was within the top k outputted results. For this, we had Recall@1,Recall@5, Recall@10, and MRR (Mean Reciprocal Rank), which measured how accurately the model returned the caption near

the top of the rankings. Higher Recall and MRR values mean a better retrieval performance since it analyzes how consistently the most similar embedding is in the top results.

Our second test was checking the robustness. For this, we analyzed how the retrieval results changed when we had paraphrased versions such as adding synonyms, word order changes, and adding more descriptive language. A strong model should return around the same similar closest neighbors regardless of slight variations in the input, since the semantic meaning is the same. From comparing the original versus the paraphrased input, we can analyze how prone the embedding space is to change/variations in language.

3.5.2 Evaluation of Contrastive Multi-Modal Fine Tuning approach

To assess the quality of the learned embedding spaces, we established a test on about 2,000 assets. For each test asset, we used its ground-truth caption as a query to retrieve matches from the full database of ~10,000 assets. All embeddings were L2-normalized, ensuring that ranking by inner product corresponds to cosine similarity.

We utilized three specific metrics to capture different dimensions of performance:

- General Accuracy (Recall@10): This measures the percentage of test queries where the correct asset appears in the top-10 results. It serves as a baseline check to ensure the model can perform basic retrieval when given a description.
- Stability: This metric evaluates robustness to linguistic noise. We generated variations of the test queries (including typos, shuffled word order, and synonyms) and measured the model's ability to maintain its retrieval performance. It is calculated by comparing the Recall@10 on reworded queries to the Recall@10 on the original queries. A score of 1.0 means the system handles rephrased queries just as well as the originals.
- Discrimination: This measures the model's ability to reject mismatched descriptions. We queried the models with text that does not describe the asset, and measured how often the model incorrectly retrieved the target asset anyway. A lower score is better, indicating the model successfully discriminated against the wrong input.

3.6 Adversarial Testing

We conducted an adversarial robustness analysis by systematically perturbing the text queries and observing the impact on retrieval. We crafted three categories of adversarial queries:

- Lexical Substitutions: Replace words with synonyms or near-synonyms, aiming to change specific terms without altering the overall meaning. This tests whether the model really understands the concept or is tied to a particular word.
- Syntactic Variations: Alter the grammar or word order of the description while preserving its semantics. This checks if the model is invariant to sentence structure and word arrangement.
- Semantic Distractions: Insert additional descriptors or irrelevant adjectives that do not change the core identity of the item. These perturbations introduce extra information or noise to see if the model can ignore it. The key meaning remains, but the modifiers should be discarded by the model ideally.

The generation of the adversarial examples was done using an automated paraphrasing script (Gemini-based GPT model) as seen in Appendix A.2, ensuring that the altered queries stayed plausible and semantically similar to the originals. Each original query in the test set was transformed into one or more perturbed versions, which we then issued to the retrieval models.
For evaluation, we measured how each model's performance changed under these perturbations, using metrics that capture both retrieval success and embedding stability: Recall@10 Drop, Cosine Similarity Drop (ΔSim) and Robustness Ratio (RR).

4 Results

Across the three models, the results are as follows:

- Unsupervised Baseline achieves a General R@10 of 1.00, a Stability score of 1.000, and a Discrimination score of 0.25.
- The Supervised Base model also attains a General R@10 of 1.00, but its Stability decreases to 0.923 while its Discrimination increases to 1.00.
- Supervised Augmented model matches the others with a General R@10 of 1.00, recovers a perfect Stability score of 1.000, and yields a Discrimination score of 0.50.

General Accuracy: As expected, all models had a perfect Recall@10 (1.00). When providing the exact ground truth caption, every model had the correct asset within the top 10 results.

Stability: The impact of training data becomes clear in the stability scores. The Unsupervised Baseline and Supervised Augmented models achieved perfect stability (1.000), consistently retrieving the correct asset despite typos or rephrasing. However, the Supervised Base model dropped to 0.923. This drop suggests that without augmentation, the Base model overfit to the specific phrasing in the training set, becoming worse at retrieving the correct assets when faced with minor linguistic variations.

Discrimination: The most significant difference was Discrimination (specificity).

The Supervised Base model failed this metric completely (1.00), retrieving the target asset 100% of the time even for irrelevant queries. This indicates severe overfitting, where the model ignored the actual content of the query, and just retrieved familiar assets. The Supervised Augmented model significantly improved this (0.50), reducing false positives by half. This demonstrates that training on paraphrased data taught the model to pay closer attention to the actual meaning of the text. The Unsupervised Baseline showed the highest specificity (0.25). This implies that, without augmentation, supervised models tend to loosen their criteria and match more aggressively, reducing their ability to filter out irrelevant results.

4.1 Adversarial Testing

The adversarial tests confirmed the earlier findings on model robustness. All models were very resistant to lexical substitutions and syntactic variations, maintaining almost perfect Recall@10 (RR ≈ 1.0) and stable embeddings. This shows that they understood query meaning rather than relying on exact wording.

The models were slightly affected by semantic distractions, showing a small drop in performance and higher embedding drift. However, they still retrieved the correct asset over 99% of the time, demonstrating strong robustness even in the presence of irrelevant modifiers.

5 Discussion and Conclusions

Our findings highlight a clear trade-off between supervised contrastive learning and unsupervised embedding-based retrieval for text-to-3D asset search. Despite their conceptual differences, all three models achieved perfect Recall@10 under ideal conditions, showing that 3D asset retrieval is straightforward when the query exactly matches the ground-truth caption. The key differences emerged once the queries were perturbed.

First, stability under linguistic variation differentiated the models. The Unsupervised Baseline and the Supervised Augmented model both maintained perfect stability, demonstrating that pre-trained text encoders and paraphrase-augmented fine-tuning effectively capture semantic meaning beyond surface form. In contrast, the Supervised Base model showed reduced stability, indicating overfitting to specific phrasing in the limited training captions.

Second, discrimination performance revealed the limitations of supervised contrastive fine-tuning without augmentation. The Supervised Base model frequently returned the target asset even for irrelevant queries, suggesting that it learned shortcut associations rather than true semantic alignment. Incorporating paraphrases substantially mitigated this issue: the Supervised Augmented model halved its false-positive rate, confirming that linguistically diverse training is essential for preventing multimodal overfitting. Notably, the Unsupervised Baseline exhibited the strongest discrimination, underscoring the robustness of pre-trained text embeddings in low-data regimes.

Adversarial testing reinforced these observations. All models handled lexical and syntactic perturbations with minimal performance loss, showing that they encode stable semantic representations. Semantic distractions introduced mild degradation, but the correct assets were still retrieved in nearly all cases (RR $\approx$ 1.0).

Overall, our results suggest three main conclusions.

(1) Unsupervised embedding-based retrieval is a strong and reliable baseline, especially when training resources are limited.

(2) Supervised contrastive fine-tuning requires paraphrase-based augmentation to avoid overfitting and to maintain robustness across linguistic variation.

(3) Augmentation meaningfully improves semantic specificity, enabling supervised models to outperform the baseline in controlled settings without sacrificing stability.

These findings point toward promising future directions: scaling up contrastive training with richer caption diversity, incorporating 3D-aware encoders rather than image proxies, and exploring hard-negative mining to further enhance discrimination. For practical 3D asset search systems - particularly those built under limited compute budgets - our study shows that embedding-based retrieval remains competitive, but supervised multimodal models can achieve superior semantic grounding when trained with sufficiently varied language.

References

D. Hendrycks and T. Dietterich. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *Proceedings of the 2019 International Conference on Learning Representations (ICLR)*. (Poster).

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of NAACL-HLT 2019, pages 4171–4186. Association for Computational Linguistics.

K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese. 2019. Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings. In *Proceedings of the Asian Conference on Computer Vision (ACCV 2018)* (Lecture Notes in Computer Science, vol. 11363), pages 100–116. Springer.

K. Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65. Association for Computational Linguistics.

Okur, E., Sahay, S., Nachman, L, 2022. Data Augmentation with Paraphrase Generation and Entity Extraction for Multimodal Dialogue System. arXiv Preprint arXiv:2205.04006.

Radford, A. et al. (2021) Learning transferable visual models from Natural Language Supervision, arXiv.org. Available at: https://arxiv.org/abs/2103.00020 (Accessed: 28 October 2025).

R. Jia and P. Liang. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2021–2031, Copenhagen, Denmark. Association for Computational Linguistics.

T. Gao, X. Yao, and D. Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pages 6894–6910, Online & Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tiange, L. (2024) CAP3D, Cap3D. Available at: https://cap3d-um.github.io/ (Accessed: 28 October 2025).

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. In Proceedings of the 2020 International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia.

V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, and W.-t. Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics.

A. Prompt Templates

A1.

```
PROMPT_TEMPLATE = f"""
You are a data augmentation assistant. Your task is to generate {NUM_PARAPHRASES} unique,
high-quality paraphrases of a 3D asset description.

Rules:
1. The paraphrases must be semantically identical to the original.
2. Do not add new information, objects, or attributes.
3. Do not remove key information.
4. Vary sentence structure and vocabulary.
5. Output *only* a valid JSON list of {NUM_PARAPHRASES} strings. Do not include any other text,
markdown, or explanations.

Original Description:
"{{caption}}"

JSON List:
"""
```

A2.

```
PROMPT_TEMPLATE = f"""
You are a robust data generation assistant. Your task is to generate
3 specific adversarial perturbations for a 3D asset description.
Original Description: "{{caption}}"
Generate exactly one variation for each of these three categories:
1. lexical_substitution: Replace words with synonyms or near-synonyms (e.g., 'armchair' -> 'lounge
chair').
2. syntactic_variation: Reorder phrases or modify sentence structure without altering semantics (e.g.,
'table made of glass' -> 'glass table').
3. semantic_distraction: Add mild modifiers or irrelevant adjectives that do not change the core object
(e.g., 'wooden chair' -> 'comfortable wooden chair').
Output *only* a valid JSON object with these three keys.
JSON Schema:
{{
  "lexical_substitution": "string",
  "syntactic_variation": "string",
  "semantic_distraction": "string"
}}
"""
```

B. LLM Usage Disclosure

Gemini 2.5 Flash was used to suggest sources for our literature review. We also used Gemini 2.5 Flash to recommend pretrained models from HuggingFace for each approach we plan on taking, and to clarify the overall pipeline for each process. Each source was manually visited and investigated after being recommended, and all writing was done by us (human) writers. The final proposal was verified for cohesion and technical accuracy with the same Gemini model.

Gemini 3.0 was used to generate the scripts for data extraction and preprocessing, as well as the majority of code in the supervised model training notebooks. All code was manually reviewed and determined to be suitable for our tasks.