ChatGPT

퀄리뉴스 기사 수집·발행 파이프라인 분석 및 개선 제안

1. 파일 비교

1.1 복구된 파일 `1jjippa_news_scheduler.py` (파일 1)

- **목적** – 커뮤니티 수집 기능과 신규 공식 소스(smt/supply) 수집을 간단히 통합한 모듈로, `--collect` / `--publish` 플래그를 통해 공식 소스에서 첫 번째 기사 링크를 가져와 제목·요약을 추출하고 커뮤니티와 함께 발행한다.
- **기능 요약**
- **공식 기사 수집** – `collect_official` 함수는 `smt_sources.json`과 `supply_sources.json`에서 소스명을 읽어 첫 기사 URL을 찾아 제목과 요약을 가져온 후 기본 정보를 기록한다.
- **커뮤니티 수집** – 기존 커뮤니티 수집 로직(`_reddit_old_new`, `_parse_forum_html`)을 유지하며, 키워드 기반 점수와 업보트/댓글 수를 사용해 후보 기사를 선정한다.
- **선정·발행** – 승인된 커뮤니티와 공식 기사를 섹션별(`표준 뉴스→글로벌 전자생산→AI 뉴스→커뮤니티`)로 합쳐 `HTML/MD/JSON` 형식으로 발행한다. 편집자 승인 UI도 포함된다.
- **한계** – 원래 시스템에 있었던 품질 게이트(QG), 팩트체크(FC), 기사 가치 점수화, 한글 부제 및 요약, AI 편집장 한마디 등 주요 기능이 대부분 제거됐다. 공식 소스 기사도 단지 첫 기사 링크와 제목/description만 기록하며 본문 길이 검증이나 신뢰도 평가가 없다.

1.2 원본 파일 `2jjippa_news_scheduler.py` (파일 2)

- **목적** – 공식 뉴스와 커뮤니티를 분리 운영하는 완전한 수집·평가·발행 파이프라인이다.
- **주요 기능**
- **품질 게이트(QG)** – 기사 본문 길이, 문장 수, 링크 수 등을 검사해 기준 이상인 기사만 PASS/HOLD로 통과시킨다. 불충분한 기사에 대해서는 REJECT 처리하며 도메인 신뢰도에 따라 점수를 가중한다.
- **팩트체크(FC)** – QG를 통과한 기사에 대해 숫자, 연도, 표준 코드, 외부 링크, PDF 등 근거 요소를 찾아 점수를 계산한다. 근거 종류가 2가지 이상이고 점수가 50점 이상이면 PASS, 그렇지 않으면 FAIL/HOLD가 된다.
- **기사 가치 평가** – 도메인 신뢰도, 키워드 포함 여부, 기사 최신도 등으로 스코어링을 수행하고 점수를 부여한다. 커뮤니티 게시물은 추천수·댓글수·조회수를 사용해 점수를 계산한다.
- **한글 부제 및 요약** – 영문 기사의 제목을 규칙 기반으로 한글 부제로 변환하고, 본문에서 첫 2~3문장을 추출한 간이 요약을 생성한다. 번역/요약은 비용을 아끼기 위해 수집 단계에서는 규칙 기반으로, 발행 단계에서만 OpenAI API를 호출한다.
- **AI 편집장 한마디** – 편집자가 한 줄 논평을 작성할 수 있도록 지원하며, editor_rules.json에서 기본 문구 및 프롬프트를 정의한다.
- **세분화된 설정과 로깅** – `config.json`에서 QG/FC 기준, 섹션 순서, 로그 파일, 백업 경로 등 세부 설정을 관리하고 `editor_rules.json`에서 도메인·키워드 가중치를 정의한다.
- **한계** – 커뮤니티 수집기의 구현이 미완성 상태였으며, Reddit/포럼 모듈이 제대로 작동하지 않아 커뮤니티가 '0건'으로 나오는 문제가 있었다. 또한 코드가 모놀리식 형태여서 유지보수가 어렵고 테스트가 힘들다.

1.3 차이점 요약

항목	파일 1 (복구본)	파일 2 (원본)
수집 대상	공식 소스(smt/supply)와 커뮤니티를 단순 통합	SMT/AI/표준/군사/우주 등 다양한 공식 소스와 커뮤니티를 구분 수집

항목	파일 1 (복구본)	파일 2 (원본)
품질 게이트 (QG)	미포함	본문 길이, 문장, 링크 등으로 PASS/HOLD/REJECT
팩트체크(FC)	없음	근거 요소 점수를 계산해 기사 신뢰도 판별
기사 가치 평가	커뮤니티에서만 간단한 점수	도메인·키워드·신규성 등 종합 스코어링
한글 부제/요약	기사 description만 저장	한글 부제와 간단 요약 생성, 번역 API 연동 준비
편집장 한마디	커뮤니티 승인 UI만 존재	editor_rules 기반으로 기본 코멘트 및 프롬프트 지원
설정 및 로그 관리	DEFAULT_CFG에 간단 설정만 포함	config.json/editor_rules.json에서 상세 설정과 백업·로그 관리
모듈 구조	단일 스크립트	품질/팩트/번역 등 기능이 한 파일에 밀집, 복잡함

파일 1은 커뮤니티 수집기 문제를 임시로 회피하기 위해 커뮤니티와 새 공식 소스의 기본적인 합병 기능만 남긴 버전이다. 파일 2는 원래 목표였던 “전 세계 표준·전자제조·AI 정보를 매일 수집→번역·요약→AI 편집장 한마디 제공”을 달성하기 위한 전체 기능을 포함하지만, 커뮤니티 미지원과 모놀리식 구조로 인해 확장성이 떨어진다.

2. 문서 분석을 통한 요구사항 정리

2.1 커버리지 확장과 키워드 관리

- **SMT/AI 소스 확장** – 가이드에서는 SMT Today, Global SMT & Packaging, SMTA 뉴스, JEDEC/IEC/IEEE Packaging Society 등 산업 협회 및 전문 매체를 신규 소스로 추가하라고 권고한다. AI 분야는 NVIDIA·MIT Tech Review·DeepMind·Anthropic 등 연구소 블로그와 AI+제조 융합 매체를 포함하여 빅테크 위주 편향을 해소해야 한다.
- **키워드 기반 수집 강화** – SMT 기술, 표준, 장비, 소재 등 한글과 영문의 세부 키워드를 리스트화하여 Reddit/포럼 검색어 및 키워드 점수에 활용해야 한다. 키워드는 유행어(Industry 4.0, CFX 등)를 포함하도록 주기적으로 갱신한다.
- **커뮤니티 포럼 개선** – community_sources.json의 Reddit/포럼 항목을 점검하고, 최소 추천/댓글 조건과 score_threshold를 낮춰 커뮤니티 글 0건 문제를 해결해야 한다. Reddit API를 사용하는 경우 OAuth 인증과 rate limit 백오프가 필요하며, 포럼은 RSS 우선·HTML 파싱 보조 방식으로 구축해야 한다【1002_2퀄리뉴스의 기술적목표.txt † L5-L16】.

2.2 품질·팩트·가치 평가 고도화

- **QG 기준 조정** – QG 기준(본문 글자 수, 문장 수, 링크 수)을 너무 엄격하거나 느슨하지 않게 조정한다. 유용한 기사인데 550자라서 탈락한다면 `min_chars`를 500으로 낮추거나 strict 모드를 false로 바꿔 HOLD 처리할 수 있다. 표준 문서처럼 짧은 기사에는 섹션별로 다른 기준을 적용한다【1003_퀄리목표 (1).txt † L1-L30】.
- **도메인·키워드 가중치 스코어링** – editor_rules.json에서 신뢰도 높은 도메인(IPC/NASA/ECSS 등)에 기본 가중치를 부여하고, 키워드별 점수를 설정하여 기사 가치 점수를 계산한다. 여러 매체에서 동일 주제가 등장하면 이슈 중요도 가중치를 추가한다. 커뮤니티 글은 추천수·댓글수·조회수에 더해 본문 길이·링크 포함 여부로 점수를 보정한다【1002_2퀄리뉴스의 기술적목표.txt † L22-L34】.
- **팩트체크 개선** – 단순 근거 개수뿐 아니라 출처의 신뢰도(도메인 분석), 숫자/통계의 맥락, PDF 링크 등 질적인 평가를 도입하고, AI 팩트체크 활용 가능성을 연구한다.

2.3 번역·요약 및 AI 편집장

- **2단계 요약** – 추출적 요약(중요 문장 추출) 후 GPT-계열 모델로 추상적 요약을 생성하여 정확성과 가독성을 모두 확보한다【1002_2퀄리뉴스의 기술적목표.txt†L39-L47】. 번역·요약에는 사내 용어집(Glossary)을 적용해 용어 일관성을 유지하고 여러 번역 API(DeepL, Google, OpenAI)를 지원한다【1002_2퀄리뉴스의 기술적목표.txt†L39-L47】.
- **AI 편집장 한마디** – 기사 요약과 메타정보를 기반으로 GPT 프롬프트를 구체화하고, length/tonality 규칙을 editor_rules.json에 명시하여 일관된 코멘트를 생성한다【1002_2퀄리뉴스의 기술적목표.txt†L39-L47】.
- **발행 시점 번역** – 수집 단계에서는 영어 제목을 간단한 규칙으로 한글 부제로 변환하고, 발행 단계에서만 번역 API를 호출해 비용을 절감한다.

2.4 운영·UI·스케줄링 개선

- **모듈화와 스케줄러** – 수집(collect_news.py), 커뮤니티 수집(collect_community.py), 평가(evaluate.py), 요약/번역(translate_summarize.py), 발행(publish_engine.py) 등으로 기능을 분리하여 유지보수성을 높인다【1003_퀄리목표 (1).txt†L31-L46】. cron 또는 task scheduler로 하루 한 번 자동 수집·발행을 수행하고 백업/로그를 체계적으로 관리한다.
- **편집자 UI 개선** – 승인 UI에 품질/팩트 상태 배지와 총점 표시, 검색·필터 기능, 한국어 요약 미리보기, 편집장 코멘트 placeholder 등을 추가해 편집 효율을 높인다. [저장]과 [발행] 버튼을 분리하여 실수로 발행되는 것을 방지한다.
- **신규 섹션과 태그** – SMT 전용 섹션 신설을 검토하고, `config.json`에서 source→section 매핑을 관리해 유연하게 조정한다.
- **중장기 로드맵** – 향후 AI 에디터 어시스턴트 도입, React/FastAPI 기반의 멀티 편집자 웹앱, 독자 피드백 데이터 분석 등을 검토할 수 있다.

3. 개발 최적 전략과 제안

종합하면, 퀄리뉴스의 목표는 “전 세계 표준·전자제조·AI 정보를 매일 자동 수집하고, 한국어 요약과 편집자 코멘트를 붙여 안정적으로 발행하는 것”이다【1003_퀄리목표 (1).txt†L1-L30】. 이를 위해 원본(파일 2)의 고도화된 기능을 유지하면서 복구본(파일 1)의 커뮤니티/공식 통합 아이디어를 발전시켜야 한다. 다음과 같은 전략을 제안한다.

3.1 파일 구조 리팩토링 및 핵심 기능 복원

1. **모듈 분리** – 파일 2에 포함된 QG, FC, 번역, 스코어링 등을 별도 모듈로 분리한다. 예를 들어 `collect_news.py`는 feeds/official_sources.json을 읽어 여러 기사 링크를 수집하고, `_extract_main_text_from_html`과 QG/FC를 적용한 뒤 후보 JSON을 만든다. `collect_community.py`는 Reddit API 및 포럼 크롤러를 구현한다. `evaluate.py`는 editor_rules.json을 로드해 스코어를 계산하고 섹션을 분류한다. `translate_summarize.py`는 2단계 요약·번역을 수행한다. 마지막으로 `publish_engine.py`는 승인된 기사만 섹션별로 모아 HTML/MD/JSON을 생성한다【1003_퀄리목표 (1).txt†L31-L46】.

2. **커뮤니티 수집기 구현** – 커뮤니티가 0건으로 나오는 문제를 해결하기 위해 Reddit API 인증과 rate limit 처리, 포럼 RSS 우선·HTML 파싱 보조 전략을 도입한다. community_sources.json에 최소 추천·댓글 수, 허용 도메인, 차단 패턴을 설정하고 필터를 적절히 완화한다. 본문 길이와 링크 여부로 추가 필터링을 적용하여 광고성 글을 제거한다【1002_2퀄리뉴스의 기술적목표.txt†L22-L34】.

3. **공식 소스 확장** – feeds/official_sources.json/smt_sources.json/supply_sources.json에 SMTA, JEDEC, IEC, IEEE Packaging Society, NVIDIA, MIT Tech Review, DeepMind 등 신뢰도 높은 소스를 추가하고 RSS를 활용한다. config.json의 `landing_rules`에 각 사이트의 우선/차단 패턴을 정의해 목록 페이지나 광고 페이지로 잘못 이동하지 않도록 한다【1002_2퀄리뉴스의 기술적목표.txt†L5-L16】.

4. **품질·팩트·가치 평가 복원 및 개선** – 파일 1에는 없는 QG/FC/스코어링을 파일 2에서 추출해 모듈화하고, 섹션별 기준을 조정한다. 기사 본문 길이, 문장 수, 링크 수 등 QG 기준을 `editor_rules.json`과 `config.json`에서 조절 가능하게 하고, 표준 뉴스는 길이 기준을 완화한다. FC에서는 신뢰도 높은 도메인 가중치와 근거 질적 평가를 도입하고, AI 팩트체크를 점진적으로 실험한다.

5. **도메인·키워드 기반 스코어링** – editor_rules.json을 새롭게 작성하여 신뢰도 높은 도메인에 기본 점수를, SMT/AI 핵심 키워드에 가중치를 부여한다. 동일 주제가 여러 매체에서 등장하면 이슈 중요도 가중치를 더해 트렌드 뉴스를 빠르게 포착한다. 커뮤니티 글에는 추천·댓글·조회수 외에 본문 길이·링크 여부에 따른 가중치를 설정한다【1002_2퀄리뉴스의 기술적목표.txt † L22-L34】.

6. **한국어 요약 및 번역 모듈** – 영문 기사에 대해 추출적 요약 후 GPT-4 o 등 모델을 사용해 한국어 요약을 생성한다. 번역·요약 시 glossary를 적용해 용어 일관성을 유지하고, OpenAI API가 없을 때는 규칙 기반 요약으로 폴백한다【1002_2퀄리뉴스의 기술적목표.txt † L39-L47】. 발행 시에만 전체 본문 번역을 호출해 비용을 절감한다.

7. **AI 편집장 한마디** – editor_rules.json에 코멘트 프롬프트와 스타일 가이드(길이·객관성 등)를 정의하고, 기사 요약과 메타정보를 넣어 GPT를 호출하여 초안을 생성한다. 편집자가 이를 검토·수정하여 최종 논평으로 사용한다.

3.2 개발 일정 및 우선순위

우선순위는 가이드에서 제안한 3단계 접근을 따른다:

1. **단기 (1–2주)** – 소스 추가 및 커뮤니티 수집기 구현. smt/official/community 소스 목록을 확장하고 키워드 리스트를 보완하여 더 많은 기사를 확보한다. config의 QG/FC 기준을 현행 통계에 맞게 조정한다.
2. **중기 (2–3주)** – editor_rules 기반 스코어링 시스템 구축과 UI/UX 개선. domain/keyword 가중치를 적용한 total_score를 계산하고 승인 UI에 표시한다. 커뮤니티 필터를 튜닝해 0건 문제를 완전히 해소한다.
3. **후기 (3–4주)** – 팩트체크 고도화와 한국어 요약·AI 편집장 기능 강화. 근거 질적 분석과 AI 팩트체크 시험, 2단계 요약 구현, 번역 API 지원 확대를 진행한다.

3.3 최종 목표와 성공 기준

- **일일 후보 확보량** – 공식+커뮤니티에서 매일 80–180건의 후보가 모이고, QG PASS ≥ 60%, FC PASS ≥ 70%를 달성한다【1003_퀄리목표 (1).txt † L1-L30】.
- **100% 한국어 요약** – 승인된 모든 기사에 한국어 요약이 제공되고 용어 일관성이 유지된다【1003_퀄리목표 (1).txt † L1-L30】.
- **발행 안정성** – 크론/스케줄러에 의해 매일 13:15(KST) 자동 발행이 4주 동안 단 한 번도 실패하지 않는다【1003_퀄리목표 (1).txt † L1-L30】.
- **커뮤니티 0건 문제 해결** – selected_community.json이 0건으로 남는 일이 재발하지 않는다【1003_퀄리목표 (1).txt † L1-L30】.
- **모듈화 및 운영 문서화** – 코드가 collect/ evaluate/ summarize/ publish 등으로 분리되고, 운영자가 매일 3분 이내에 수집·승인·번역·발행 과정을 수행할 수 있는 매뉴얼을 제공한다【1003_퀄리목표 (1).txt † L31-L46】.

4. 결론

복구본(파일 1)은 커뮤니티와 새로운 SMT 소스의 간단한 통합본일 뿐이며, 원본(파일 2)에 있던 품질·팩트·평가·번역 기능이 대부분 삭제되어 있어 퀄리뉴스의 비전과 목표를 달성할 수 없다. 반대로 원본은 기능이 풍부하지만 커뮤니티 수집 미완성과 모놀리식 구조로 유지보수가 어렵다.

따라서 최선의 선택은 파일 2를 기반으로 하되, 복구본에서의 통합 아이디어를 참고하여 전체 파이프라인을 모듈화하고, 위 문서들이 제시한 개선사항을 반영하여 확장하는 것이다. 이를 통해 “전 세계 표준·전자제조·AI 정보를 매일 자동 수집하고 한국어 요약과 편집자 코멘트를 붙여 안정적으로 발행한다”는 목표를 실현할 수 있다.
