

Daily Progress Report — 2025-09-25 (Thu)

Prepared by: Tiong-Sik Ng 

English 

1) Summary of Accomplishments

- Focused on synchronization between utterances, chatbot output text, and generated TTS audio responses.
- Developed and refined migration scripts to normalize utterances and align cache keys.
- Ensured consistent handling of both utterance-hash and response-hash entries in DynamoDB for reliable cache hits.
- Generated a wide range of utterance variants (e.g., “지금 바빠요” , “바빠요” , “바빠”) to test and validate normalization and cache behavior.

2) Current System Status

Working Components

- Utterance normalization and hash synchronization scripts.
- Cache hit/miss flow with response-hash fallback.
- Variant generation tools for testing multiple utterance forms.

3) Technical Validation Results

- Verified that normalized utterances correctly map to cache entries.
- Confirmed that response-hash rows are created and linked during migration.
- Variant generation successfully demonstrates normalization consistency.

4) Next Steps

- Expand migration to full dataset in DynamoDB.
- Conduct larger-scale Connect call-flow testing with multiple utterance variants.
- Prepare workflow documentation for variant-based testing and cache validation.

5) Risk Assessment & Mitigation

Low Risk

- Normalization logic is stable and consistent across scripts.

Medium Risk

- Large numbers of utterance variants may grow DynamoDB table size and require pruning/limits.

Mitigation Strategy

- Limit variant generation per base utterance and monitor table growth.
-

Korean 🇰🇷

1) 성과 요약 ✅

- 발화, 챗봇 출력 텍스트, 생성된 음성 응답(TTS) 간 동기화 작업에 집중.
- 발화를 정규화하고 캐시 키를 맞추기 위한 마이그레이션 스크립트 개발 및 개선.
- 캐시 적중률 향상을 위해 발화 해시와 응답 해시 항목을 모두 일관되게 처리.
- 다양한 발화 변형(예: “지금 바빠요” , “바빠요” , “바빠”)을 생성하여 정규화 및 캐시 동작 검증.

2) 현재 시스템 상태 ↻

작동 중인 구성요소

- 발화 정규화 및 해시 동기화 스크립트.
- 응답 해시 기반의 캐시 히트/미스 처리 흐름.
- 다중 발화 형태를 위한 변형 생성 도구.

3) 기술적 검증 결과 📊

- 정규화된 발화가 캐시 항목에 올바르게 매핑되는 것을 확인.
- 마이그레이션 과정에서 응답 해시 행이 생성 및 연결되는 것을 검증.
- 발화 변형 생성이 정규화 일관성을 성공적으로 보여줌.

4) 다음 단계 📌

- DynamoDB 전체 데이터셋으로 마이그레이션 확장.
- 다중 발화 변형을 포함한 Connect 콜 플로우 대규모 테스트 수행.
- 변형 기반 테스트 및 캐시 검증을 위한 워크플로우 문서 준비.

5) 위험 평가 및 완화 🛡️

낮은 위험

- 정규화 로직이 안정적이고 스크립트 간 일관성 유지.

중간 위험

- 발화 변형 수가 많아지면 DynamoDB 테이블 크기가 증가할 수 있으며, 정리 또는 제한 필요.

완화 전략

- 기본 발화당 변형 생성 개수를 제한하고 테이블 성장을 모니터링.