

CopyscriptAI Bug Fix & Improvement To-Do List

This document provides a structured list of all bugs and improvements needed for the CopyscriptAI project. Items are organized by category and priority, with current status indicated.

API Integration Issues

OpenRouter API

- ☒ **COMPLETED:** Fix OpenRouter API routing in text processing functions
 - **FIXED:** `text_processor.py` functions (`humanize_text`, `check_grammar`) now properly respect OpenRouter configuration
 - **FIXED:** Added proper import of `make_openrouter_api_call` function
 - **FIXED:** Updated functions to check `context.config.use_openrouter` flag before routing API calls
 - **VERIFIED:** Script 2's `ai_utils.py` and both scripts' `chunking_utils.py` already had proper OpenRouter routing
 - **NOTE:** Test with actual article generation to verify OpenRouter API calls are working properly
- ☒ **COMPLETED:** Fix OpenRouter API routing in FAQ generation functions
 - **FIXED:** `faq_handler.py` function `generate_faq_section()` now properly respects OpenRouter configuration
 - **FIXED:** Added proper import of `make_openrouter_api_call` function
 - **FIXED:** Updated function to check `context.config.use_openrouter` flag before routing API calls
 - **VERIFIED:** `generate_faqs()` function already used proper routing via `gpt_completion()`
 - **NOTE:** Both FAQ generation functions now properly route through OpenRouter when configured
- ☐ **HIGH:** Fix hourly limit issues with OpenRouter API
 - Implement automatic back-off on 429 errors
 - Add retry decorator with exponential backoff and jitter using tenacity library
 - Reference: <https://openrouter.ai/docs/api-reference/limits>
- ☒ **COMPLETED:** Fix misleading log messages in API routing
 - **FIXED:** Updated log messages in `text_processor.py` to correctly indicate "OpenRouter" vs "OpenAI"
 - **VERIFIED:** Other functions already had proper logging

WordPress API Connection

- ☒ **COMPLETED:** Fix WordPress API settings and user management
 - Move WordPress configurations (categories, author, post status) from `config.py` files to `.env` files
 - Set 'uncategorized' (ID: 1) as the default category
 - Set 'draft' as the default post status
 - Implement functionality to display WordPress URL links for successfully posted articles
- ☐ **MEDIUM:** Fix user login for Script1 (currently only allows admin user)

SERP API

- ☐ **LOW:** Investigate SERP API credit consumption
 - Document which functions and activities use SERP API
 - Explain unexpected credit consumption when not using external links and PAA

Content Generation Improvements

Text Processing

- ☒ **COMPLETED:** Implement distinct BLOCKNOTES_COMBINE_PROMPT
 - ☒ Created dedicated prompt for combining block notes chunks
 - ☒ Implemented consistent chunking between script1 and script2
 - ☒ Added proper prompt initialization and configuration
 - ☒ Verified proper usage in combine_chunk_results_with_llm function
 - **NOTE:** New prompt ensures single focused paragraph with most important points only
- ☒ **COMPLETED:** Fix Grammar Tool in Script 1
 - ☒ Resolved error: "RetryError[<Future at 0x2a4e2c8fd50 state=finished raised KeyError>]"
 - ☒ Fixed parameter formatting in grammar check function (changed from `grammar_check=text` to `text=text`)
 - ☒ **COMPLETED:** Add proper sequencing for humanization and language tool when both are enabled
 - ☒ **COMPLETED:** Fix issues when grammar checking and humanization features are both enabled
 - ☒ **COMPLETED:** Ensure grammar checking works correctly after humanization is applied
- ☐ **MEDIUM:** Fix hallucination/repetition issues
 - Address long/repetitive subtitle generation in Script 1
 - Fix repetitive paragraph generation in Script 2 when creating longer articles
- ☒ **COMPLETED:** Improve summary generation
 - ☒ **FIXED:** Added SUMMARY_COMBINE_PROMPT and BLOCKNOTES_COMBINE_PROMPT to script2's prompts.py (previously missing)
 - ☒ **FIXED:** Implemented distinct prompts for summary vs blocknotes combining with different requirements:
 - Summary: Multi-paragraph comprehensive overview
 - Blocknotes: Single focused paragraph (max 150 words)
 - ☒ **FIXED:** Ensured professional, non-conversational tone by updating both prompts
 - ☒ **FIXED:** Removed conversational phrases like "Here's your summary"
 - ☒ **FIXED:** Verified proper implementation in both script1 and script2
 - **NOTE:** Both prompts are now properly defined in prompts.py and correctly used in combine_chunk_results_with_llm function
 - **NOTE:** Previously script2 was missing dedicated prompts for combining summaries and blocknotes

Summary and Keynote Issues

- ☒ **COMPLETED:** Fix broken summary generation in Script 2
 - **FIXED:** Implemented dedicated `generate_article_summary` method in ContentGenerator class
 - **FIXED:** Added proper chunking and combining logic following the same approach as Script One
 - **FIXED:** Ensured multi-paragraph summary (2-3 paragraphs) is generated instead of just 3 lines
 - **FIXED:** Fixed prompt handling to explicitly request comprehensive paragraphs
 - **FIXED:** Added proper error handling for summary generation failures
 - **METHODOLOGY:**
 - Created a dedicated method in ContentGenerator class that mirrors Script One's implementation
 - Implemented proper chunking with `chunk_article_for_processing`
 - Added LLM combination of chunks with `combine_chunk_results_with_llm`
 - Used the same approach for model selection as Script One (supporting OpenRouter)
 - Fixed imports to ensure all required modules are available
 - **NOTE:** Testing is still pending to verify the fix completely resolves the issue
- ☒ **COMPLETED:** Fix missing keynote section in Script 2
 - **FIXED:** Resolved issue where block notes (keynotes) were not appearing in WordPress output
 - **FIXED:** Fixed bug where block notes were being added to `article_dict` instead of `article_components`
 - **FIXED:** Added proper error handling around block notes generation to prevent silent failures
 - **FIXED:** Enhanced `format_article_for_wordpress` function with parameter to control block notes inclusion
 - **FIXED:** Improved how block notes are formatted in WordPress, properly handling multiple paragraphs
 - **METHODOLOGY:**
 - Debugged the block notes generation flow and identified that they were generated but not properly included
 - Created a dedicated article dictionary specifically for block notes generation
 - Fixed the integration between block notes generation and WordPress formatting
 - Added explicit parameter to toggle block notes in WordPress output
 - Enhanced block notes formatting to better handle multiple paragraphs
 - **NOTE:** Testing is still pending to verify the fix completely resolves the issue

PAA (People Also Ask) Functionality

- ☐ **MEDIUM:** Add control for maximum PAA questions
 - Implement option to set maximum number of scraped questions
 - Add ability to set random range (e.g., 3-10)
 - Fix issue where PAA titles are removed when humanization is enabled (Script 2)

Title Management

- ☒ **COMPLETED:** Enable title crafting option
 - **FIXED:** Added `enable_title_crafting` boolean toggle to Config class in both scripts

- **FIXED:** Implemented TITLE_CRAFT_PROMPT in both scripts' prompts.py
- **FIXED:** Added title_craft field to Prompts dataclass
- **FIXED:** Updated prompt initialization in both scripts' main.py
- **NOTE:** Now supports both direct keyword usage and LLM-enhanced titles
- ☒ **COMPLETED:** Fix URL prefix in Script 2
 - **FIXED:** Added use_keyword_for_url and url_duplicate_handling options to Config
 - **FIXED:** Implemented URL generation from long-tail keywords
 - **FIXED:** Added proper handling for duplicate URLs using incremental numbers
 - **VERIFIED:** Successfully tested with WordPress posting functionality
 - **NOTE:** URL generation now supports both keyword-based and full-title options

Image Handling

- ☐ **MEDIUM:** Enhance image handling features
 - Add image alignment options (center, left, right)
 - Implement image compression using Pillow
 - Add option to prevent duplicate images in the same article

Error Handling & CSV Processing

- ☒ **COMPLETED:** Improve CSV parsing in Script 2
 - **FIXED:** Implemented flexible CSV parsing to handle different subtitle counts across articles
 - **FIXED:** Added error handling for varying subtitle quantities
 - **FIXED:** Created unified CSV processor in utils/unified_csv_processor.py
 - **FIXED:** Added detailed per-article subtitle count reporting during initialization
 - **VERIFIED:** Successfully tested with test_flexible_csv.py

Script-Specific Issues

Script 1

- ☐ **MEDIUM:** Fix long subtitle hallucination
- ☐ **LOW:** Document how the script works with context information

Script 2

- ☒ **COMPLETED:** Add randomize_images parameter to Config class
 - **FIXED:** Added missing randomize_images parameter to script2's Config class
 - **FIXED:** Ensured compatibility with image_handler.py functionality
 - **FIXED:** Set default value to False for predictable behavior
- ☐ **HIGH:** Fix language tool functionality
- ☐ **MEDIUM:** Fix token limit handling and provide better error messages
- ☐ **MEDIUM:** Investigate hallucination issues with LLM model + RAG
- ☐ **LOW:** Clarify OpenAI/tiktoken dependency and usage
- ☐ **LOW:** Document relationship between context window size and repetition issues
- ☐ **LOW:** Investigate Windows library compatibility issues

Not Prioritized Issues (For Future Consideration)

- ☐ RAG related issues:
 - Cloudflare URL access error
 - Unused parameter: rag_openrouter_model
 - Number of scraped URLs control
-

Status Legend

- ☒ Completed
- ☐ In progress/Not started

Priority Levels

- **HIGH:** Critical functionality issues, severe bugs
- **MEDIUM:** Important improvements, non-critical bugs
- **LOW:** Minor improvements, documentation, clarifications