

Project Title

AI-Generated Content
Detection Algorithm Using
DSA



Team Information

Team Name:
Sapphire

Team Members:

Atharv Gangwar (Team Lead) –
230112386 | atharvgangwar8@gmail.com

Dhruv Negi – 23011451 |
negi67291@gmail.com

Aditya Awasthi – 23011858 |
adityaawasthi069@gmail.com

Nivedan Belwal – 230111118 |
nivedanbelwal627@gmail.com

Problem Statement

- With the rise of powerful generative AI models like **ChatGPT, Bard, and Claude**, it's becoming increasingly difficult to **differentiate between human-written content and AI-generated content**. This creates significant challenges in:
 - Academics** (plagiarism, AI-written assignments)
 - Journalism** (credibility and originality)
 - Online content moderation**
- Most existing solutions rely on AI models, which are **resource-heavy, dependent on cloud inference, and sometimes inaccurate**.



What's Unique About This Project?



No heavy ML model required



Fast, lightweight, and interpretable



Offline-capable algorithm using pure DSA concepts



Better performance for hybrid content (AI-assisted human writing)



Instead of deep learning, we use smart algorithmic techniques with classic Data Structures & Algorithms to spot AI patterns.

Step-by-Step Implementation

- **Step 1: Preprocessing the Text**

- Remove stopwords
- Convert to lowercase
- Tokenize text

```
mirror_mod = modifier_ob.  
set mirror object to mirror.  
mirror_mod.mirror_object =  
operation == "MIRROR_X":  
mirror_mod.use_x = True  
mirror_mod.use_y = False  
mirror_mod.use_z = False  
operation == "MIRROR_Y":  
mirror_mod.use_x = False  
mirror_mod.use_y = True  
mirror_mod.use_z = False  
operation == "MIRROR_Z":  
mirror_mod.use_x = False  
mirror_mod.use_y = False  
mirror_mod.use_z = True  
  
selection at the end -add  
mirror_ob.select= 1  
modifier_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier_ob.  
mirror_ob.select = 0  
= bpy.context.selected_object  
data.objects[one.name].select  
  
print("please select exactly  
  
-- OPERATOR CLASSES ----  
  
types.Operator):  
X mirror to the selected  
object.mirror_mirror_x"  
mirror X"  
  
context):  
context.active_object is not
```

Step 2: Feature Extraction Using DSA

- **Hashing**
 - Word frequency map
 - Compare with AI-generated patterns
- **Trie (Prefix Tree)**
 - Store AI-specific phrases
 - Check how many are found in the text
- **Graph Algorithms (DFS/BFS)**
 - Construct semantic graph
 - Analyze natural vs. mechanical structure
- **Dynamic Programming (Levenshtein Distance)**
 - Compare with known AI-generated samples
 - Measure text similarity
- **Heap (Priority Queue)**
 - Extract top N frequent words
 - Match with typical AI word distribution

Step-by-Step Implementation

Step-by-Step Implementation

- **Step 3: Classification & Decision**
- Assign weighted scores to extracted features
- Use a threshold to classify:
**AI-Generated or
Human-Written**


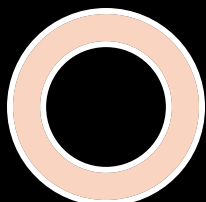

System Architecture (Simplified Flow)

.....
.....
.....
.....


.....
.....
.....
.....



.....
.....
.....
.....



Project Deliverables

- Functional **AI detection web system**
 - **Web API** for text classification
 - **Frontend interface** for users
 - **Project report & documentation**
- 

Future Enhancements



Integration with **TF-IDF, BERT** for deeper NLP



Expand database of known AI phrases and datasets



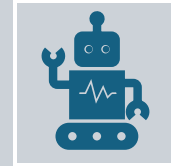
Blockchain proof for human-authored content



Conclusion



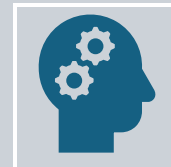
In a world increasingly influenced by generative AI, **distinguishing between human-written and AI-generated content has become more important than ever**—especially in areas like education, journalism, and digital integrity.



Our project, **AI-Generated Content Detection Using DSA**, provides a **lightweight, fast, and interpretable solution** by combining the power of classical **Data Structures & Algorithms** with modern software technologies. Unlike existing AI-heavy detectors, our system is designed to work **efficiently without needing massive datasets or training**.



By leveraging **hashing, trie, graph theory, dynamic programming, and heaps**, we've created a system that doesn't just detect AI content—it explains **how** and **why** it detects it. This transparency and efficiency are what make our solution stand out.



Our approach proves that smart algorithm design can solve modern AI-era problems, offering accuracy and simplicity without relying on complex machine learning models.



Thank You

We're ready to answer any questions.