

**SK네트웍스 Family AI과정 12기  
 모델링 및 평가 테스트 계획 및 결과 보고서**



| **산출물 단계** | 모델링 및 평가 |
| --- | --- |
| **평가 산출물** | 테스트 계획 및 결과 보고서 |
| **제출 일자** | 2025.08.08 |
| **깃허브 경로** | <https://github.com/SKNETWORKS-FAMILY-AICAMP/SKN12-FINAL-3TEAM> |
| **작성 팀원** | 이준석 |

1. **1차 테스트 개요**

* 기간: 2025.08.07-2025.08.08
* 유형: 1차 모듈 통합 테스트
* 목표: 독립적으로 개발된 AI 엔진, 백엔드 서버, 프론트엔드, 외부 서비스 연동 모듈들의 첫 통합 후 개별 동작 검증 및 모듈 간 인터페이스 정합성을 확인합니다.

1. **테스트 목적**

**AI 파이프라인 정확성 검증**

- WhisperX 화자 분리 및 음성 인식 정확도

- 전처리 데이터 구조화 및 컨텍스트 처리

- BERT 노이즈 필터링 적절성

- LLM 업무 생성 품질

**시스템 통합 안정성 검증**

- 프론트엔드-백엔드 데이터 동기화

- 사용자 인터페이스

- 외부 서비스 실시간 연동

1. **테스트 환경**

3.1 하드웨어 환경

* CPU: Intel 13th i5-1340p
* RAM: 32GB
* 디스크: 500GB SSD

3.2 소프트웨어 환경

* 운영 체제: windows 10 64-bit
* 데이터베이스: PostgreSQL 17.5 on x86\_64-windows (Prisma ORM 사용)
* 웹 서버: Express 4.18
* 애플리케이션 서버: Node.js v22.16.0
* 프로그래밍 언어: TypeScript 5.8 / Python 3.12.9 / Javascript
* 테스팅 도구: Jest (백엔드), Vitest + React Testing Library (프론트엔드), Jest/Supertest (API 테스트)
* 외부 연동: Slack API, Notion API, Jira API
* 실시간 통신: [Socket.i](http://socket.io)o 4.7.2

1. **테스트 케이스**

4.1 WhisperX

| **테스트 케이스 ID** | **테스트 항목** | **테스트 설명** | **예상 결과** | **성공**  **여부** | **비고** |
| --- | --- | --- | --- | --- | --- |
| 1-1 | 화자 구분 | 화자 구분이 정확히 되었는지 | 화자 구분에 예외 없음 | 실패 | 화자 구분에 예외 있음 |
| 1-2 | 데이터 규격 | 데이터가 예시대로 제작되었는지 | 규격대로 반환 | 성공 |  |
| 1-3 | 파일 업로드 | 비어있는 데이터는 없는지 | 파일 업로드 완료 메시지 출력 | 성공 |  |
| 1-4 | 타임스탬프 정렬 | 타임스탬프 정렬 정확성 | 모든 발화가 시간 순으로 정렬 | 성공 |  |
| 1-5 | 인접 발화 병합 | 동일 화자의 인접 발화 병합 여부 | 연속된 같은 화자의 문장이 자연스럽게 병합 | 성공 |  |

단계별 진행 및 예상 결과

1. 녹음된 음성 파일이 WhisperX를 거치며 텍스트로 결과가 출력
2. 형태는 {full text, segments{start, end, text, words {word, start, end, score}, language, duration, character}로 출력 (실제로 STT를 통해 생성되는 부분은 words 부분이며, 나머지는 디버깅용 확인 부분)
3. 가공된 데이터가 {word, start, end, score} 형태인지 확인
4. 실패 부분:

1-1 (화자 구분 유무)

* 예상 결과: 화자 구분이 예외 없이 잘 처리됨
* 실제 결과: 화자 구분에 예외 있음, 말한 것과 사람 간의 매칭이 일치하지 않음

→ 위스퍼 모델 자체의 한계인 관계로 근본적 해결 불가 → 사용자가 직접 컨펌 후 업로드하도록 하여 오류내용을 직접 확인하도록 유도 예정

4.2 전처리

| **테스트 케이스 ID** | **테스트 항목** | **테스트 설명** | **성능 기준** | **예상 결과** | **성공**  **여부** | **비고** |
| --- | --- | --- | --- | --- | --- | --- |
| 2-1 | 컬럼 생성 여부 | 예시대로 컬럼이 설정되었는지 | {"prev", "target", "next", "labels"} 컬럼 생성됨 | 컬럼 생성에 문제 없음 | 성공 |  |
| 2-2 | 타겟 문장 분할 여부 | 타겟 문장이 한 문장으로 처리되어있는지 | 모든 문장이 한 문장으로 예외없이 정제됨 | 모든 문장이 한 문장으로 정제됨 | 성공 |  |
| 2-3 | 문장 구조 확인 | prev, next 문장이 두 문장씩 포함되었는지 | 전부 두 문장으로 구성됨 | 문장 구성에 문제 없음 | 성공 |  |
| 2-4 | Labels 컬럼 공백 여부 | labels 컬럼이 비어있는 채로 생성되었는지 | 예외 없이 빈칸으로 반환됨 | labels 컬럼이 모두 빈칸임 | 성공 |  |
| 2-5 | 비정상 문자/기호 제거 여부 | 비정상 문자/기호 제거 여부 | 특수 문자, 이모지 등 불필요한 요소 제거 | 특수문자, 이모지 정상적으로 제거됨 | 성공 |  |
| 2-6 | 발화자 태깅의 일관성 | 발화자 태깅의 일관성 | 동일 인물에게 일관된 이름이 부여됨 | 발화자에 맞게 태깅됨 | 실패 | 발화자에 맞게 태깅되지 않음 |

단계별 진행 및 예상 결과:

1. STT로 추출한 텍스트 데이터를 bert 모델에 넣기 위하여 {timestamp, timestamp\_order, speaker, prev, target, next, label} 형태로 데이터 가공 필요
2. 텍스트 데이터의 {full text, segments{start, end, text, words {word, start, end, score}, language, duration, character} 형태를 1번에 기술된 형태로 가공 ({timestamp, timestamp\_order, speaker, prev, target, next, label} 부분만 bert 모델에 삽입, 나머지 부분은 디버깅용 확인 부분)
3. 가공된 데이터가 {timestamp, timestamp\_order, speaker, prev, target, next, label} 형태인지 확인
4. 실패 부분:

2-6 (발화자 태깅의 일관성)

* 예상 결과: 동일 인물에게 일관된 이름이 부여됨
* 실제 결과: 발화자에 맞게 태깅되지 않음

→ 위스퍼 모델 자체의 한계인 관계로 근본적 해결 불가 → 사용자가 직접 컨펌 후 업로드하도록 하여 오류내용을 직접 확인하도록 유도 예정

4.3 Bert 모델 & 후처리

| **테스트 케이스 ID** | **테스트 항목** | **테스트 설명** | **성능 기준** | **예상 결과** | **성공**  **여부** | **비고** |
| --- | --- | --- | --- | --- | --- | --- |
| 3-1 | 라벨 빈칸 유무 | 라벨에 빈칸은 없는지 | 모든 열에 예외 없이 빈칸 없음 | labels 컬럼에 빈칸 없음 | 성공 |  |
| 3-2 | 유무효 발화 비율 | 0과 1의 비율이 한쪽에 치우치지 않았는지 | labels 0:1 비율이 6:4 비율 이내 | 최대 6:4 비율 유지 | 성공 | 라벨 0, 1 확인후 문제없음 확인 |
| 3-3 | 데이터 제외 | 라벨 0인 데이터가 제외되지 않았는지 | 예외 없이 라벨 1만 제외됨 | 로그 파일에 라벨 0 데이터 없음 | 성공 |  |
| 3-4 | 로그 생성 | 라벨 1인 데이터의 로그 파일이 잘 생성되었는지 | 라벨 1 데이터가 빠짐없이 로깅됨 | 빠진 데이터 없음 | 성공 |  |
| 3-5 | 칼럼 삭제 | 라벨 컬럼이 삭제되었는지 | 라벨 컬럼 삭제됨 | 라벨 컬럼 없음 | 성공 |  |
| 3-6 | 라벨 분류 | 라벨은 적절히 분류되었는지 | 로그 목록에서 유효발화 비율이 20% 이내 | 유효 발화 비율 20% 이내 | 성공 |  |

단계별 진행 및 예상 결과:

1. Bert 모델 거쳐서 label에 유효 발화는 0, 무효 발화는 1로 기록되고, 끝부분에 confidence, text\_length 역시 계산되어 기록
2. 최상단엔 total\_triplets, 유효 발화인 valid\_count, 무효발화인 noise\_count, 전체 대비 노이즈 비율인 noise\_ratio, valid\_triplets가 있고, 1번은 이 valid\_triplets 안에 기록
3. 무효 발화 부분은 별도로 log 파일 생성되어 저장
4. LLM에 들어가기 위해, 기존 형태인 {total\_triplets, valid\_count, noise\_count, noise\_ratio, valid\_triplets{timestamp, timestamp\_order, speaker, prev, target, next, label}}에서 후처리를 거쳐 {filtered\_text, filtered\_triplets\_count, original\_length, filtered\_length, reduction\_ratio, filtered\_triplets{timestamp, timestamp\_order, speaker, text}} 형태로 가공 ({timestamp, timestamp\_order, speaker, text} 이 부분만 LLM에 삽입, 나머지 부분은 디버깅용 항목)
5. 이 형태로 출력되는지, 로그 파일에 label이 1인 것만 있는지 확인
6. 실패 부분: 없음

4.4 LLM

| **테스트 케이스 ID** | **테스트 항목** | **테스트 설명** | **성능 기준** | **예상 결과** | **성공**  **여부** | **비고** |
| --- | --- | --- | --- | --- | --- | --- |
| 4-1 | 회의 요약 | 회의 요약이 잘 생성되었는지 | 문장, 문체에 문제 없음. 전반적 내용을 잘 함축함. | 중요한 내용이 생략되거나 제외되지 않음 | 성공 |  |
| 4-2 | 태스크 생성 | 태스크가 빠짐없이 생성되었는지 | 빠진 태스크 없음 | 회의록에 만들어진 태스크가 예외 없이 생성됨 | 성공 |  |
| 4-3 | 업무 분배 | 업무 대상이 정확히 설정되었는지 | 대화에서 언급된 대상에게 정확하게 업무가 할당됨 | 실제 데이터의 언급 대상과 회의록에서 생성된 태스크 대상이 일치함 | 실패 | 업무 대상이 잘못 표시 |
| 4-4 | 공백 처리 | 빈칸은 내용 없음으로 처리되었는지 | 빈칸으로 반환하지 않고 내용 없음으로 처리됨 | 회의록 결과에 빈칸 없음 | 성공 |  |
| 4-5 | 내용 요약 | 요약의 구조화 여부 | 요약 결과가 기획안 템플릿에 맞게 구조화 | 개요/기능/일정 등 항목별 정리됨 | 성공 |  |
| 4-6 | 정확도 측정 | Epic/Story/Task 분류 정확성 | 분류 기준이 일관되고 논리적임 | 각 업무가 올바른 단계로 배치됨 | 실패 | 태스크, 서브태스크 생성 안됨 |

단계별 진행 및 예상 결과:

1. LLM에 들어가기 전 후처리를 통해 {original\_filtered\_text, stats{original\_length, refined\_length, sentence\_count, paragraph\_count, estimated\_tokens}, cleaned\_sentences, paragraph} 형태로 생성
2. 이 형태를 LLM에 삽입하여 notion 페이지 생성용 데이터 생성
3. 2번으로부터 Task와 subtask 정리 데이터 생성
4. 정해진 형태로 출력되는지, 태스크가 제대로 생성되고 빠진 게 없는지, 업무 분배가 제대로 되는지, 빈칸 처리가 되는 지 등을 확인
5. 실패 부분:

4-3 (업무 분배)

* 예상 결과: 실제 데이터의 언급 대상과 회의록에서 생성된 태스크 대상이 일치함
* 실제 결과: 업무 대상이 잘못 표시

→ 제대로 업무 분배가 이루어질 수 있도록 프롬프트 엔지니어링 및 재확인 프롬프트를 통해 수정 예정

4-6 (정확도 측정)

* 예상 결과: 각 업무가 올바른 단계로 배치됨
* 실제 결과: 태스크, 서브태스크 생성 안됨

→ 태스크, 서브태스크 생성하여 배치되도록 프롬프트 엔지니어링을 통해 수정 예정

1. **과거 이력 관리**

| **테스트 항목** | **버전** | **테스트 날짜** | **테스트 결과** | **수정 사항** |
| --- | --- | --- | --- | --- |
| 모듈 병합 테스트 | v1.0 | 2025-08-07 | 성공 | 초기 통합 환경 안정성 확인 완료 |
| 모듈 기능 테스트 | v1.1 | 2025-08-08 | 성공 | 발화자 태깅 일관성 수정 필요 |

1. **테스트 결과 및 결론**

본 1차 모듈 통합 테스트는 AI 엔진, 백엔드 서버, 프론트엔드, 외부 서비스 연동 모듈의 기능성과 성능을 종합적으로 검증하는 데 중점을 두었습니다.

기능 테스트와 성능 테스트 결과, 전체 모듈의 약 80% 이상이 정상 동작하였으며, LLM 업무 분배 및 정확도 측정 항목에서 일부 오류가 확인되었습니다.

이번 테스트 결과, 일부 기능(화자 구분, 업무 분배, 태스크 분류)에서 정확도 저하같은 오류가 발견되었는데, 이는 STT 모델인 WhisperX의 한계, 담당자 매핑과 업무 분류에 관한 프롬프트 엔지니어링 미비의 문제였습니다.

따라서 사용자가 컨펌하도록 유도하고, 프롬프트 엔지니어링 재설정 및 재확인 프롬프트를 추가하여 담당자 매핑과 업무 분류에 대해 개선 뒤 버전 테스트를 통해 오류 부분이 수정되었는지에 대해 v1.2에 반영할 예정입니다.