Hierarchical Article Generation Algorithm

Overview

The proposed algorithm follows a structured approach to generate articles with variable hierarchical levels. The system utilizes three primary integer parameters to define the article structure and conceptualizes content as a tree structure with variable depth based on user specifications.

Core Parameters

The algorithm operates using three essential integer parameters:

1. **Sections Parameter** (integer): Specifies the number of main sections
2. **Subsections Parameter** (integer): Defines the number of subsections per section
3. **Paragraphs Parameter** (integer): Determines the number of paragraphs per subsection

Hierarchical Structure Logic

Three-Level Hierarchy

When `subsections parameter ≠ 0`, the system generates a three-level tree structure:

- **Level 1**: Main sections (root nodes)
- **Level 2**: Subsections (child nodes)
- **Level 3**: Paragraph points (leaf nodes)

Two-Level Hierarchy

When `subsections parameter = 0`, the system generates a simplified two-level structure:

- **Level 1**: Main sections
- **Level 2**: Paragraph points (directly under sections)

Example Implementation

For parameters `(2 sections, 3 subsections, 4 paragraphs)`:

```
Article Root
├── Section 1
│   ├── Subsection 1.1
│   │   ├── Paragraph Point 1
│   │   ├── Paragraph Point 2
│   │   ├── Paragraph Point 3
│   │   └── Paragraph Point 4
│   ├── Subsection 1.2
│   │   ├── Paragraph Point 1
│   │   ├── Paragraph Point 2
│   │   ├── Paragraph Point 3
│   │   └── Paragraph Point 4
│   └── Subsection 1.3
│       ├── Paragraph Point 1
│       ├── Paragraph Point 2
│       ├── Paragraph Point 3
│       └── Paragraph Point 4
└── Section 2
    ├── Subsection 2.1
    │   ├── Paragraph Point 1
    │   ├── Paragraph Point 2
    │   ├── Paragraph Point 3
    │   └── Paragraph Point 4
    ├── Subsection 2.2
    │   ├── Paragraph Point 1
    │   ├── Paragraph Point 2
    │   ├── Paragraph Point 3
    │   └── Paragraph Point 4
    └── Subsection 2.3
        ├── Paragraph Point 1
        ├── Paragraph Point 2
        ├── Paragraph Point 3
        └── Paragraph Point 4
```

Flexible Paragraph Generation

To enhance article naturalness, the system incorporates a **variable paragraph feature**. When "flexible paragraphs" mode is enabled:

- The language model gains autonomy to determine optimal paragraph counts
- Fixed numerical constraints are replaced with adaptive content generation
- Each section or subsection can have varying numbers of paragraphs based on content requirements
- The end nodes (paragraph points) can be dynamically generated by the LLM

Implementation Requirements

The algorithm requires modifications to four core components:

1. Outline Generation Module

- Must construct and store the hierarchical tree structure based on specified parameters
- Should handle both fixed and flexible paragraph modes
- Must maintain structural integrity across different hierarchy levels

2. Paragraph Generation Module

- Must generate content according to section hierarchy and subsection titles
- Should incorporate paragraph indexing for proper content organization
- Must adapt to variable paragraph counts when flexible mode is enabled

3. Grammar Processing Module

- Must traverse the tree structure while maintaining awareness of hierarchical depth

- Should understand node relationships and structural dependencies
- Must apply appropriate grammar rules based on content level and context

4. Humanization and Formatting Modules

- Must account for the tree structure during content processing
- Should adapt behavior based on presence or absence of subsections
- Must maintain consistent formatting across different hierarchical complexities
- Should ensure natural flow between sections, subsections, and paragraphs

Benefits

This systematic approach ensures:

- **Scalable article generation** across different content lengths and complexities
- **Structural coherence** throughout all hierarchical levels
- **Natural content flow** that adapts to topic requirements
- **Flexible content organization** suitable for various article types
- **Consistent quality** across all generated content sections

Conclusion

The hierarchical article generation algorithm provides a robust framework for creating well-structured, naturally flowing articles with variable complexity levels. By implementing tree-based logic with flexible parameter handling, the system can generate content that meets diverse requirements while maintaining professional quality and readability.