1. **Step 1: Chosen NLP technique and rationale.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tool/Model** | **Reason for Trying** | **Outcome** | **Decision** |
| **SpaCy** | **Lightweight, efficient NLP library with features like NER and dependency parsing.** | **- Struggled with specific queries (e.g., object types, features).** | **Not suitable; lacked generality and customization.** |
| **Hugging Face Transformers  (Flan-T5-small, Flan-T5-large)** | **Open Source. Great performance** | **- Flan-T5-small: Inconsistent, generic outputs. - Flan-T5-large: Performed ok after prompt refinement;** | **Chose Flan-T5-large for its flexibility and strong performance. Identified Llama-7-b as best, but didn’t try due to storage, compute constraints** |
| **OpenAI GPT** | **Closed Source. Best performance.** | **- High-quality results. - Expensive for prototyping.** | **Did not use due to cost; should prioritize for production.** |

1. **Step 2: Building 3D Mesh**

**I searched for multiple papers that could help me create a 3D mesh (useful for engineering , and not just stylistic). A list of papers is in Literature Review.**

**The final comparison, came between Point-E and CLIP-Forge. Used Point-E for quick prototyping.**

|  |  |  |
| --- | --- | --- |
| **Criteria** | **Point-E** | **CLIP-Forge** |
| **Ease of Use** | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ |
| **Output Quality** | ⭐⭐⭐ | ⭐⭐⭐⭐ |
| **Resource Requirements** | ⭐⭐⭐⭐⭐ | ⭐⭐⭐ |
| **Suitability for Refinement** | ⭐⭐⭐ | ⭐⭐⭐⭐ |
| **Setup Complexity** | ⭐⭐⭐⭐⭐ | ⭐⭐ |
| **Overall Applicability** | Great for quick prototypes. | Better for precise, detailed meshes. |

Step 3: Optimizing based on User Feedback

The first decision was to choose between Voxels vs SDFs vs Meshes vs Parametric Surfaces (NURBS) VS CAD. **Chose voxels.**

* **Voxels**: Great for volumetric representation but memory-intensive.
* **SDFs**: Compact and smooth but computationally demanding.
* **Meshes**: Most versatile for general-purpose 3D work; limited for smooth, high-precision surfaces.
* **NURBS/Parametric Surfaces**: Precise and compact, ideal for smooth curves; harder for general shapes.
* **CAD**: Best for engineering and manufacturing due to its precision and integration with real-world constraints.

This step is unfinished. The idea was to design basic ‘code blocks’ which can do specific things, such as, scale along a direction (to change mass), create holes, etc. Then maybe an NLP block could identify key user requests and parameters, and those can be used as inputs to these functions.