ChatGPT

# Quality Evaluation and Workflow Analysis for AI-Generated CV Content

## 1. Existing CV contents and AI enhancement pipeline

### 1.1 Data sources and workflow

The interactive CV site (`adrianwedd.github.io/cv`) uses a **JavaScript application** (`assets/script.js`) that defines **data endpoints** for the base CV, activity summary, AI-generated enhancements and GitHub API. The configuration specifies that data are fetched from local JSON files (`data/base-cv.json`, `data/ai-enhancements.json`) and from the GitHub API; a 5-minute cache is used to reduce repeated requests [1]. During initialization, the application concurrently calls `loadCVData`, `loadActivityData`, `loadAIEnhancements` and `loadGitHubStats` [2] and stores the results in memory for later rendering. This means that the quality of content delivered to users depends on the quality of these JSON files and the GitHub statistics.

The **base CV** provides detailed professional information such as employment history, projects, skills, achievements and education. For example, the professional summary describes the candidate as an *"AI Engineer and Software Architect"* who develops autonomous systems and emphasizes experience in machine learning and software architecture [3]. Achievements include delivering **15+ AI-powered autonomous systems**, reducing operational costs by **40 %**, publishing research and mentoring junior developers [4] [5] [6]. The CV lists projects such as **TicketSmith**, **Agentic Research Engine**, **TEL3SIS**, **Agentic Index**, **VERITAS** and **ModelAtlas** with associated technologies and metrics [7] [8]. Skills cover programming languages (Python, JavaScript/TypeScript, Go, Rust), AI disciplines and DevOps tools [9], while achievements and education highlight recognition and self-directed learning [10].

The **AI-enhancements file** contains the output of an AI editing pass. It stores the original and enhanced professional summary. The enhanced version reframes the summary with **measurable impact statements** (e.g., delivering *"15+ autonomous systems"* and increasing efficiency by **40 %**), lists specific languages (Python, TypeScript) and emphasises reliability and human-AI collaboration [11]. Importantly, it notes that *"the numbers provided are placeholders that should be adjusted to match actual achievements"* [12] —this highlights that quality depends on replacing placeholders with verified data. The file also contains a skills analysis recommending additional AI/ML and cloud technologies and an action plan for improving GitHub activity and expertise [13]. Token analytics show that the enhancement process consumed 2,830 tokens across four requests [14].

### 1.2 Current workflow quality

1. **Data completeness** – The base CV provides extensive experience and projects. However, the `activity-summary.json` file referenced in the script is missing (404). The site therefore cannot display recent GitHub activity or languages, leaving the "Commits (30 days)" and other live stats blank. Such gaps undermine factual integrity.

2. **Accuracy and authenticity** – The AI-enhanced summary uses placeholders for key metrics and recommends verifying them [12]. Without cross-checking actual GitHub contributions, there is

risk of exaggeration. The base CV lists achievements like *“95 % reliability”* and *“15+ projects”* [15] but the repository metadata available through the API connector reveals that the user’s repository `adrianwedd/adrianwedd` is publicly accessible and the default branch is *main* [16]; it does not provide commit counts. A robust workflow must therefore integrate GitHub metrics (e.g., commit count, stars, languages) to substantiate claims.

3. **Consistency and narrative flow** – The base CV maintains chronological order across positions and projects. However, the AI-enhanced summary shortens the original narrative and emphasises results; ensuring that this summary aligns with the detailed experience sections is essential to maintain coherence.

4. **Workflow transparency** – The script logs token usage and AI budget in `ai-enhancements.json`, indicating an attempt at governance. Still, there is no description of how prompts were constructed, what model was used or how human reviewers intervened. Documenting these steps would enable traceability and improvement.

## 2. Quality dimensions and scoring criteria

Many modern evaluation frameworks emphasise a combination of **human-centric qualities** (fluency, coherence, relevance, factual consistency and fairness) and **reference-based metrics** [17]. Leveraging these principles and the strategic objectives provided, the following multi-dimensional scoring matrix can be used for CV content:

| Dimension | Indicators / Metrics | Example methods |
|---|---|---|
| **Contextual intelligence & semantic coherence** | • **Professional persona alignment** – does the summary reflect the individual’s actual experience (roles, skills) from the base CV? | |
| • **Industry vernacular** – use of correct technical terms. | | |
| • **Narrative arc** – chronological coherence (e.g., progression from Full-Stack Developer to AI Engineer). | | |
| • **Audience adaptation** – tone and jargon appropriate for recruiters vs. technical peers. | • Compare summaries to structured CV data using semantic similarity metrics (e.g., cosine similarity between base CV sections and summary). | |
| • Check for missing or repeated information. | | |
| • Use heuristics to detect whether domain-specific terms are used correctly. | | |
| **Linguistic excellence & style** | • **Fluency and grammar** – assess readability, grammar and sentence structure. | |

| Dimension | Indicators / Metrics | Example methods |
|---|---|---|
| • **Tone consistency** – balanced professional voice vs. marketing hype. | | |
| • **Readability index** – Flesch–Kincaid or similar. | | |
| • **Persuasive impact** – presence of action verbs, quantifiable results. | • Automated readability tests and grammar checkers. | |
| • Prompt-based evaluators or LLM-as-judge models to rate persuasive quality. | | |
| • Comparison with best-in-class professional profiles for tone. | | |
| **Factual accuracy & professional integrity** | • **Technical claim verification** – cross-reference skills and technologies with public repositories and contributions. | |
| • **Achievement authenticity** – validate metrics like “40 % cost reduction” via client testimonials or GitHub statistics. | | |
| • **Skill representation** – detect skills not backed by experience. | | |
| • **Ethical representation** – avoid inflated claims or misrepresentation. | • Use GitHub APIs to extract commit counts, languages, stars and activity for relevant repositories; compare against claims. | |
| • Maintain a whitelist of verifiable metrics; flag placeholders for review. | | |
| • Incorporate human reviewer checks for sensitive claims. | | |
| **Strategic and market intelligence** | • **Market relevance** – alignment with current AI/ML trends (LLMs, MLOps, edge AI). | |
| • **Competitive differentiation** – highlight unique achievements and projects. | | |
| • **Skill gaps and recommendations** – identify missing emerging technologies. | | |
| • **ROI of enhancement** – estimate how much the AI enhancements improved the profile relative to token cost. | • Compare skills and projects against trending technologies using industry reports. | |
| • Evaluate uniqueness of contributions relative to peer profiles. | | |

| Dimension | Indicators / Metrics | Example methods |
|---|---|---|
| • Track token usage vs. improvement (metrics such as word-count change, improvement indicators in `ai-enhancements.json` [18] ). | | |

### Scoring methodology

1. **Assign weights** to each dimension based on stakeholder priorities (e.g., factual accuracy could be weighted higher than stylistic polish).
2. **Rate each indicator on a 0–5 scale** using automated tools and human review. For example, fluency can be rated by grammar checkers, while persona alignment may require a reviewer to ensure that the summary reflects actual experience.
3. **Compute a composite score** (weighted average) and set **quality gates**. Content falling below threshold triggers a revision cycle.

# 3. Proposed content review workflow

## 3.1 Automated analysis pipeline

1. **Data ingestion and normalization** – Load base CV data, AI enhancements and GitHub statistics from their respective endpoints [19] ; transform into a unified schema.
2. **Contextual cross-checking** – For each generated section, compute semantic similarity to the original CV (e.g., vector embeddings) to detect missing or contradictory information. Flag any section where similarity drops below a threshold.
3. **Language quality evaluation** – Run automated grammar checks, readability tests and style analysis. Evaluate tone consistency against a corpus of high-quality CVs.
4. **Factual verification** –
5. Use GitHub API connectors to retrieve metadata for relevant repositories (e.g., `adrianwedd/adrianwedd` and project repositories) and extract commit counts, languages and update dates [16] .
6. Cross-check technologies listed in the CV (Python, TensorFlow, etc.) against languages used in repositories; flag discrepancies.
7. Identify numeric claims (e.g., number of projects, efficiency gains) and label them as "verified" or "requires evidence." Where `ai-enhancements.json` contains placeholder numbers [12] , require the user to provide actual data.
8. **Strategic analysis** – Compare the CV's skills and projects to market trends. The AI enhancement already suggests focusing on LLMs, MLOps, edge AI and AI ethics [20] ; incorporate these as part of an automated suggestions module.
9. **Composite scoring and recommendations** – Use the scoring matrix to compute scores. Provide targeted recommendations for sections falling below thresholds (e.g., "increase GitHub activity to support AI automation claims"). Recommendations should include actionable steps like contributing to open source, adding quantifiable results or updating skills.

### 3.2 Human-in-the-loop review

While automatic metrics are essential, **human review remains critical**. Microsoft's evaluation playbook notes that automatic metrics should complement but not replace human judgment [21]. Human reviewers should:

- **Validate authenticity** of achievements by checking references (e.g., client testimonials, publications).
- **Ensure narrative coherence** across sections and check for overuse of AI-generated phrasing.
- **Assess tone and personality** to confirm that the enhanced summary maintains the individual's voice.

### 3.3 Workflow improvements for the CV system

1. **Fix missing data sources** – Add or update `activity-summary.json` to reflect actual GitHub activity (commits in the last 30 days, languages used). Without this file, the live stats remain empty.
2. **Audit AI enhancement prompts** – Record prompts, models used and hyper-parameters to enable reproducibility and debugging. Include token usage and cost per enhancement (already partially logged in `ai-enhancements.json` [14]).
3. **Placeholder replacement** – Design the AI workflow to detect placeholders (e.g., numbers flagged as placeholders [12]) and automatically request user input or fetch metrics from GitHub to replace them. This reduces the risk of misrepresentation.
4. **Versioning and change tracking** – Store different versions of CV sections and maintain diff logs so reviewers can trace how content evolved over time. This can inform continuous improvement.
5. **Feedback loop** – Collect recruiter and peer feedback on the generated CV and feed ratings back into the scoring model to refine weights and thresholds.

## 4. Summary of key recommendations

- **Adopt a multi-dimensional scoring matrix** incorporating contextual alignment, linguistic quality, factual accuracy and strategic relevance. Use automated metrics for fluency, coherence and similarity but retain human review to ensure fairness and authenticity [17].

- **Strengthen data integrity** by ensuring all referenced data files exist (e.g., activity summary) and by verifying numeric claims through GitHub APIs and external evidence. Treat placeholder numbers in AI enhancements as red flags that require confirmation [12].

- **Enhance workflow transparency** by logging prompts, model versions and token usage; implement version control for CV content and track enhancements over time. Provide a structured process for replacing placeholders with verified data.

- **Introduce human-in-the-loop checkpoints** to assess tone, narrative coherence and ethical representation; automatic metrics alone cannot ensure fairness and contextual appropriateness [21].

- **Align strategic suggestions with market trends** by incorporating insights from industry reports (e.g., focus on LLMs, MLOps, edge AI, AI ethics [20]) and highlighting unique projects and achievements to differentiate the profile.

Implementing these improvements will elevate both the **quality of the generated CV content** and the **robustness of the workflow** that produces it, ensuring that enhancements preserve authenticity while maximizing professional impact.

---

[1] [2] [19] adrianwedd.github.io
https://adrianwedd.github.io/cv/assets/script.js

[3] Adrian Wedd - AI Engineer & Software Architect
https://adrianwedd.github.io/cv/

[4] [5] [6] [7] [8] [9] [10] [15] adrianwedd.github.io
https://adrianwedd.github.io/cv/data/base-cv.json

[11] [12] [13] [14] [18] [20] adrianwedd.github.io
https://adrianwedd.github.io/cv/data/ai-enhancements.json

[16] adrianwedd
https://api.github.com/users/adrianwedd

[17] [21] Evaluation metrics | Microsoft Learn
https://learn.microsoft.com/en-us/ai/playbook/technology-guidance/generative-ai/working-with-llms/evaluation/list-of-eval-metrics