Design for Operationalization & Improvement

This section outlines a strategic plan to enhance the topic extraction system for production use, focusing on improving topic quality, and ensuring reliability, performance, and scalability through AI and automation.

1. Improving Topic Extraction Using Topic Modeling

1.1 Gensim + LDA / NMF

- **Gensim** is a Python library optimized for NLP and topic modeling.
- **LDA (Latent Dirichlet Allocation)** and **NMF (Non-negative Matrix Factorization)** are unsupervised algorithms to discover hidden topics in text.

How they help:

1. **Aggregate across multiple pages**: Instead of treating pages independently, you can model topics for an entire website, blog, or product category.
2. **Capture hidden topics**: While your current system uses NER, noun chunks, and keywords, LDA/NMF can reveal recurring latent topics that may not be explicit.
3. **Improve topic relevance**: Using TF-IDF or CountVectorizer as input, LDA/NMF ranks topics by probability distribution, giving a more global understanding.

Workflow:

1. **Collect texts**: Aggregate title, headings, and body content from multiple pages.
2. **Preprocess**: Tokenization, lemmatization, stopwords removal (your current StopwordManager can be reused).
3. **Vectorize**: Use CountVectorizer (for LDA) or TfidfVectorizer (for NMF).
4. **Apply model**: Fit LDA/NMF with n_topics (configurable) to discover recurring themes.
5. **Extract keywords per topic**: Rank words by weight within topics to define high-level topics.
6. **Combine with current priority scoring**: You can assign **weights** based on the LDA/NMF probability and your existing hierarchy (title>heading>body).

Benefits:

- Reduces noise from single-page anomalies.
- Captures multi-page thematic trends.
- Can automatically suggest new topics or clusters.

2. Implementing Multi-Page Extraction with Scrapy

Scrapy is ideal for multi-page websites. Here's how it fits:

2.1 Spider Design

- **Master spider**: Crawl multiple pages (category → sub-pages → articles/products).
- **Parse method**: Extract content using ContentFetcher logic.
- **Pipelines**: Modular processing for:
    1. Text cleaning & filtering (stopwords, URL keywords)
    2. Topic extraction (current NLP or LDA/NMF)
    3. Database insertion / caching

Scrapy advantage: Can follow pagination automatically, obey robots.txt, and scale to hundreds of pages.

2.2 Pipelines

- **Pipeline-based modularity** helps separate concerns:

Item -> CleanContentPipeline -> StopwordPipeline -> NLPTopicPipeline -> LDA/NMFPipeline -> DatabasePipeline

- Each pipeline can handle:
    - Text extraction and normalization
    - Stopword removal
    - Entity/Noun Chunk extraction
    - Topic modeling (optional)
    - Database or JSON storage
- Pipelines can also **store intermediate results**, e.g., store TF-IDF vectors for later batch modeling.

---

3. Handling Anti-Scraping / Scaling

Large websites often block bots. Here's how to handle it:

3.1 Headers and User Agents

- Rotate User-Agent headers to mimic different browsers.
- Set Accept-Language and Referer headers to appear human-like.

3.2 Proxies

- **Residential IPs**: Appear as normal users, low chance of being blocked.
- **Datacenter IPs**: Faster, cheaper, but more likely to be blocked.
- **Rotation strategy**:
    - Assign new proxy every request.
    - Rotate headers along with IPs.
    - Limit requests per IP per minute.

Scrapy integration:

- Use scrapy-rotating-proxies or custom middleware.
- Use DOWNLOADER_MIDDLEWARES for dynamic IP and headers assignment.

---

4. Benefits of This Approach

Feature	Benefit
Multi-page scraping	Capture full website topics, not just a single page
LDA / NMF topic modeling	Discover hidden recurring topics across pages
Scrapy pipelines	Modular, maintainable, scalable scraping and processing
Proxy rotation	Avoid blocks, increase reliability of scraping
Header rotation	Mimics real browsers to avoid detection

✅ **Summary**:

- **Single page**: Use your existing priority-based NLP scoring.
- **Multi-page website**: Use Scrapy, pipelines, and LDA/NMF for topic modeling.
- **Proxies & headers**: Make scraping scalable and robust.
- **Combined approach**: Merge per-page NLP scores with global topic modeling for high-quality, reliable results.