

SMALL LANGUAGE MODEL CASE STUDY

Humanizer SLM

Constraint-first text rewriting that improves naturalness while preserving names, numbers and critical facts.

Modern content teams, SaaS platforms, and digital agencies are generating an increasing volume of written output with the aid of large language models. While AI accelerates content production, it introduces a recognizable pattern in prose stilted phrasing, templated structure, and a mechanical rhythm that sophisticated readers, search algorithms, and platform detectors can identify instantly.

Project classification: This prototype explores controlled text rewriting that improves tone and readability while preserving protected facts and entities.



INDUSTRY Content Technology & AI Writing	CLIENT TYPE AI model product	PROJECT STAGE Prototype
EVIDENCE Pilot-observed	DELIVERY SCOPE Fine-tuned rewriting model, fact locking and version comparison	

More natural text with critical facts preserved The prototype rewrites text for more natural tone while preserving protected names, numbers and facts.	Less manual proofreading and correction Automated comparison reduces the amount of manual proofreading needed to identify unintended changes.	Configurable tone and domain behaviour Tone and domain controls support different communication styles and use cases.
--	---	---

Who needed the solution

Modern content teams, SaaS platforms, and digital agencies are generating an increasing volume of written output with the aid of large language models. While AI accelerates content production, it introduces a recognizable pattern in prose stilted phrasing, templated structure, and a mechanical rhythm that sophisticated readers, search algorithms, and platform detectors can identify instantly. The client required a reliable and repeatable solution to transform AI-generated or robotic text into natural, human-quality prose without compromising factual accuracy or domain integrity.

What needed to change

Rewriting must be sufficiently adaptive to sound human, yet strictly constrained to preserve factual accuracy. Balancing these opposing requirements forms the core engineering challenge addressed by Humanizer SLM.

AI-generated text often sounded repetitive or robotic

AI-generated text is increasingly detectable by humans, search engines, and content screening tools, reducing trust and engagement.

Generic rewriting tools changed names, numbers or URLs

Generic rewriting tools introduce errors by altering names, numbers, URLs, and key facts, making outputs unreliable for professional use.

Prompting alone did not reliably preserve critical facts

Rewriting must be sufficiently adaptive to sound human, yet strictly constrained to preserve factual accuracy.

Teams needed configurable tone and domain behaviour

Content teams lacked tone control across different communication styles such as professional, friendly, persuasive, and concise.

Editors needed transparency and rollback

No audit trail existed to track changes, review differences, or revert to previous versions.

03 / WORKFLOW TRANSFORMATION

Before and after

Previous workflow

- AI-generated text often sounded repetitive or robotic
- Generic rewriting tools changed names, numbers or URLs
- Prompting alone did not reliably preserve critical facts
- Teams needed configurable tone and domain behaviour
- Editors needed transparency and rollback

Structured workflow

- Fine-tune a focused rewriting model
- Lock critical entities
- Configure tone and domain
- Validate and compare versions
- Review structured outputs

04 / SOLUTION

How we approached it

The solution is built on a fine-tuned Google Flan-T5 model a lightweight, instruction-following sequence-to-sequence transformer selected for its efficiency and precision in constrained text transformation tasks. Model Selection: Google Flan-T5 Flan-T5 was selected for the following reasons: 1.

01

Fine-tune a focused rewriting model

Adapt Flan-T5 to a constrained text-transformation task.

02

Lock critical entities

Protect names, numbers, URLs and domain-sensitive phrases during generation.

03

Configure tone and domain

Control style through reusable templates rather than separate models.

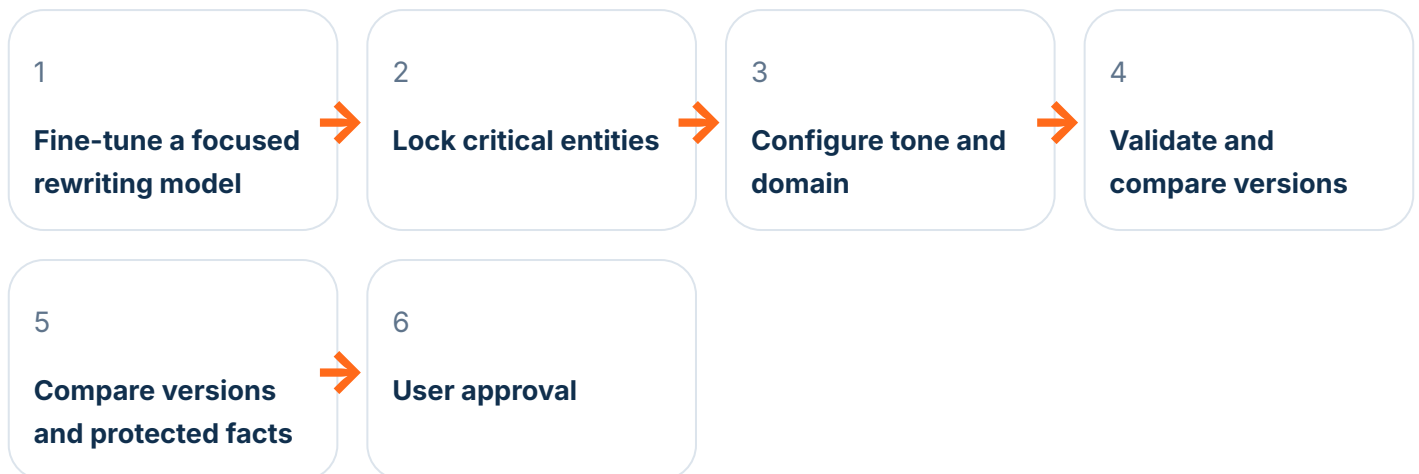
04

Validate and compare versions

Check factual integrity and expose differences, history and rollback options.

05 / EXAMPLE WORKFLOW

From input to reviewable output



06 / DELIVERY SCOPE

What the delivery covered

Rewriting workflow

- Input text and protected-entity capture
- Tone and domain configuration
- Versioned rewrite generation
- Diff and factual-preservation review

Model and guardrails

- Flan-T5 fine-tuning with LoRA
- Fact and entity locking
- Constraint-based output validation
- Quality and mutation testing

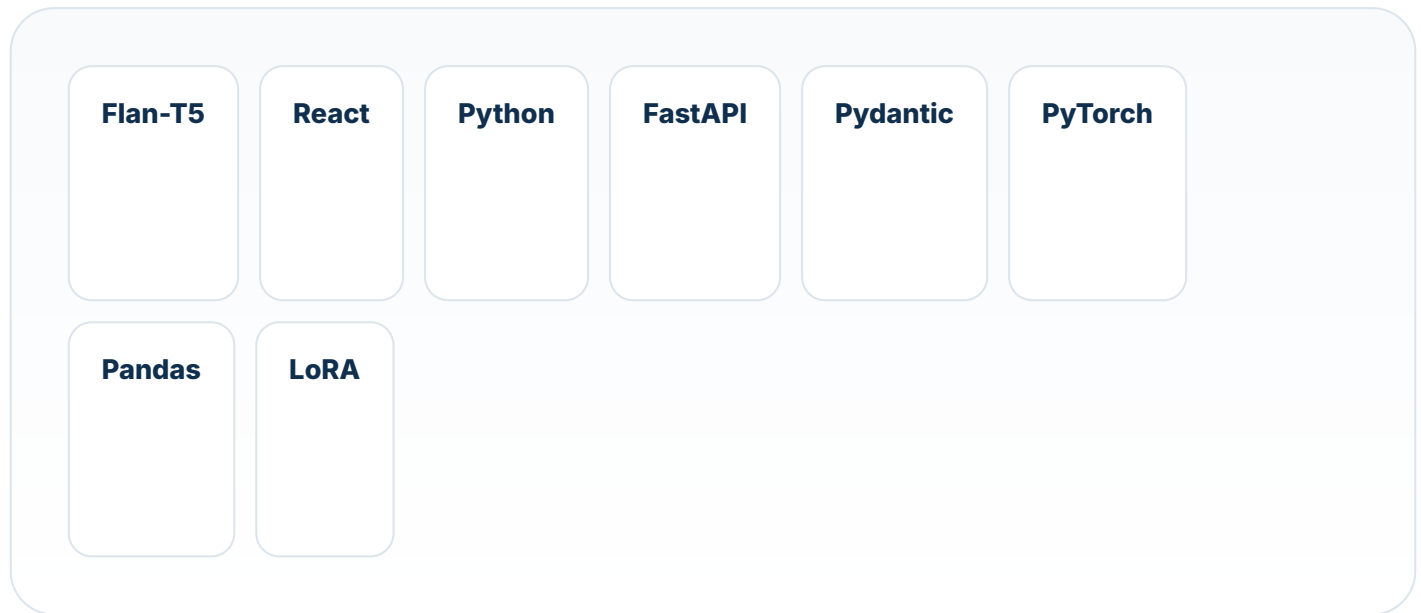
Application engineering

- React interface
- FastAPI service layer
- Pydantic schemas
- Python processing and comparison pipeline

Scope boundary

The prototype is positioned around clarity, tone and factual preservation, not evasion of integrity controls. No zero-mutation claim is made without a documented test method.

How the system is organised



Protected-entity locking

Names, numbers and selected domain facts are locked before generation and checked afterward.

Diff and version review

Users can compare versions and inspect changes before accepting the rewrite.

Acceptable-use boundary

The product should be positioned around clarity and tone, not bypassing integrity or detection controls.

Factual-preservation testing

Any zero-mutation claim requires documented test coverage and methodology.

Value observed during validation

- | | | |
|-------|--|----------------|
| 1 | More natural text with critical facts preserved
The prototype rewrites text for more natural tone while preserving protected names, numbers and facts. | Pilot-observed |
| <hr/> | | |
| 2 | Less manual proofreading and correction
Automated comparison reduces the amount of manual proofreading needed to identify unintended changes. | Pilot-observed |
| <hr/> | | |
| 3 | Configurable tone and domain behaviour
Tone and domain controls support different communication styles and use cases. | Pilot-observed |
| <hr/> | | |
| 4 | Transparent changes through diff and version history
Constraint checks reduce factual mutations introduced by unconstrained rewriting. | Pilot-observed |
| <hr/> | | |
| 5 | Lower-cost deployment through a focused small language model
Version history and diff review make editorial changes easier to inspect before approval. | Pilot-observed |
| <hr/> | | |

Technology and component roles

FL

Flan-T5

Fine-tuned text-rewriting model

RE

React

Reviewer or product user interface

PY

Python

AI, data-processing and backend logic

FA

FastAPI

Backend APIs and workflow orchestration

PY

Pydantic

Structured schema and output validation

PY

PyTorch

Model training and inference

PA

Pandas

Tabular data processing and analysis

LO

LoRA

Parameter-efficient model fine-tuning

NU

NumPy

Numerical processing

10 / RELATED WORK

Explore related case studies

AI SAFETY & CONTENT INTEGRITY

AI Text Detector SLM

A fine-tuned small language model for explainable AI-text detection and adversarial robustness.

CONTENT TECHNOLOGY & MARKETING

YourSocial

A structured content platform that moves users from thinking to outlining, writing and controlled refinement.

ENTERPRISE PRODUCTIVITY

MeetSage

AI-powered meeting capture, transcription, analysis, reports and conversational follow-up.

Working on a similar challenge?

Discuss the workflow, data, integrations and evidence required for a practical first release.