AI USAGE DOCUMENTATION: BiographyViz

Christina Maliariti, Lise Feringa, Felipe Bengoa

1. AI Tools Used

Tool	Model	Purpose
Qwen	3.5 Plus	OCR Transcriptions (Original JPG Letters)
Claude (Anthropic)	Claude Sonnet 4.5	Primary development assistant throughout the project
ChatGPT	5.2	Revising grammar and structure
Cursor IDE	GPT-4 / Claude integration	Code implementation via prompt-based editing

2. Development Process Overview

This project was developed through an iterative human-AI collaboration. As a group we provided domain expertise, biographical knowledge, design decisions, and quality control. Claude assisted with code architecture, debugging and documentation.

All final decisions, corrections, and validations were made by us. AI-generated code reviewed, tested, and modified before integration.

3. Prompt Log by Development Phase

Description: Initial discussion on tool architecture, technology stack selection, and core design decisions.

Key prompts included:

- ”We need a tool that visualizes biographical archives with zero setup for academic users. What technology stack would you recommend?”
- ”Why should we use client-side processing instead of a Python backend?”
- ”Compare Transformers.js vs spaCy for our use case”

Decisions made:

- Next.js 14 with TypeScript and Tailwind CSS
- Client-side processing (no backend)
- Transformers.js over spaCy (zero-setup priority)

- IndexedDB for local storage
- Webpack over Turbopack (WebAssembly compatibility)

What was changed: We validated each decision against project constraints (academic audience, zero-setup requirement, GitHub deployment target).

4. Core Visualizations

Prompt category: Feature implementation. Development of Timeline, Network, Map, and Analytics views.

Key prompts included:

- ”Implement a timeline component that shows letters, photos, and trips with filtering by type and date range”. We tried several different visualizations formats, but ended deciding for the horizontal timeline and leaflet maps. For network analysis, we changed it several times to get to a clear result that could show connections, betweenness and temporality.
- ”Create a network visualization using vis.js with 4-level hierarchy: person → letters → entities → sub-entities”

What was changed:

- Network icon design was simplified after reviewing AI-generated output
- Timeline color scheme was adjusted to match biographical categories

5. NER Implementation

Implementing Named Entity Recognition for automated entity extraction from letter content.

Key prompts included:

- ”Implement local NER using Transformers.js BERT-base-NER model”
- ”The NER model fails to load with error: TypeError: Cannot convert undefined or null to object at Object.keys.”
- ”Fix token recombination for subword tokens, Gauguin is being split into G + ##au + ##guin”

Root causes identified through debugging:

- Turbopack incompatibility with WebAssembly: Fixed by forcing Webpack mode
- BERT subword tokenization fragmenting historical names: Fixed with improved token recombination algorithm

- Van Gogh dictionary activating for all datasets: Fixed with conditional dictionary loading

6. What the AI Did vs. What the Group Did

Task	AI Contribution	Group Contribution
Technology stack	Proposed options with trade-offs	Made final decisions based on project constraints
Code implementation	Generated code via prompts	Tested, debugged, validated, modified
NER debugging	Diagnosed errors from logs	Provided console logs, identified visual errors
Biographical content	Drafted text structures	Verified all facts, corrected errors
Design decisions	Suggested UI patterns	Approved or rejected based on usability testing
Data validation	N/A	Manually verified entity extraction quality
Analysis and Academic Writing	Revised structure and grammar	Wrote the report with critical reflections, analysis and methodology

7. Critical Reflection

Where AI assistance was most valuable

- Boilerplate code generation (reducing development time significantly)
- Debugging complex issues (NER token recombination, WebAssembly compatibility)
- Explaining technical trade-offs clearly

Where AI assistance required significant correction

- **Biographical accuracy**: AI had no knowledge of Luis Mitrovic and generated plausible but incorrect biographical details that required researcher correction
- **Visual quality**: Generated UI code often required aesthetic adjustments
- **Domain specificity:** NER limitations for historical corpora required group-curated domain dictionaries
- **Context retention:** Over long conversations, AI occasionally repeated suggestions already tried or lost track of previous decisions