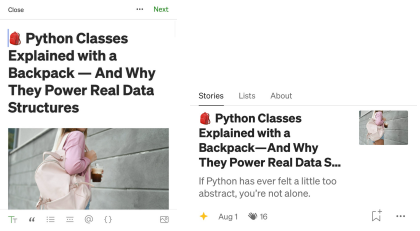
Redesign Project

Gyeongbin Ryoo

1. A clear description of the problem statement, i.e. the usability flaw, supported by examples, diagrams, photos, or screenshots of the original interface.



The Medium mobile app enables reading and writing articles, but one recurring usability flaw is the **mismatch between how subtitles appear in different views**. As shown in the attached figures, the subtitle "🎒 Python Classes Explained with a Backpack—And Why They Power Real Data S…" demonstrates this inconsistency:

•**Preview Mode (Figure 1)**: Shows complete subtitle text

•**Published View (Figure 2)**: Truncates subtitle to "…Data S…" with ellipses

This inconsistency forces writers to toggle repeatedly between modes to confirm formatting, resulting in **wasted time, formatting uncertainty, and reduced trust in the WYSIWYG editor**.

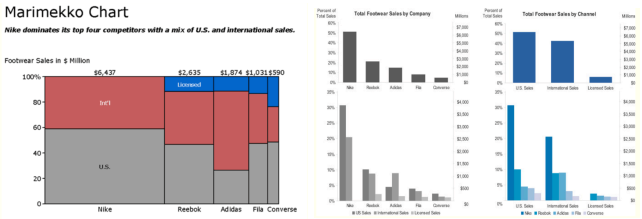
Heuristic Violations: This flaw violates core Nielsen heuristics: users cannot see how subtitles will actually render until switching modes, breaking WYSIWYG expectations (**Visibility of System Status**), Preview and published views apply different truncation rules, violating internal consistency and industry standards (**Consistency & Standards**), Medium provides no early warning about truncation—users discover errors only after publishing (**Error Prevention**), and writers must remember potential truncation rules instead of recognizing visual cues like character counters or overflow warnings (**Recognition over Recall**).

External Validation: Similar frustrations appear across platforms. **Reddit users** report subtitle positioning issues in VLC ("subs stick to bottom of video, not screen") and **Plex** ("subtitles displayed in the middle"), reinforcing that inconsistent text rendering creates widespread user frustration. Such complaints underscore the need for a unified, principle-based solution that addresses all four heuristic violations simultaneously.

2. Your evidence-based justification for why the flaw is a problem.

2.1 Visualization Critiques as Analogies

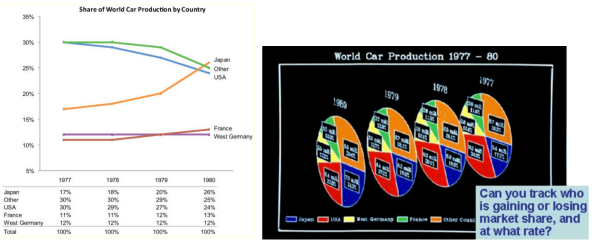
Marimekko Chart Redesign (Few, 2006):



(**Image reference**: https://www.perceptualedge.com/example13.php)

The Marimekko chart forced viewers to compare both **width and height** meaning you had to **compare areas of rectangles**—humans aren't good at judging areas—while the redesign used **bar charts** to enable direct, accurate length comparisons. Similarly, Medium's subtitle preview problem makes writers mentally juggle **two inconsistent formats** (editor vs. preview), imposing unnecessary cognitive effort that could be avoided with inline cues.

World Car Production Pie Charts (Spielman, 1995):



(**Image Reference**: https://www.perceptualedge.com/example7.php)

The tilted 3D pie chart hid which countries were growing or shrinking because the angle distorted perception and concealed the real story. The redesign fixed this by using a **line chart with a table**, making both **trends and exact values easy to see**. In the same way, Medium's editor presents **a polished but misleading view**—users only discover subtitle issues like truncation, emoji shifts, or line breaks after switching to preview. Like the improved chart, a better editor would expose these "truths" directly with transparent cues such as counters or inline warnings.

2.2 Cognitive & Rhetorical Analysis

Mode switching increases cognitive overhead when writers must remember truncation or emoji shifts rather than recognizing them directly, placing additional demands on **working memory**. Research emphasizes the importance of respecting a viewer's limited working memory by avoiding designs, such as legends or animations, that can confuse rather than clarify, confirming that extraneous cognitive demands should be avoided (Franconeri et al., 2021, pp. 112, 136–138). Additionally, the medium conceals truncation risk instead of making it transparent through pixel budgets or inline preview markers, violating principles of honest information design. Effective visualization design should make **uncertainty** visible rather than hidden to support users' cognitive processing (Franconeri et al., 2021).

3. A concise description and mockups of your proposed redesign.

To address the identified usability flaws, I propose an **AI-driven Subtitle Optimization System** integrated with a live preview panel that eliminates editor/preview inconsistencies while providing intelligent writing assistance.

A. Real-Time Live Preview: A real-time live preview offers a **side-by-side display** as users type, ensuring subtitle formatting matches exactly how it will appear in all contexts.

B. Subtitle "Budget Meter" : A **pixel budget counter** shows the remaining space with precise measurements, while **color-coded warnings**—such as "Feed fits" in green and "18px overflow → will truncate" in red—clearly indicate status. These **transparent constraints** align with visualization best practices by exposing baselines and presenting honest, interpretable scales.

C. AI Suggest-and-Explain System (Core Technical Implementation): The enhanced editor integrates three **core AI features**: real-time **truncation warnings** that alert users with precise character counts (e.g., "⚠ Will truncate at 2 lines on 375px"), **smart suggestion chips** that provide context-aware recommendations such as "Tighten wording" (clearly marked as "Strongly Recommended"), and **interactive tooltips** that offer contextual

explanations—including options like “This will shorten your title with a concise version”—with straightforward Apply or Dismiss actions.

3.1 Technical Architecture

Mobile-First Text Fitting
• `fit_delta_px` — wraps text to **MOBILE_LINE_WIDTH_PX × MOBILE_MAX_LINES**, returns overflow (>0) or spare margin (<0); if it fits, computes `fit_norm = min(1.0, margin/24.0)`.

Content Analysis
• `top_keywords` / `keyword_coverage` — extracts salient words and measures coverage vs. the current subtitle.
• `tfidf_similarity` — TF-IDF + cosine similarity between candidate and article.
• `extract_article_tags` — TF-IDF n-grams used by the add-detail strategy.

Three-Strategy Candidate Generation
• `gen_tighten_variants` — remove extras (punctuation/parentheticals/tail words).
• `gen_replace_concise_variants` — shorter substitutes (caps, stop-word drop).
• `gen_add_detail_variants` — append concise, context tags (uses `extract_article_tags`).

Scoring & Ranking (Multi-Factor)
• `score_candidate` — composite score = **0.5·fit_norm + 0.3·coverage + 0.2·similarity**.
• `rank_candidates` — sorts by **score**, then **margin_px**, then **shorter text**.

Badges & Final Recommendation
• `is_strong` — threshold gate (fits & margin≥20px & coverage≥0.66 & sim≥0.50).
• `suggest_tighten` / `suggest_replace_concise` / `suggest_add_detail` — run a strategy, rank, and annotate each top result with `"strength"` and `"why"`.
• `decide_action_badges` — calls all three `suggest_*`, builds `badges = {strategy: top_is_strong}`, and picks a global best by **strength → score → margin**.

Why This Technical Approach Works:
1.Addresses Heuristic Violations:
This technical approach directly resolves key heuristic violations: real-time pixel calculations provide immediate **visibility** of fitting status, using the same rendering pipeline as the published view ensures **consistency**, proactive warnings **prevent truncation errors**, and clear visual cues **eliminate the need for users to re**call complex formatting rules.
2.Cognitive Load Reduction:
This approach reduces cognitive load by using **multi-criteria optimization** to address layout, semantic, and keyword constraints simultaneously, providing **explainable recommendations** that clearly show their reasoning (e.g., “Keeps keywords: Python, classes”), and enabling **one-click acceptance** to eliminate the need for trial-and-error editing cycles.
3.User Control and Freedom (Heuristic #3)
Writers have full control to **accept, reject, or edit AI suggestions**; the system augments their process **without forcing automation**, and delivers **crisp inline messages**—such as “Will truncate in Feed”—in place of hidden tooltips

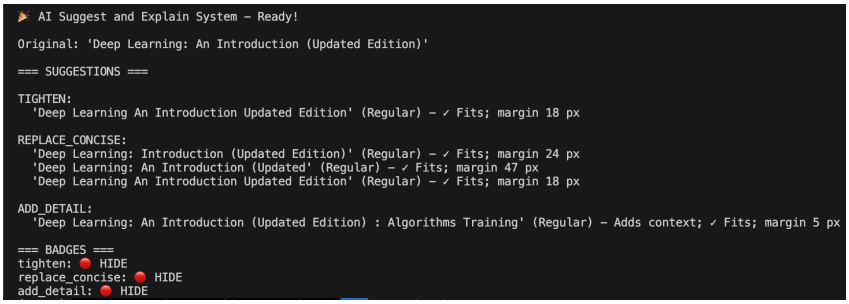
4. Working Prototype & Technical Validation

Functional Python Implementation

The functional Python implementation uses a **technology stack** of scikit-learn for TF-IDF analysis, PIL/Pillow for text measurement, and regex for tokenization, is **fully cross-platform** with support for macOS and Linux and delivers **performance** of real-time candidate generation and scoring for interactive use.

Prototype Testing Results
Input: "Python Classes Explained with a Backpack — And Why They Power Real Data Structures"
Generated Suggestions:
•**TIGHTEN:** "Python Classes Explained with a Backpack" (Fits; margin 18 px)
•**REPLACE_CONCISE:** "Python Classes: Backpack Analogy" (Fits; margin 45 px)
•**ADD_DETAIL:** "Python Classes Explained with a Backpack — Beginner Guide" (Fits; margin 8 px)

System Output:



Working Python Implementation: The system delivers a **complete subtitle optimization** algorithm with multi-factor scoring, integrates **cross-platform text measurement** and **TF-IDF similarity analysis**, and supports real-time candidate generation along with an intelligent badge system.
View Source Code: https://gist.github.com/growingpenguin/a914ebae4e8e649e1aa8a248367a110e

UI Integration Points: The **Figma prototype** provides interactive mockups that showcase the integration of **AI-powered subtitle suggestion chips**, enabling users to test all **three optimization strategies**—Tighten, Concise, and Detail—while viewing a live, multi-context preview panel; the live demo is available through the provided **Figma link**, and a companion GitHub repository includes the runnable HTML/CSS/JS demo and assets.
Figma Demo: https://www.figma.com/design/00fEInYY0d3geOVDO5EZn2/Medium?node-id=0-1&t=6dS9KJIX3XaZiv9c-1

GitHub Repo: https://github.com/growingpenguin/hci_redesign_project.git

5. Conclusion

By combining **Nielsen's heuristics analysis** with a working AI implementation, this redesign demonstrates how targeted technical interventions can meaningfully improve Medium's subtitle editing experience, **eliminating unnecessary mode-switching** with live previews, **preventing truncation errors** through accurate prediction, **reducing cognitive load** via multi-factor scoring, and **maintaining semantic quality** with TF-IDF similarity. The result is a minimal, plausible, and impactful system that resolves key heuristic violations, proactively prevents errors, saves writers time through intelligent automation, and fosters greater trust in the platform with transparent, explainable recommendations.

6. References

1.Nielsen Norman Group — 10 Usability Heuristics for User Interface Design
2.Perceptual Edge (Stephen Few) — Visualization critique examples
3.Franconeri, Padilla, Hullman, et al. (2021) — The Science of Visual Data Communication: What Works, Psychological Science in the Public Interest
4.Reddit discussions on UI consistency issues (One UI 7, VLC, Plex subtitle positioning)
5.scikit-learn Documentation — TF-IDF Implementation and Best Practices