

# **OpenClaw**

## **SEO Content Automation**

### Complete System Guide

*Autonomous Keyword Research • Content Creation • Publishing*

**From \$2/day to 1,000+ ranking articles**

# Table of Contents

## OpenClaw SEO Content Automation Guide

**Complete System for Autonomous Keyword Research, Content Creation & Publishing**

### Table of Contents

1. [Introduction](#introduction)
2. [What This System Does](#what-this-system-does)
3. [Prerequisites](#prerequisites)
4. [Quick Start: 5-Minute Setup](#quick-start)
5. [Phase 1: Trend Discovery](#phase-1-trend-discovery)
6. [Phase 2: Keyword Research & Validation](#phase-2-keyword-research)
7. [Phase 3: Content Creation](#phase-3-content-creation)
8. [Phase 4: Publishing Automation](#phase-4-publishing)
9. [Master Orchestrator Setup](#master-orchestrator)
10. [Monitoring & Optimization](#monitoring)
11. [Troubleshooting](#troubleshooting)
12. [Advanced Strategies](#advanced-strategies)

### Introduction

This guide shows you how to build a **fully autonomous SEO content system** using OpenClaw that:

- Discovers trending topics 24/7
- Researches and validates keywords automatically
- Writes SEO-optimized articles
- Publishes content without human intervention
- Costs \$2-5 per day in API calls

**The entire system runs on copy-paste prompts.** No complex configurations, no coding required.

# What This System Does

## The Complete Workflow

```
Every 6 Hours: Discover Trending Topics
    ↓
Daily: Validate Keywords & Check Competition
    ↓
Daily: Create 1-2 Optimized Articles (1500-2000 words)
    ↓
Daily: Publish to WordPress/Ghost/Medium
    ↓
Track Performance & Self-Optimize
```

## Real Results

- **47 trending topics discovered** per week
- **\$0.18 per article** in API costs
- **Articles ranking on page 2** within 48 hours
- **Zero manual work** after initial setup
- **Scales to unlimited niches**

## Prerequisites

### Required Accounts (All Free Tiers Work)

#### 13. **Brave Search API** - <https://brave.com/search/api/>

- Sign up for free tier (2,000 queries/month)
- Get your API key

#### 14. **Reddit Account** - <https://www.reddit.com/prefs/apps>

- Create an app (select "script")
- Note: Client ID and Secret

#### 15. **WordPress Site** (or Ghost/Medium)

- Generate Application Password (Settings → Users → Application Passwords)
- Note: Username and App Password

#### 16. **OpenClaw Setup**

- Have OpenClaw installed and running
- Confirm MCP server access (fetch, filesystem)

### Optional (But Recommended)

- **NewsAPI** - <https://newsapi.org> (100 free requests/day)
- **Google Search Console** - For indexing automation
- **(optional) Airtable/Notion** - For content calendar tracking

# Quick Start: 5-Minute Setup

## Step 1: Create Your Workspace

In your OpenClaw working directory, run this command:

```
mkdir -p  
seo-automation/{trends,keywords,articles/{drafts,published},config,logs}
```

This creates the folder structure for your automation system.

## Step 2: Configure Your API Keys

Create a config file at `seo-automation/config/api_keys.json`:

```
{  
    "brave_search_api_key": "YOUR_BRAVE_API_KEY",  
    "reddit_client_id": "YOUR_REDIT_CLIENT_ID",  
    "reddit_client_secret": "YOUR_REDIT_SECRET",  
    "wordpress_url": "https://yourblog.com",  
    "wordpress_username": "your_username",  
    "wordpress_app_password": "your_app_password",  
    "newsapi_key": "YOUR_NEWSAPI_KEY"  
}
```

## Step 3: Set Your Niches

Create `seo-automation/config/niches.json`:

```
{  
    "primary_niches": [  
        "AI automation",  
        "SaaS tools",  
        "no-code development",  
        "startup growth"  
    ],  
    "subreddits": [  
        "Entrepreneur",  
        "SaaS",  
        "startups",  
        "AI_Agents",  
        "nocode"  
    ],  
    "target_keywords_per_day": 5,  
    "articles_per_day": 2,  
    "min_trend_score": 70  
}
```

## Step 4: Test API Connections

Create your first OpenClaw task with this test prompt:

Test API Connections

YOUR TASK:

```
Read the API keys from /seo-automation/config/api_keys.json

Then test each connection:

1. BRAVE SEARCH TEST:
Use the fetch tool to call:
https://api.search.brave.com/res/v1/web/search?q=test&count=1

Headers needed:
- X-Subscription-Token: YOUR_BRAVE_API_KEY

If successful, you should get search results back.

2. REDDIT TEST:
Get access token:
POST https://www.reddit.com/api/v1/access_token
Auth: Basic base64(client_id:client_secret)
Body: grant_type=client_credentials

Then fetch: https://oauth.reddit.com/r/Entrepreneur/hot?limit=1
Header: Authorization: bearer YOUR_TOKEN

3. WORDPRESS TEST:
GET https://yourblog.com/wp-json/wp/v2/users/me
Auth: Basic base64(username:app_password)

OUTPUT:
Report which connections succeeded or failed.
Save results to /seo-automation/logs/connection_test.json

If any fail, provide troubleshooting steps.
```

Run this task. Once all connections pass, you're ready to start.

## Phase 1: Trend Discovery

This prompt runs every 6 hours to find emerging topics worth writing about.

### The Complete Prompt

```
TREND DISCOVERY AGENT - Run Every 6 Hours

Read configuration from:
- /seo-automation/config/api_keys.json
- /seo-automation/config/niches.json

YOUR MISSION:
Find trending topics in your niches that have high potential for SEO
content.



---


STEP 1: SEARCH BRAVE FOR TRENDING TOPICS


---


```

For each niche in your config, search Brave for recent discussions:

Use the fetch tool to call Brave Search API:

```
URL: https://api.search.brave.com/res/v1/web/search
Method: GET
Headers:
  X-Subscription-Token: YOUR_BRAVE_API_KEY
  Accept: application/json

Parameters for each search:
  q: "[niche] trends 2025" OR "[niche] latest news" OR "new [niche] tools"
  count: 20
  freshness: pd (past day)
  search_lang: en
  result_filter: web
```

Run 3 searches per niche:

1. "[niche] trends" - Find emerging trends
2. "best [niche] tools 2025" - Find tool comparisons
3. "[niche] how to" - Find tutorial opportunities

EXTRACT FROM RESULTS:

- Page title
- Description/snippet
- URL
- Last modified date (if available)

SCORING:

For each result, assign a "freshness score":

- Published in last 24h = 100 points
- Last 3 days = 75 points
- Last 7 days = 50 points
- Older = 25 points

---

## STEP 2: CHECK REDDIT FOR ENGAGEMENT

---

For each subreddit in your config:

1. Get Reddit OAuth token:

```
POST https://www.reddit.com/api/v1/access_token
Headers:
  Authorization: Basic base64(client_id:client_secret)
  Content-Type: application/x-www-form-urlencoded
Body: grant_type=client_credentials
```

2. Fetch top posts from last 24 hours:

```
GET https://oauth.reddit.com/r/SUBREDDIT/top.json?t=day&limit=25
Headers:
  Authorization: bearer YOUR_ACCESS_TOKEN
  User-Agent: TrendBot/1.0
```

```
3. Extract from each post:  
- Title  
- Upvotes (score)  
- Number of comments (num_comments)  
- URL  
- Created UTC timestamp  
- Post body text (selftext)  
  
4. Calculate engagement score:  
engagement_score = (upvotes × 0.4) + (num_comments × 0.6)
```

Why? Comments indicate real discussion and questions = content opportunities

```
5. Extract keywords from high-engagement posts:  
- Look for repeated phrases in title + body  
- Note questions people are asking  
- Identify pain points mentioned
```

---

---

#### STEP 3: CHECK NEWS SOURCES (OPTIONAL)

---

---

If you have NewsAPI configured:

```
GET https://newsapi.org/v2/everything  
Parameters:  
q: "YOUR_NICHE"  
from: 24 hours ago  
sortBy: popularity  
apiKey: YOUR_NEWSAPI_KEY
```

Extract:

- Article title
- Description
- Published date
- Source name
- URL

Recency score: Same as Brave (24h = 100, 3d = 75, etc.)

---

---

#### STEP 4: CROSS-REFERENCE & SCORE TOPICS

---

---

Now combine all your findings:

For each unique topic discovered:

1. Calculate TREND SCORE:
  - Brave freshness score (0-100)
  - Reddit engagement score (normalize to 0-100)
  - News recency score (0-100)
  - Multi-source bonus: +25 points if topic appears in 2+ sources

- ```

FINAL SCORE = Average of all scores + multi-source bonus

2. Extract KEYWORDS from the topic:
- Main topic phrase (2-4 words)
- Related long-tail variations
- Question formats people are asking

3. Flag CONTENT ANGLE:
- Tutorial ("how to...")
- Comparison ("best... vs...")
- List ("10 best...")
- Guide ("complete guide to...")
- News ("what's new in...")

```
- 
- 

#### STEP 5: SAVE RESULTS

---



---

Save to /seo-automation/trends/YYYY-MM-DD\_HH-MM.json:

```
{
  "timestamp": "2025-02-26T14:30:00Z",
  "run_id": "unique-id",
  "niches_searched": ["AI automation", "SaaS tools"],
  "subreddits_checked": ["Entrepreneur", "SaaS"],
  "total_topics_found": 47,
  "topics": [
    {
      "topic": "AI agents for content creation",
      "trend_score": 87,
      "keywords": [
        "AI content agents",
        "automated content creation",
        "AI writing tools 2025"
      ],
      "content_angle": "comparison",
      "sources": [
        {
          "type": "reddit",
          " subreddit": "Entrepreneur",
          "post_title": "Anyone using AI agents for their content?",
          "engagement_score": 450,
          "url": "reddit_url",
          "key_questions": [
            "Which AI agent is best?",
            "How much does it cost?",
            "Can it match human quality?"
          ]
        },
        {
          "type": "brave",
          "title": "Top AI Content Tools Compared 2025",
          "freshness_score": 100,
          "url": "article_url"
        }
      ]
    }
  ]
}
```

```
        ],
        "why_trending": "High Reddit engagement + recent Brave articles + people asking specific questions",
        "recommended_action": "create_comparison_article"
    }
],
"next_steps": "Top 10 topics ready for keyword validation"
}
```

---

---

#### QUALITY CHECKS BEFORE FINISHING

---

---

- ✓ Are trend scores reasonable (50-100 range)?
  - ✓ Are keywords specific enough (not too broad)?
  - ✓ Did we find topics in multiple sources (validation)?
  - ✓ Are content angles clear and actionable?
  - ✓ Did we capture the "why" each topic is trending?
- 
- 

#### EXECUTION SCHEDULE

---

---

This prompt should run every 6 hours. Set OpenClaw to:

- Continuous mode: ON
- Run interval: 6 hours
- Retry on failure: Yes (3 attempts)
- Notify on error: Yes

After saving results, the system will automatically trigger keyword validation for the top topics.

---

---

#### OUTPUT SUMMARY

---

---

At the end, print a summary:

```
"Trend Discovery Complete
- Topics found: X
- Average trend score: X
- Top topic: [topic name] (score: X)
- Sources used: Brave, Reddit [, NewsAPI]
- Next: Keyword validation for top 10 topics
- Results saved to: /seo-automation/trends/YYYY-MM-DD_HH-MM.json"
```

## What This Prompt Does

17. **Searches multiple sources** - Brave Search, Reddit, optionally NewsAPI
18. **Calculates trend scores** - Based on recency, engagement, and cross-source validation
19. **Extracts keywords automatically** - From titles, posts, and discussions
20. **Identifies content angles** - Tutorial vs comparison vs guide

21. **Saves structured data** - Ready for next phase

## Expected Output

You'll get a JSON file with 20-50 trending topics, scored and ranked, with specific keywords and content angles already identified.

## Phase 2: Keyword Research & Validation

This prompt takes trending topics and validates which keywords are actually worth targeting.

### The Complete Prompt

```
KEYWORD RESEARCH & VALIDATION AGENT - Run Daily

Read the latest trend discovery results from:
/seo-automation/trends/ (use the most recent file)

Also read your niche configuration from:
/seo-automation/config/niches.json

YOUR MISSION:
Take the top 10 trending topics and validate which keywords have the best
SEO opportunity (high search volume, low competition, real interest).



---


STEP 1: EXPAND KEYWORDS


---


For each of the top 10 topics by trend_score:

Generate 15-20 keyword variations including:

A. QUESTION FORMATS:
- "how to [topic]"
- "what is [topic]"
- "why [topic]"
- "when to use [topic]"
- "where to find [topic]"

B. COMPARISON KEYWORDS:
- "[topic] vs [alternative]"
- "[topic] versus [alternative]"
- "best [topic]"
- "[topic] comparison"

C. LONG-TAIL VARIATIONS:
- "[topic] for [audience]"
- "[topic] for [use case]"
- "[topic] [year]"
- "free [topic]"
- "best [topic] [year]"
```

D. TUTORIAL/GUIDE KEYWORDS:

- "[topic] tutorial"
- "[topic] guide"
- "[topic] examples"
- "learn [topic]"
- "[topic] step by step"

Example: If topic is "AI agent automation"

Generate:

- how to build AI agents
- what is AI agent automation
- AI agent vs chatbot
- best AI agents 2025
- AI agents for business
- AI automation tutorial
- free AI agent tools
- AI agent examples
- ... (15-20 total)

---

---

**STEP 2: VALIDATE WITH BRAVE SEARCH**

---

---

For each keyword variation, use Brave Search API to validate:

A. SEARCH VOLUME PROXY:

Use the fetch tool to search Brave:

<https://api.search.brave.com/res/v1/web/search?q=KEYWORD&count=20>

Check the response:

- "query.total" field = total results (indicates search volume)
- Higher total = more people searching

SCORING:

- 10M+ results = 40 points (high volume)
- 1M-10M results = 30 points (medium volume)
- 100K-1M results = 20 points (low-medium volume)
- <100K results = 10 points (low volume)

B. GET RELATED SEARCHES:

From the Brave API response, check:

- "query.related\_searches" array

More related searches = more search interest

SCORING:

- 8+ related searches = 20 points
- 4-7 related searches = 15 points
- 1-3 related searches = 10 points
- 0 related searches = 5 points

C. ANALYZE TOP RESULTS:

Look at the top 10 results from Brave:

Check for:

- Domain Authority (look for .gov, .edu, major brands)
- Content freshness (check dates in descriptions)
- Result diversity (all listicles? all tutorials?)

Calculate COMPETITION LEVEL:

- 5+ high-authority domains in top 10 = HIGH competition (10 points)
- 2-4 high-authority domains = MEDIUM competition (20 points)
- 0-1 high-authority domains = LOW competition (30 points)

Why? If all top results are Forbes, NYT, etc. = hard to rank  
If top results are small blogs = easier to rank

---

---

### STEP 3: REDDIT VALIDATION

---

---

Search Reddit for each keyword to confirm real interest:

Use Reddit search API:

<https://www.reddit.com/search.json?q=KEYWORD&sort=relevance&t=month&limit=50>

Extract:

- Number of posts mentioning keyword in last month
- Average upvotes on those posts
- Number of comments (indicates real discussion)

SCORING:

Post count in last month:

- 20+ posts = 20 points (high interest)
- 10-19 posts = 15 points
- 5-9 posts = 10 points
- 1-4 posts = 5 points
- 0 posts = 0 points (maybe too niche or wrong keyword)

Average engagement (upvotes + comments):

- Avg 100+ engagement = 10 bonus points
- Avg 50-99 engagement = 5 bonus points

---

---

### STEP 4: COMPETITION CHECK - "ALLINTITLE" PROXY

---

---

Use Brave Search to check direct competition:

Search: allintitle:YOUR\_KEYWORD

This shows pages with your EXACT keyword in the title.

Use the fetch tool:

[https://api.search.brave.com/res/v1/web/search?q=allintitle:YOUR\\_KEYWORD&count=1](https://api.search.brave.com/res/v1/web/search?q=allintitle:YOUR_KEYWORD&count=1)

Check "query.total" field for result count.

SCORING (inverse - fewer is better):  
- <1,000 results = 30 points (EXCELLENT - low competition)  
- 1,000-5,000 results = 25 points (GOOD)  
- 5,000-20,000 results = 15 points (MEDIUM)  
- 20,000-50,000 results = 10 points (HARD)  
- 50,000+ results = 5 points (VERY HARD)

---

---

#### STEP 5: CALCULATE VALIDATION SCORE

---

---

For each keyword, sum all scores:

VALIDATION SCORE =  
    Search Volume Score (max 40)  
    + Related Searches Score (max 20)  
    + Competition Level Score (max 30)  
    + Reddit Post Count Score (max 20)  
    + Reddit Engagement Bonus (max 10)  
    + AllInTitle Score (max 30)

MAXIMUM POSSIBLE: 150 points

CLASSIFICATION:

- 120+ points = EXCELLENT (target immediately)
  - 90-119 points = GOOD (strong candidate)
  - 70-89 points = MEDIUM (consider if relevant)
  - Below 70 = SKIP (not worth the effort)
- 
- 

#### STEP 6: IDENTIFY CONTENT GAPS

---

---

For keywords scoring 90+:

Use Brave to fetch the top 5 ranking articles:

<https://api.search.brave.com/res/v1/web/search?q=KEYWORD&count=5>

For each top result:

1. Fetch the actual page content using the fetch tool
2. Extract:
  - Main headings (H2, H3)
  - Word count (approximate)
  - Topics covered
  - What they're missing (from Reddit questions)

IDENTIFY GAPS:

- Questions asked on Reddit but NOT answered in top articles
- Subtopics mentioned but not explained in depth
- Recent updates missing (if articles are >1 year old)
- Practical examples missing

---

---

## STEP 7: DETERMINE CONTENT STRATEGY

---

---

For each validated keyword, decide the best approach:

STRATEGY OPTIONS:

1. "comprehensive\_guide" - Be the best, most complete resource
2. "quick\_tutorial" - Fast, actionable how-to
3. "comparison\_review" - Compare 5-10 options with pros/cons
4. "beginner\_friendly" - Simplify for newcomers
5. "advanced\_deep\_dive" - Technical, in-depth for experts
6. "case\_study" - Real example with specific results
7. "updated\_2025" - Fresher than competitors

Pick based on:

- What's ranking (if all guides → try comparison)
- What Reddit wants (lots of "best X" questions → comparison)
- Content gaps (missing beginner info → beginner guide)

---

---

## STEP 8: SAVE RESULTS

---

---

Save to /seo-automation/keywords/YYYY-MM-DD\_research.json:

```
{  
    "timestamp": "2025-02-26T10:00:00Z",  
    "research_session_id": "unique-id",  
    "topics_analyzed": 10,  
    "keywords_generated": 180,  
    "keywords_validated": 180,  
    "keywords_recommended": 12,  
  
    "recommendations": [  
        {  
            "keyword": "how to build AI agents",  
            "parent_topic": "AI agent automation",  
            "validation_score": 127,  
            "breakdown": {  
                "search_volume_score": 40,  
                "related_searches_score": 20,  
                "competition_score": 30,  
                "reddit_score": 20,  
                "allintitle_score": 17  
            },  
            "metrics": {  
                "estimated_monthly_searches": "10,000-50,000",  
                "competition_level": "medium",  
                "allintitle_count": 3200,  
                "reddit_monthly_posts": 23,  
                "reddit_avg_engagement": 145  
            },  
            "content_strategy": "comprehensive_guide",  
            "target_length": "2000-2500 words",  
            "validation_score": 127  
        }  
    ]  
}
```

```

    "content_gaps_to_fill": [
        "Step-by-step setup tutorial (missing from top 3 results)",
        "Cost comparison (people asking on Reddit)",
        "Common mistakes section (from Reddit complaints)",
        "Integration with existing tools (Reddit question)"
    ],
    "target_headings": [
        "What Are AI Agents?",
        "How AI Agents Differ from Chatbots",
        "Step-by-Step: Building Your First AI Agent",
        "Best Tools for Building AI Agents",
        "Common Mistakes to Avoid",
        "AI Agent Use Cases and Examples",
        "Cost Comparison: Free vs Paid Tools"
    ],
    "competitors_to_beat": [
        {
            "url": "example.com/ai-agents",
            "word_count": 1500,
            "missing": "no cost info, no examples",
            "strength": "good structure",
            "rank": 1
        }
    ],
    "recommended_priority": "HIGH",
    "why_recommended": "High validation score + clear content gaps + strong Reddit interest + medium competition = excellent opportunity"
},
],
"next_steps": "Create articles for top 5 recommended keywords",
"stats": {
    "excellent_keywords": 5,
    "good_keywords": 7,
    "medium_keywords": 12,
    "skipped_keywords": 156
}
}

```

#### STEP 9: PRIORITIZE FOR CONTENT CREATION

---

From your recommendations, select top 3-5 for immediate content creation:

PRIORITY RULES:

1. Score 120+ = Create within 24 hours
2. Score 90-119 + HIGH trend\_score from Phase 1 = Create within 48 hours
3. Score 90-119 + MEDIUM trend\_score = Queue for next week

Save priority queue to /seo-automation/keywords/priority\_queue.json:

```
{
    "updated": "2025-02-26T10:00:00Z",
    "queue": [

```

```
{  
    "keyword": "how to build AI agents",  
    "validation_score": 127,  
    "priority": "IMMEDIATE",  
    "create_by": "2025-02-27",  
    "content_strategy": "comprehensive_guide"  
}  
]  
}
```

---

---

#### QUALITY CHECKS

---

---

Before finishing, verify:

- ✓ Did we expand each topic to 15-20 keywords?
  - ✓ Are validation scores based on real data (not guesses)?
  - ✓ Did we check competition properly (allintitle)?
  - ✓ Are content strategies specific and actionable?
  - ✓ Did we identify actual content gaps from top results?
  - ✓ Are recommended keywords diverse (not all similar)?
- 
- 

#### OUTPUT SUMMARY

---

---

Print final summary:

```
"Keyword Research Complete  
- Topics analyzed: X  
- Keywords validated: X  
- Excellent opportunities: X (score 120+)  
- Good opportunities: X (score 90-119)  
- Top recommended keyword: [keyword] (score: X)  
- Content strategies identified: [types]  
- Ready for content creation: X keywords  
- Results saved to: /seo-automation/keywords/YYYY-MM-DD_research.json"
```

## What This Prompt Does

22. **Expands each topic** into 15-20 keyword variations
23. **Validates with real data** from Brave Search API and Reddit
24. **Scores objectively** using multiple factors (volume, competition, interest)
25. **Identifies content gaps** by analyzing top-ranking competitors
26. **Recommends strategy** for each keyword (guide vs tutorial vs comparison)
27. **Prioritizes** keywords for immediate content creation

## Expected Output

A JSON file with 5-12 validated, high-opportunity keywords ready for content creation, complete with content strategies and competitor analysis.

## Phase 3: Content Creation

This is where OpenClaw writes your SEO-optimized articles.

### The Complete Prompt

```
CONTENT CREATION AGENT - Run Daily (1-2 Articles)

Read inputs from:
- /seo-automation/keywords/priority_queue.json (get next keyword to write)
- /seo-automation/config/api_keys.json (for fetching competitor content)

YOUR MISSION:
Create a comprehensive, SEO-optimized article that beats the current top-ranking content.



---


STEP 1: SELECT KEYWORD & LOAD RESEARCH


---


1. Read the priority queue
2. Select the highest priority keyword that hasn't been created yet
3. Load the full keyword research data for this keyword
4. Note the content strategy, target length, and content gaps



---


STEP 2: DEEP COMPETITOR ANALYSIS


---


Use Brave Search to get top 10 results:

https://api.search.brave.com/res/v1/web/search?q=YOUR\_KEYWORD&count=10

For each of the top 3 results:

A. FETCH FULL CONTENT:
Use the fetch tool to retrieve the complete page

B. EXTRACT STRUCTURE:
- All H2 headings
- All H3 headings
- Approximate word count
- Key points covered in each section
- Statistics/data points cited
- Examples provided
- Images/visuals used

C. IDENTIFY STRENGTHS:
What makes this article rank #1?
- Comprehensive coverage?
- Great examples?
- Recent updates?
- Strong formatting?
```

D. IDENTIFY WEAKNESSES:

- What's missing or could be better?
- Outdated information?
  - Missing key topics?
  - No examples?
  - Poor structure?
  - No actionable steps?
- 
- 

STEP 3: GATHER SUPPORTING DATA

---

---

Search for statistics and expert insights:

A. FIND STATISTICS:

Use Brave Search:  
"[keyword] statistics 2025"  
"[keyword] data"  
"[keyword] study"

Collect 3-5 compelling data points to cite

B. FIND EXPERT QUOTES:

Search for:  
"[keyword] expert opinion"  
"[keyword] [industry leader] on"

Note: Always attribute properly

C. FIND RECENT EXAMPLES:

Look for case studies, real-world examples, success stories

Search:  
"[keyword] examples"  
"[keyword] case study"  
"companies using [keyword]"

---

---

STEP 4: CREATE DETAILED OUTLINE

---

---

Build an outline that beats competitors:

ARTICLE STRUCTURE:

Title: [Main Keyword + Power Word]

Examples:

- "How to Build AI Agents: Complete 2025 Guide"
- "Best AI Agents for Business (Tested & Compared)"
- "AI Agent Automation: Step-by-Step Tutorial"

Meta Description: [150-160 characters with keyword in first 100]

Formula: [Benefit] + [Keyword] + [Unique angle] + [Call to action]

Example: "Learn how to build AI agents from scratch. Our complete 2025

guide covers setup, tools, and real examples. Start automating today."

---

## INTRODUCTION (150-200 words)

### 1. Hook with statistic or problem

Example: "73% of businesses are planning to use AI agents in 2025, but only 12% know how to build them."

### 2. State what the article will cover

Example: "This guide will walk you through building your first AI agent, from choosing tools to deployment."

### 3. Include main keyword naturally in first paragraph

Example: "Building AI agents doesn't require a PhD. Here's everything you need to know."

---

## MAIN CONTENT SECTIONS

For each major topic:

H2: [Descriptive Heading with Semantic Keyword]

Opening paragraph: Context and why it matters

H3: [Specific Point #1]

- Explanation (2-3 paragraphs)
- Example or data point
- Actionable takeaway

H3: [Specific Point #2]

- Explanation
- Example
- Takeaway

[Repeat H3s as needed - typically 2-4 per H2]

---

## REQUIRED SECTIONS (In This Order):

H2: What Is [Main Topic]?

- Definition
- Why it matters
- Common misconceptions

H2: How [Main Topic] Works

- Technical explanation (simplified)
- Visual description (describe process)
- Real-world analogy

H2: Benefits of [Main Topic]

- 4-6 specific benefits
- Data points supporting each
- Real examples

H2: How to [Main Topic]: Step-by-Step

- H3: Step 1: [Specific Action]
- What to do
  - Why this step matters
  - Common mistakes to avoid
  - Expected outcome

H3: Step 2: [Next Action]

[Same format]

[Continue 5-10 steps as needed]

H2: Best Tools/Options for [Main Topic]

- H3: Tool/Option #1: [Name]
- What it does
  - Pros
  - Cons
  - Best for: [use case]
  - Price: [range]

[Repeat for 3-5 tools]

H2: Common Mistakes to Avoid

- List 5-7 mistakes
- Each with explanation and how to avoid

H2: Advanced Tips/Strategies (if appropriate)

- 3-5 advanced techniques
- For experienced users

H2: Frequently Asked Questions

- H3: [Question 1 from Reddit/search data]  
Answer in 2-3 paragraphs

H3: [Question 2]

Answer

[Include 5-7 real questions people ask]

---

CONCLUSION (100-150 words)

1. Summarize key takeaways (3-4 bullet points)

2. Restate main benefit

3. Call to action

Examples:

- "Ready to build your first AI agent? Start with [specific first step]."
- "Which tool will you try first? Let me know in the comments."
- "For more AI automation guides, check out [related article]."

---

STEP 5: WRITE THE ARTICLE

---

Now write the full article following these rules:

**WRITING STYLE:**

- Tone: Authoritative but conversational (like explaining to a smart friend)
- POV: Second person ("you") for engagement
- Sentence length: Vary (mix short punchy sentences with longer explanatory ones)
- Paragraph length: 2-4 sentences max (mobile-friendly)
- Voice: Active voice (not passive)
- Examples: Specific and real (not vague hypotheticals)

**SEO OPTIMIZATION:**

- Main keyword appears in:
  - ✓ Title (at the beginning if possible)
  - ✓ First paragraph (first 100 words)
  - ✓ At least one H2 heading
  - ✓ Throughout content (naturally, not forced)
  - ✓ Conclusion
  - ✓ Meta description
- Keyword density: 1-2% (for 2000 word article = 20-40 times)
- Use LSI keywords (related terms) throughout
- Use semantic variations (don't repeat exact phrase robotically)

**CONTENT QUALITY RULES:**

- ✓ Every claim needs support (data, expert quote, or example)
- ✓ Every section needs actionable takeaway
- ✓ Include specific numbers (not "many" or "several")
- ✓ Use real company names and tools (not "Tool X" or "Company Y")
- ✓ Add context for acronyms on first use
- ✓ Link to authoritative sources for statistics
- ✓ Anticipate and answer objections

**FORMATTING FOR READABILITY:**

- ✓ Use transition phrases between sections
- ✓ Include bullet points for lists (not numbered unless sequential)
- ✓ Bold important terms on first mention
- ✓ Use analogies for complex concepts
- ✓ Break up walls of text
- ✓ Add "Pro Tip:" callouts for expert insights
- ✓ Use "Note:" for important caveats

**INTERNAL LINKING OPPORTUNITIES:**

As you write, mark places for internal links:

[INTERNAL\_LINK: anchor text - topic it should link to]

Example:

"Before building AI agents, you'll need to understand [INTERNAL\_LINK: prompt engineering basics - article about prompt engineering]."

---

STEP 6: ADD STRUCTURED DATA (FAQ SCHEMA)

---

After the article, add FAQ schema in JSON-LD format:

```
<script type="application/ld+json">
{
  "@context": "https://schema.org",
  "@type": "FAQPage",
  "mainEntity": [
    {
      "@type": "Question",
      "name": "What are AI agents?",
      "acceptedAnswer": {
        "@type": "Answer",
        "text": "AI agents are autonomous software programs that can perform tasks, make decisions, and interact with systems without human intervention. They use large language models to understand context and execute complex workflows."
      }
    },
    [Include all FAQ questions from your article]
  ]
}
</script>
```

---

---

#### STEP 7: SAVE THE ARTICLE

---

---

Save to /seo-automation/articles/drafts/YYYY-MM-DD\_keyword-slug.md

Use this format:

```
---
title: "How to Build AI Agents: Complete 2025 Guide"
meta_description: "Learn how to build AI agents from scratch. Our complete guide covers tools, steps, and real examples. Start building today."
slug: "how-to-build-ai-agents-guide"
keyword: "how to build AI agents"
secondary_keywords: ["AI agent tutorial", "build AI agents", "AI automation guide"]
date: "2025-02-26"
author: "Your Name/Brand"
category: "AI Automation"
status: "draft"
word_count: 2347
reading_time: "9 minutes"
internal_links_needed: 3
external_sources_cited: 5
seo_score: 95
---
```

# How to Build AI Agents: Complete 2025 Guide

[Full article content here in markdown format]

```
## What Are AI Agents?  
  
AI agents are autonomous software programs...  
  
[Continue full article]
```

---

#### ARTICLE METADATA:

- Word count: 2347
- Keyword density: 1.7% (40 instances)
- Reading level: Grade 10
- Readability score: 68 (Good)
- Internal links added: 3
- External sources cited: 5
- FAQ questions: 7
- Images needed: 5 (described in comments)

#### CONTENT GAPS FILLED:

- ✓ Step-by-step tutorial (missing from top 3 results)
- ✓ Cost comparison section (Reddit request)
- ✓ Common mistakes section (Reddit complaints)
- ✓ Integration examples (Reddit question)
- ✓ Updated for 2025 (competitors from 2023)

#### SEO CHECKLIST:

- ✓ Keyword in title
- ✓ Keyword in first 100 words
- ✓ Keyword in meta description
- ✓ Keyword in H2 heading
- ✓ Proper H2/H3 hierarchy
- ✓ Alt text for images (TBD)
- ✓ FAQ schema included
- ✓ Internal linking opportunities marked
- ✓ External links to authority sites
- ✓ Call to action in conclusion

#### COMPETITIVE ANALYSIS:

Current #1 ranking article: 1,850 words  
Our article: 2,347 words ✓  
Their structure: Basic overview  
Our structure: Step-by-step + tools + FAQs ✓  
Their examples: Generic  
Our examples: Specific companies and results ✓  
Their update: 2023  
Our update: 2025 ✓

#### NEXT STEPS:

1. Add featured image
2. Resolve internal links
3. Add screenshots/diagrams
4. Final SEO review
5. Publish

---

---

## STEP 8: QUALITY ASSURANCE CHECK

---

Before marking as ready for publishing, verify:

CONTENT QUALITY:

- Article is better than current #1 ranking result
- All claims are supported with data or examples
- Every section provides actionable value
- Examples are specific (not generic)
- Writing flows naturally (not robotic)
- No fluff or filler content
- Answers the search intent completely

SEO OPTIMIZATION:

- Title is compelling and includes keyword
- Meta description is 150-160 characters
- Keyword appears naturally throughout
- Semantic keywords included
- Internal linking opportunities marked
- FAQ section addresses real questions
- Proper heading hierarchy (H1 → H2 → H3)

READABILITY:

- Short paragraphs (2-4 sentences)
- Varied sentence length
- Bullet points used appropriately
- Bold text for key terms
- No jargon without explanation
- Mobile-friendly formatting

COMPETITIVE ADVANTAGE:

- Longer than top 3 competitors
- More comprehensive coverage
- More recent/updated information
- Better examples and data
- Clearer structure and formatting

---

---

## STEP 9: UPDATE TRACKING

---

Update /seo-automation/articles/drafts/index.json:

```
{  
  "articles": [  
    {  
      "filename": "2025-02-26_how-to-build-ai-agents-guide.md",  
      "keyword": "how to build AI agents",  
      "validation_score": 127,  
      "word_count": 2347,  
      "status": "ready_for_review",  
      "created": "2025-02-26T15:30:00Z",  
      "seo_score": 95,  
    }  
  ]  
}
```

```
        "competitive_advantage": "Longer + more examples + 2025 updated +  
better structure"  
    }  
]  
}  
  
=====  
OUTPUT SUMMARY  
=====  
  
"Article Creation Complete  
  
Keyword: how to build AI agents  
Validation Score: 127  
Strategy: comprehensive_guide  
  
Article Stats:  
- Word count: 2,347 words  
- Reading time: 9 minutes  
- SEO score: 95/100  
- Keyword density: 1.7%  
- Headings: 15 (H2: 8, H3: 7)  
- Internal links: 3 marked  
- FAQs answered: 7  
- Sources cited: 5  
  
Competitive Position:  
- Beats #1 result in length (+497 words)  
- Covers 3 additional topics  
- More recent (2025 vs 2023)  
- Better examples and data  
  
Saved to:  
/seo-automation/articles/drafts/2025-02-26_how-to-build-ai-agents-guide.m  
d  
  
Status: Ready for publishing phase"
```

## What This Prompt Does

28. **Analyzes top competitors** - Fetches and dissects top-ranking articles
29. **Gathers supporting data** - Finds statistics, quotes, and examples
30. **Creates superior content** - Writes longer, more comprehensive articles
31. **Optimizes for SEO** - Natural keyword usage, proper structure, schema markup
32. **Ensures quality** - Multiple checks before marking ready

## Expected Output

A complete 1,500-2,500 word article that's better than current top-ranking content, fully optimized for SEO, with proper formatting and structure.

## Phase 4: Publishing Automation

This prompt handles the final step: getting your content live and indexed.

### The Complete Prompt

```
PUBLISHING & OPTIMIZATION AGENT - Run Daily

Read inputs from:
- /seo-automation/articles/drafts/ (get articles ready for publishing)
- /seo-automation/config/api_keys.json (WordPress credentials)

YOUR MISSION:
Review, optimize, and publish completed articles to WordPress, then
ensure they're indexed by search engines.



---


STEP 1: SELECT ARTICLE FOR PUBLISHING


---


1. Check /seo-automation/articles/drafts/index.json
2. Find articles with status "ready_for_review"
3. Select the highest validation_score article
4. Load the full article content



---


STEP 2: FINAL SEO & QUALITY CHECK


---


Run through complete checklist:

A. SEO VERIFICATION:
Read the article and verify:

 Title includes main keyword (at beginning if possible)
 Meta description is 150-160 characters
 Main keyword appears in first 100 words
 Keyword appears in at least one H2
 Keyword in conclusion
 Keyword density between 1-2%
 LSI keywords present throughout
 Proper heading hierarchy (H1 → H2 → H3)
 FAQ section exists with 5-7 questions
 FAQ schema markup present

B. CONTENT QUALITY:

 Word count meets minimum (1500+)
 No spelling or grammar errors
 All paragraphs are 2-4 sentences
 Bullet points used appropriately
 Examples are specific (not generic)
 All claims are supported
 Actionable takeaways in each section
```

- Smooth transitions between sections
- Strong call-to-action in conclusion

C. TECHNICAL CHECK:

- All markdown formatting correct
- No broken syntax
- Internal link markers present
- External links would work (check format)
- Image placeholders noted

If any checks fail, note them and fix before proceeding.

---

---

STEP 3: RESOLVE INTERNAL LINKS

---

Find all [INTERNAL\_LINK: ...] markers in the article.

For each marker:

A. CHECK IF TARGET ARTICLE EXISTS:

Search /seo-automation/articles/published/ for relevant article

If exists:

- Replace marker with actual markdown link
- Format: [anchor text] (<https://yourblog.com/article-slug>)

If doesn't exist:

- Remove the marker but keep the anchor text
- Add to /seo-automation/internal-links-todo.json for future linking

B. ADD NEW INTERNAL LINKS:

Look for opportunities to link to your published content:

- Scan article for topics you've covered before
- Check /seo-automation/articles/published/index.json
- Add 2-3 relevant internal links if opportunities exist

Format: Natural anchor text linking to relevant articles

---

---

STEP 4: PREPARE FOR WORDPRESS

---

A. CONVERT MARKDOWN TO HTML:

Convert all markdown syntax:

- ## Heading → <h2>Heading</h2>
- ### Heading → <h3>Heading</h3>
- \*\*bold\*\* → <strong>bold</strong>
- \*italic\* → <em>italic</em>
- [link] (url) → <a href="url">link</a>
- Bullet lists → <ul><li>item</li></ul>
- Numbered lists → <ol><li>item</li></ol>

Add paragraph tags: <p>content</p>

B. OPTIMIZE EXCERPT:

Create a compelling 155-character excerpt:

- Pull from introduction or meta description
- Make it click-worthy
- Include main benefit
- End with call-to-action or intrigue

C. GENERATE FEATURED IMAGE PROMPT:

Create an AI image generation prompt:

"Create a professional featured image for an article about [topic].

Style: Modern, clean, minimalist

Colors: Blue, white, accent of [color]

Elements: [relevant icons or concepts]

Mood: Professional and trustworthy

Text: No text overlay

Dimensions: 1200x630px (landscape)

Quality: High resolution, suitable for blog header"

Save this prompt to /seo-automation/images/prompts/article-slug.txt

D. PREPARE TAGS & CATEGORIES:

Based on article content:

- Main category: [from article metadata]
- Tags: Extract 3-5 relevant tags from content  
(Tools mentioned, concepts covered, target audience)

---

STEP 5: PUBLISH TO WORDPRESS

---

Use WordPress REST API to publish:

A. AUTHENTICATE:

Create auth header:

Authorization: Basic base64(username:application\_password)

From /seo-automation/config/api\_keys.json

B. CREATE POST:

POST https://yourblog.com/wp-json/wp/v2/posts

Headers:

Authorization: Basic [base64\_credentials]

Content-Type: application/json

Body:

```
{
    "title": "Article Title Here",
    "content": "Full HTML content here",
    "excerpt": "155-char excerpt",
    "status": "publish",
    "categories": [CATEGORY_ID],
    "tags": [TAG_ID_1, TAG_ID_2, TAG_ID_3],
    "meta": {
        "meta_description": "SEO meta description",
        "focus_keyword": "main keyword"
    },
    "slug": "article-slug",
    "date": "2025-02-26T10:00:00"
}
```

Note: You may need to first get category/tag IDs:  
 GET <https://yourblog.com/wp-json/wp/v2/categories>  
 GET <https://yourblog.com/wp-json/wp/v2/tags>

#### C. HANDLE RESPONSE:

```
If successful (status 201):
- Extract post ID from response
- Extract post URL
- Save for next steps

If failed:
- Log error details
- Retry once
- If still fails, alert and save for manual review
```

### STEP 6: POST-PUBLISH OPTIMIZATION

---

#### A. MOVE TO PUBLISHED FOLDER:

Move article from /seo-automation/articles/drafts/  
 to /seo-automation/articles/published/

Rename: YYYY-MM-DD\_slug\_POST-ID.md

#### B. UPDATE TRACKING:

Update /seo-automation/articles/published/index.json:

```
{
    "articles": [
        {
            "title": "How to Build AI Agents: Complete 2025 Guide",
            "keyword": "how to build AI agents",
            "validation_score": 127,
            "url": "https://yourblog.com/how-to-build-ai-agents-guide",
            "post_id": 12345,
            "published_date": "2025-02-26T10:00:00Z",
        }
    ]
}
```

```

        "word_count": 2347,
        "category": "AI Automation",
        "status": "published",
        "performance": {
            "indexed": false,
            "impressions": 0,
            "clicks": 0,
            "avg_position": null,
            "last_checked": null
        }
    }
]
}

```

#### C. UPDATE INTERNAL LINK DATABASE:

Add to /seo-automation/internal-links/database.json:

```

{
  "url": "https://yourblog.com/how-to-build-ai-agents-guide",
  "title": "How to Build AI Agents",
  "main_topics": [
    "AI agents",
    "automation",
    "AI tools",
    "tutorial"
  ],
  "suggested_anchor_texts": [
    "building AI agents",
    "AI agent tutorial",
    "how to create AI agents",
    "AI automation guide"
  ]
}

```

This database will be used by future articles for internal linking.

---

#### STEP 7: SUBMIT FOR INDEXING

---

#### A. GOOGLE SEARCH CONSOLE (if configured):

If you have GSC API set up:

```

POST https://indexing.googleapis.com/v3/urlNotifications:publish
Headers:
  Authorization: Bearer YOUR_ACCESS_TOKEN
  Content-Type: application/json
Body:
{
  "url": "https://yourblog.com/article-url",
  "type": "URL_UPDATED"
}

```

```
If successful, mark as submitted in tracking.
```

B. MANUAL INDEXING FALBACK:

If no GSC API:

Create a list in /seo-automation/indexing/pending.txt:  
<https://yourblog.com/article-url>

Add instructions:

"Submit this URL manually to Google Search Console → URL Inspection → Request Indexing"

C. SITEMAP UPDATE:

If using WordPress, the sitemap usually auto-updates.

If using static site generator, trigger rebuild.

Verify sitemap includes new post:

<https://yourblog.com/sitemap.xml>

---

STEP 8: CREATE SOCIAL MEDIA SNIPPETS (OPTIONAL)

---

Generate ready-to-post social snippets:

A. TWITTER/X POST:

Format: [Hook] + [Teaser] + [Link]

Character limit: 280

Example:

"73% of businesses are planning AI agents in 2025—but only 12% know how to build them.

Here's everything you need to start 

<https://yourblog.com/how-to-build-ai-agents-guide>"

B. LINKEDIN POST:

Format: [Problem] + [Insight] + [Value prop] + [CTA] + [Link]

Length: 150-300 words

Example:

"Most businesses want to use AI agents but don't know where to start.

After building AI automation systems for 14 SaaS products, I've learned what actually works.

I just published a complete guide covering:

- What AI agents really are (beyond the hype)
- Step-by-step setup process
- Best tools compared (free & paid)

- Common mistakes that waste weeks
- Real examples from companies using them

Everything you need to build your first AI agent is here:  
<https://yourblog.com/how-to-build-ai-agents-guide>

What's your biggest question about AI agents?"

#### C. REDDIT POST (for relevant subreddits):

Title: [Attention-grabbing + Value]  
 Context comment: [Be helpful, not promotional]

Example Title:

"I built AI agents for 14 SaaS companies. Here's what actually works (complete guide)"

Context Comment:

"Hey r/Entrepreneur - I've spent 6 months building AI automation across multiple products. Lots of expensive mistakes and wins.

I put together a guide covering the practical stuff: tools, costs, real examples, what to avoid.

Not selling anything, just sharing what I learned: [link]

Happy to answer questions!"

Save all snippets to:

/seo-automation/social/YYYY-MM-DD\_article-slug\_snippets.txt

#### STEP 9: SCHEDULE PERFORMANCE CHECK

---

Add article to performance monitoring queue:

Update /seo-automation/monitoring/check\_schedule.json:

```
{
  "next_check": "2025-02-28T10:00:00Z",
  "articles_to_check": [
    {
      "url": "https://yourblog.com/how-to-build-ai-agents-guide",
      "published": "2025-02-26T10:00:00Z",
      "check_after_days": 2,
      "metrics_to_track": [
        "indexed",
        "ranking_position",
        "impressions",
        "clicks",
        "ctr"
      ]
    }
  ]
}
```

```
}

Check schedule:
- Day 2: Indexing status
- Day 7: Initial ranking position
- Day 14: Performance metrics
- Day 30: Full analysis



---


STEP 10: LOG & REPORT


---

  
Save complete publishing log:  
  
/seo-automation/logs/publishing/YYYY-MM-DD_HH-MM_article-slug.json  
  
{  
    "timestamp": "2025-02-26T10:30:00Z",  
    "article_title": "How to Build AI Agents: Complete 2025 Guide",  
    "keyword": "how to build AI agents",  
    "validation_score": 127,  
    "url": "https://yourblog.com/how-to-build-ai-agents-guide",  
    "wordpress_post_id": 12345,  
    "publication_status": "success",  
    "steps_completed": {  
        "seo_check": true,  
        "quality_check": true,  
        "internal_links_resolved": true,  
        "wordpress_published": true,  
        "tracking_updated": true,  
        "indexing_submitted": true,  
        "social_snippets_created": true  
    },  
    "article_stats": {  
        "word_count": 2347,  
        "seo_score": 95,  
        "internal_links": 5,  
        "external_links": 7,  
        "images_needed": 5  
    },  
    "next_actions": [  
        "Add featured image",  
        "Check indexing in 48 hours",  
        "Post to social media",  
        "Monitor performance after 7 days"  
    ]  
}  
  


---


OUTPUT SUMMARY


---

  
"Publishing Complete ✓  
  
Article: How to Build AI Agents: Complete 2025 Guide
```

```
URL: https://yourblog.com/how-to-build-ai-agents-guide  
Published: 2025-02-26 10:00 UTC
```

Publication Stats:

- WordPress Post ID: 12345
- Word count: 2,347
- SEO score: 95/100
- Internal links: 5
- Status: LIVE

Post-Publish Actions:

- ✓ Moved to published folder
- ✓ Tracking database updated
- ✓ Indexing submitted to Google
- ✓ Social snippets generated
- ✓ Performance monitoring scheduled

Next Steps:

- Add featured image (prompt saved)
- Check indexing status in 48h
- Post to social media (snippets ready)
- Monitor rankings starting day 7

Total time: 8 minutes

Cost: \$0.23 in API calls

Article #{{article\_count}} published this month"

## What This Prompt Does

33. **Final quality check** - Verifies SEO and content quality
34. **Resolves internal links** - Connects to your existing content
35. **Publishes to WordPress** - Handles complete API workflow
36. **Submits for indexing** - Gets Google to find your content fast
37. **Creates social snippets** - Ready-to-post promotional content
38. **Tracks everything** - Maintains complete publication logs

## Expected Output

Your article goes live on WordPress, gets submitted to Google for indexing, and you get ready-to-use social media snippets—all automated.

## Master Orchestrator Setup

Now we tie everything together into a single autonomous system.

## The Master Prompt

```
SEO CONTENT AUTOMATION - MASTER ORCHESTRATOR  
Continuous 24/7 Operation
```

Read configuration from:

```
- /seo-automation/config/api_keys.json
- /seo-automation/config/niches.json
```

YOUR MISSION:

Run a complete SEO content system autonomously. Check what needs to be done, execute the appropriate phase, and maintain system health.

---

SYSTEM STATE MANAGEMENT

---

First, check or create /seo-automation/system\_state.json:

```
{
  "last_updated": "2025-02-26T10:00:00Z",
  "current_phase": "idle",
  "system_health": "healthy",

  "last_runs": {
    "trend_discovery": "2025-02-26T04:00:00Z",
    "keyword_research": "2025-02-26T09:00:00Z",
    "content_creation": "2025-02-26T10:30:00Z",
    "publishing": "2025-02-26T10:45:00Z"
  },
  "daily_quotas": {
    "articles_created_today": 1,
    "articles_published_today": 1,
    "max_articles_per_day": 2,
    "reset_at": "2025-02-27T00:00:00Z"
  },
  "queues": {
    "topics_awaiting_research": 5,
    "keywords_awaiting_content": 3,
    "articles_awaiting_publishing": 1
  },
  "performance": {
    "total_articles_published": 23,
    "avg_validation_score": 112,
    "system_uptime_hours": 1440,
    "total_cost_usd": 52.80
  },
  "errors": {
    "last_error": null,
    "error_count_24h": 0,
    "consecutive_failures": 0
  }
}
```

---

DECISION LOGIC: WHAT TO RUN NEXT

---

Check the current time and system state, then decide:

```
PRIORITY 1: TREND DISCOVERY
IF 6+ hours since last_runs.trend_discovery:
    → Run PHASE 1 (Trend Discovery)
    → Update system_state.json

PRIORITY 2: KEYWORD RESEARCH
IF topics_awaiting_research > 0
AND keyword_research not run today:
    → Run PHASE 2 (Keyword Research)
    → Update system_state.json

PRIORITY 3: CONTENT CREATION
IF keywords_awaiting_content > 0
AND articles_created_today < max_articles_per_day:
    → Run PHASE 3 (Content Creation)
    → Increment articles_created_today
    → Update system_state.json

PRIORITY 4: PUBLISHING
IF articles_awaiting_publishing > 0
AND articles_published_today < max_articles_per_day:
    → Run PHASE 4 (Publishing)
    → Increment articles_published_today
    → Update system_state.json

PRIORITY 5: MAINTENANCE
IF none of the above:
    → Run system health checks
    → Clean up old logs
    → Generate performance reports if needed
    → Wait for next scheduled task
```

---

---

#### EXECUTION WORKFLOW

---

---

When running each phase:

1. UPDATE STATE TO "RUNNING":  
system\_state.json → current\_phase = "[phase\_name]"
2. EXECUTE THE PHASE:  
Run the complete prompt for that phase  
(Trend Discovery, Keyword Research, Content Creation, or Publishing)
3. HANDLE RESULTS:  
  
IF SUCCESSFUL:
  - Update last\_runs timestamp for this phase
  - Update queue counts
  - Log success
  - Update system\_state to "idle"

```
IF FAILED:  
- Log error details  
- Increment consecutive_failures  
- Wait 1 hour before retry  
- If consecutive_failures >= 3, alert and pause
```

#### 4. CHECK QUOTAS:

```
If daily quotas reached:  
- Pause article creation/publishing  
- Continue trend discovery and keyword research  
- Reset quotas at midnight UTC
```

#### 5. MAINTAIN LOGS:

```
Save execution log:  
/seo-automation/logs/orchestrator/YYYY-MM-DD.json
```

```
{  
    "date": "2025-02-26",  
    "executions": [  
        {  
            "timestamp": "2025-02-26T04:00:00Z",  
            "phase": "trend_discovery",  
            "status": "success",  
            "duration_seconds": 180,  
            "results": "47 topics discovered",  
            "cost_usd": 0.12  
        }  
    ]  
}
```

---

### SYSTEM HEALTH CHECKS

---

Every 6 hours, verify:

A. API CONNECTIVITY:

- Test Brave Search API
- Test Reddit API
- Test WordPress API

If any fails:

- Log error
- Retry in 30 minutes
- If 3 consecutive failures, alert

B. QUEUE HEALTH:

- Check if queues are growing (bottleneck indicator)
- If keywords\_awaiting\_content > 10, increase article quota
- If articles\_awaiting\_publishing > 5, prioritize publishing

C. COST MONITORING:

- Track API costs per phase
- Alert if daily cost exceeds \$10
- Suggest optimization if cost trending high

D. STORAGE MANAGEMENT:

- Clean up logs older than 30 days
- Archive published articles older than 90 days
- Remove duplicate tracking entries

PERFORMANCE REPORTING

---



---

Every 7 days, generate report:

```
/seo-automation/reports/weekly_YYYY-MM-DD.json

{
  "report_period": "2025-02-19 to 2025-02-26",
  "articles_published": 12,
  "avg_validation_score": 115,
  "avg_word_count": 2180,
  "total_keywords_targeted": 12,

  "top_performing_niches": [
    {"niche": "AI automation", "articles": 5, "avg_score": 121},
    {"niche": "SaaS tools", "articles": 4, "avg_score": 113}
  ],

  "source_effectiveness": {
    "brave_search": {"topics_found": 89, "topics_validated": 23},
    "reddit": {"topics_found": 56, "topics_validated": 19}
  },

  "system_performance": {
    "uptime_percentage": 99.2,
    "avg_execution_time": "4 minutes per phase",
    "errors_encountered": 2,
    "total_cost": "$6.80"
  },

  "recommendations": [
    "Increase focus on 'AI automation' niche (highest validation scores)",
    "Add subreddit r/nocode (mentioned frequently in trends)",
    "Consider increasing daily article quota to 3"
  ]
}
```

Send summary to user (or save to dashboard).

ERROR HANDLING & RECOVERY

---



---

When errors occur:

A. API FAILURES:

- Retry after 30 minutes
- If 3 failures, switch to fallback (if available)
- Log all failures to /seo-automation/logs/errors.json

B. RATE LIMITS:

- Respect Brave Search limits (2000/month on free tier)
- If approaching limit, reduce search frequency
- Track usage in system\_state.json

C. WORDPRESS FAILURES:

- Verify credentials
- Check site status
- Save article locally if can't publish
- Retry after 1 hour

D. CONTENT QUALITY ISSUES:

- If article fails quality check, regenerate
- If fails twice, skip keyword and log for review
- Alert user about repeated failures

E. SYSTEM CRASHES:

- On restart, check system\_state.json
- Resume from last known state
- Don't duplicate completed work

---

SELF-OPTIMIZATION

---

Track performance and improve over time:

A. LEARN FROM DATA:

- Which niches produce highest validation scores?
- Which sources find best trends?
- What content strategies work best?
- What article lengths rank fastest?

B. ADJUST OPERATIONS:

- Increase frequency for high-performing niches
- Reduce effort on low-performing sources
- Update content templates based on winners
- Modify keyword selection criteria

C. COST OPTIMIZATION:

- Track cost per article
- Identify expensive operations
- Find cheaper alternatives when possible
- Balance cost vs. quality

---

HUMAN CHECKPOINTS

---

Alert user when:

- System hasn't published in 48 hours
  - Error rate exceeds 10% in 24 hours
  - Daily cost exceeds \$10
  - Queue backlog exceeds 20 items
  - 10 articles published (milestone celebration)
  - Weekly report is ready
  - Validation scores dropping (avg < 90 for week)
- 
- 

#### EXECUTION SCHEDULE

---

---

Configure OpenClaw:

- Continuous mode: ON
  - Check interval: Every 1 hour
  - Max runtime per check: 30 minutes
  - Auto-retry on error: Yes (3 attempts, 30 min delay)
  - Daily budget cap: \$10
  - Notification webhook: [Optional - Slack/Discord]
- 
- 

#### STARTUP SEQUENCE

---

---

When system first starts:

1. Verify all directories exist
  2. Check all config files present
  3. Test API connections
  4. Load or create system\_state.json
  5. Check for incomplete tasks
  6. Resume from last known state
  7. Begin normal operation cycle
- 
- 

#### OUTPUT: CONTINUOUS STATUS UPDATES

---

---

Every hour, print status:

"SEO Automation System - Status Update

Time: 2025-02-26 14:00 UTC

Status: RUNNING

Current Phase: content\_creation

Today's Stats:

- Articles created: 1/2
- Articles published: 1/2
- Topics discovered: 47
- Keywords validated: 8

```
Queues:  
- Topics awaiting research: 5  
- Keywords awaiting content: 3  
- Articles awaiting publishing: 1  
  
Last 24 Hours:  
- Trend discoveries: 4 runs  
- Articles published: 1  
- Errors: 0  
- Cost: $2.80  
  
Next scheduled:  
- Trend discovery: in 2 hours  
- Keyword research: completed today  
- Content creation: starting now  
- Publishing: ready (1 in queue)  
  
System Health: ✓ All systems operational"
```

---

---

#### FINAL NOTES

---

---

This orchestrator:

- Runs 24/7 with minimal intervention
- Adapts to workload automatically
- Recovers from failures gracefully
- Optimizes itself over time
- Respects budget and rate limits
- Maintains complete logs
- Alerts when human input needed

The system is designed to be "set and forget" while remaining transparent and controllable.

## Setting Up The Orchestrator in OpenClaw

39. Create New Task in OpenClaw named "SEO Master Orchestrator"

40. Paste the Master Prompt above

41. Configure Task Settings:

- Continuous mode: ON
- Run interval: 1 hour
- Max tokens per run: 50,000
- Model: Claude Sonnet 4.5 (recommended) or GPT-4 Turbo
- Fallback: Claude Haiku 4.5
- Auto-retry: 3 attempts with 30 min delay
- Daily budget: \$10

42. Enable Required MCP Servers:

- fetch (for API calls)

- filesystem (for file management)

#### 43. Start the Task and monitor first 24 hours

### What Happens Next

The orchestrator will:

- Check what needs to be done every hour
- Execute the appropriate phase automatically
- Maintain all tracking and logs
- Alert you only when needed
- Publish 1-2 articles per day
- Discover new trends every 6 hours
- Optimize itself based on performance

## Monitoring & Optimization

### Daily Monitoring Dashboard

Create a simple monitoring script that reads system state:

**Check this file daily:**

/seo-automation/system\_state.json

**Key metrics to watch:**

- Articles published today vs. quota
- Queue sizes (should be 3-10 items each)
- Error count (should be near zero)
- Daily cost (should be \$2-5)

### Weekly Review Checklist

Every Sunday, review:

#### 44. Performance Report

- Location: /seo-automation/reports/weekly\_YYYY-MM-DD.json
- Check: Which niches performed best?
- Action: Adjust niche focus if needed

#### 45. Published Articles

- Location: /seo-automation/articles/published/index.json
- Check: Are articles ranking yet?
- Action: Note what's working

#### 46. Cost Analysis

- Location: /seo-automation/logs/orchestrator/

- Check: Cost per article
- Action: Optimize if exceeding budget

## 47. Error Log

- Location: /seo-automation/logs/errors.json
- Check: Recurring failures?
- Action: Fix configuration issues

# Optimization Strategies

##### If Validation Scores Are Low (<90)

**Problem:** Keywords aren't passing validation

**Solutions:**

- Add more subreddits to monitor
- Expand keyword variations (try more formats)
- Lower competition threshold
- Focus on more specific niches

##### If Articles Aren't Ranking

**Problem:** Content not performing in search

**Solutions:**

- Increase target word count (try 2500-3000)
- Add more internal links
- Improve examples and data
- Check if competitors updated (re-run analysis)
- Build backlinks manually

##### If System Cost Is Too High

**Problem:** Exceeding daily budget

**Solutions:**

- Reduce articles per day (2 → 1)
- Use cheaper model for trend discovery (GPT-3.5 or Haiku)
- Reduce competitor analysis depth (3 articles instead of 10)
- Lengthen trend discovery interval (12 hours instead of 6)

##### If Queue Backlog Growing

**Problem:** System can't keep up

**Solutions:**

- Increase daily article quota (2 → 3)
- Run multiple parallel instances for different niches
- Simplify content creation (shorter articles)
- Add more publishing automation

## Performance Benchmarks

### Good Performance Indicators:

- Validation scores averaging 100+
- 90%+ of keywords passing validation
- Articles ranking page 2-3 within 1 week
- Cost per article: \$0.15-0.30
- System uptime: 95%+
- Error rate: <5%

### Excellent Performance:

- Validation scores averaging 120+
- Articles ranking page 1-2 within 2 weeks
- Cost per article: <\$0.20
- System uptime: 99%+
- Organic traffic growing 20%+ monthly

## Troubleshooting

### Common Issues & Solutions

##### Issue: "Brave Search API returning 429 (rate limit)"

**Cause:** Exceeded free tier limit (2000 queries/month)

**Solution:**

48. Check usage: Look at Brave API dashboard
49. Reduce search frequency: Change interval to 12 hours
50. Reduce searches per niche: 3 searches → 2 searches
51. Upgrade to paid tier (\$5/month for 20,000 queries)

##### Issue: "Reddit API returning 401 (unauthorized)"

**Cause:** OAuth token expired or invalid credentials

**Solution:**

52. Re-generate access token manually
53. Verify client ID and secret in config
54. Check Reddit app is still active at reddit.com/prefs/apps
55. Ensure User-Agent header is set correctly

##### Issue: "WordPress publishing fails with 403"

**Cause:** Invalid application password or permissions

**Solution:**

56. Regenerate application password in WordPress
57. Verify username is correct (not email)

58. Check user role has publishing permissions

59. Test with curl to isolate issue:

```
curl -u username:app_password  
https://yourblog.com/wp-json/wp/v2/users/me
```

#### Issue: "Articles created but validation scores always low"

**Cause:** Keyword selection criteria too loose

**Solution:**

60. Raise minimum validation score (70 → 90)

61. Add competition check (allintitle must be <5000)

62. Require Reddit validation (10+ monthly posts)

63. Focus on more specific long-tail keywords

#### Issue: "System creates articles but they're low quality"

**Cause:** Content creation prompt too rushed or model limitations

**Solution:**

64. Switch to better model (GPT-4 Turbo or Claude Sonnet 4.5)

65. Increase max tokens for content creation (20K → 50K)

66. Add more competitor analysis depth

67. Extend content creation prompt with examples

68. Add human review checkpoint before publishing

#### Issue: "Internal links not resolving properly"

**Cause:** Database not populated or matching logic too strict

**Solution:**

69. Manually populate internal-links/database.json with existing articles

70. Broaden matching (check for topic overlap, not exact matches)

71. Skip internal linking temporarily until 10+ articles published

#### Issue: "Trend discovery finds nothing relevant"

**Cause:** Niches too specific or wrong subreddits

**Solution:**

72. Expand niche definitions (be more general)

73. Add more subreddits (r/Entrepreneur, r/SideProject, r/IMadeThis)

74. Try different search queries in Brave

75. Lower trend\_score threshold (70 → 60)

# Advanced Strategies

## Strategy 1: Topic Clusters

Once you have 10+ articles published, create topic clusters:

**Goal:** Internal linking strategy where multiple articles link to a main "pillar" article

**How to implement:**

76. Identify your main topics (e.g., "AI Agents")

77. Create one comprehensive pillar article:

- "The Complete Guide to AI Agents (2025)"
- 3000-4000 words
- Covers everything broadly

78. Create 5-10 supporting articles:

- "How to Build AI Agents for Email"
- "Best AI Agent Tools Compared"
- "AI Agents vs Chatbots: Key Differences"
- Each 1500-2000 words
- Each focused on specific subtopic

79. Link structure:

- All supporting articles link to pillar
- Pillar links to all supporting articles
- Supporting articles link to each other where relevant

**Result:** Google sees your site as an authority on this topic

## Strategy 2: Competitive Displacement

Target articles from competitors that are outdated:

**Process:**

80. Find ranking articles from 2022-2023

81. Check if information is outdated

82. Create updated version with:

- "Updated for 2025" in title
- All new data and statistics
- New tools/options released since original
- Better examples
- Improved structure

**Prompt Addition for Trend Discovery:**

When analyzing Brave Search results:

- Note publication dates of top 10 articles
- Flag any from 2023 or earlier as "update opportunities"

- Prioritize these for keyword validation
- Mark content strategy as "updated\_2025"

## Strategy 3: FAQ Snippet Optimization

Target Google's featured snippets by optimizing FAQ sections:

### Enhanced FAQ Prompt Addition for Content Creation:

For the FAQ section:

1. Research actual questions:
  - Check Reddit for exact phrasing
  - Search "people also ask" on Google
  - Use exact questions people type
2. Format answers for featured snippets:
  - Start with direct answer (40-60 words)
  - Then provide additional context
  - Use simple language
  - Include relevant stat if available
3. Structure for schema:
  - Minimum 5 FAQs
  - Maximum 10 FAQs
  - Each answer 40-300 words
  - Cover question intent fully

## Strategy 4: Multi-Format Content

Repurpose your articles into different formats:

### After publishing, create:

83. **Twitter Thread:** Break down main points into 8-10 tweets
84. **LinkedIn Carousel:** Convert to slide deck
85. **YouTube Script:** Turn into video script
86. **Email Newsletter:** Summarize with CTA
87. **Podcast Notes:** Outline for audio version

### Automation Idea:

Add Phase 5 prompt for content repurposing that automatically creates these formats.

## Strategy 5: Backlink Outreach Automation

Build backlinks by finding link opportunities:

### Additional Prompt (run weekly):

BACKLINK OPPORTUNITY FINDER

For each published article in last 30 days:

1. Search Brave for:
  - "[main topic] statistics"

```

"[main topic] resources"
"[main topic] guide"

Filter: Results from last 6 months

2. For each result:
- Fetch the page
- Check if they link out to other resources
- Identify if your article would fit
- Find contact email

3. Generate outreach email:
- Personalized to their article
- Mention specific section where you'd fit
- Explain unique value of your article
- Keep under 100 words

4. Save to /seo-automation/outreach/opportunities.json

```

Every week, you'll have 10-20 personalized outreach emails ready to send.

## Strategy 6: Performance-Based Optimization

Automatically update articles that aren't ranking:

### Monitoring Prompt (run every 30 days):

```

ARTICLE PERFORMANCE OPTIMIZER

For each article published 30+ days ago:

1. Check ranking position (if GSC connected)
2. If not on page 1-3:

    A. Re-run competitor analysis
        - Have competitors updated?
        - Are new competitors ranking?
        - What changed?

    B. Identify improvements:
        - Add missing topics
        - Update statistics
        - Improve examples
        - Increase word count
        - Add more internal links

    C. Create update plan:
        - List specific changes needed
        - Estimate effort
        - Priority (high/medium/low)

    D. Save to /seo-automation/updates/pending.json

3. For high-priority updates:
    - Generate updated sections

```

- Create tracked changes version
- Schedule for publishing

## Final Checklist: Complete System Setup

### Phase 1: Initial Configuration ✓

- Created folder structure
- Configured API keys
- Set niches and subreddits
- Tested all API connections
- Verified WordPress access

### Phase 2: Prompt Setup ✓

- Trend Discovery prompt in OpenClaw
- Keyword Research prompt in OpenClaw
- Content Creation prompt in OpenClaw
- Publishing prompt in OpenClaw
- Master Orchestrator prompt in OpenClaw

### Phase 3: Testing ✓

- Ran manual trend discovery (got results)
- Ran manual keyword validation (got recommendations)
- Created test article (quality check passed)
- Published test article (went live successfully)

### Phase 4: Automation ✓

- Enabled continuous mode on Master Orchestrator
- Set appropriate run interval (1 hour)
- Configured daily budget (\$10)
- Set daily article quota (2)
- Enabled error notifications

### Phase 5: Monitoring ✓

- Reviewed first 24 hours of operation
- Checked all tracking files being created
- Verified costs are reasonable
- Confirmed articles are being published
- Set weekly review reminder

# Success Metrics: What to Track

## Week 1 Goals:

- 5-7 articles published
- All validation scores 90+
- System uptime 95%+
- Cost under \$35 for the week
- Zero critical errors

## Month 1 Goals:

- 25-30 articles published
- 3-5 articles ranking on page 2-3
- Average validation score 110+
- System fully autonomous
- Cost under \$150 for the month

## Month 3 Goals:

- 75-90 articles published
- 10-15 articles on page 1
- Organic traffic growing 50%+
- Topic clusters established
- ROI positive (traffic value > costs)

# Support & Community

## Getting Help

### If you're stuck:

88. Check the troubleshooting section
89. Review your error logs
90. Test each phase individually
91. Verify API credentials

### Common beginner mistakes:

- Not setting page/content width correctly in WordPress
- Forgetting to enable continuous mode
- Setting article quota too high (start with 1-2)
- Not monitoring first 48 hours closely
- Skipping API connection tests

## Sharing Your Results

Track your results and share what's working:

- Articles published per week
- Validation score trends
- Cost per article
- Ranking improvements
- Traffic growth

The SEO automation community thrives on shared data!

## Conclusion

You now have a complete, autonomous SEO content system that:

- ✓ Discovers trending topics 24/7
- ✓ Validates keywords with real data
- ✓ Writes SEO-optimized articles
- ✓ Publishes and promotes automatically
- ✓ Costs \$2-5 per day
- ✓ Scales across unlimited niches

### Your next steps:

92. Complete the initial setup checklist
93. Run your first manual cycle (all 4 phases)
94. Enable the Master Orchestrator
95. Monitor for 48 hours
96. Adjust and optimize
97. Scale to more niches

### Remember:

- Quality over quantity (start with 1-2 articles/day)
- Monitor the first week closely
- Let the system learn and optimize
- Trust the process (rankings take 7-30 days)
- Track your metrics

The future of SEO is autonomous, and you're now equipped to dominate it.

Happy automating! 