CALL 모델

모델 정의서

AI 기반 통합 통화 품질 분석 모델

팀원: 오현서, 김기훈, 노준석, 오정우

목차

1. 모델 개요

1.1. 모델 정의 및 개발 배경

- 모델 정의

Call 모델은 고객 상담 통화의 품질을 종합적으로 분석하고 평가하기 위해 설계된 AI 기반의 엔드-투-엔드(End-to-End) 자동화 모델입니다. 본 모델은 음성-텍스트 변환(STT), 화자 분리(Speaker Diarization), 자연어 처리(NLP), 그리고 대형 언어 모델(LLM)을 포함한 최신 AI 기술을 유기적으로 통합합니다. 원본 음성 파일을 입력받아, 상담 품질을 다각도로 측정하는 6개 핵심 영역의 18개 정량적 지표를 자동으로 산출함으로써, 원시 데이터를 실행 가능한 통찰력으로 변환합니다.

- 개발 배경

전통적인 콜센터 품질 보증(Quality Assurance, QA) 프로세스는 상당한 한계점을 내포하고 있습니다. 대부분의 QA 활동은 전체 통화 중 극히 일부만을 무작위로 샘플링하여 평가자가 수동으로 청취하고 주관적으로 평가하는 방식에 의존합니다. 이러한 접근법은 다음과 같은 근본적인 문제점을 가집니다.

1. 자원 집약성 및 확장성 부족: 모든 통화를 수동으로 평가하는 것은 물리적으로 불가능하며, 샘플링 기반 평가는 전체 상담 품질을 대표하지 못할 수 있습니다.
2. 주관성 및 비일관성: 평가자의 컨디션, 경험, 개인적 편향에 따라 평가 결과가 달라져 일관성과 객관성을 확보하기어렵습니다.
3. 피드백 지연: 분석과 피드백 전달에 시간이 소요되어, 실시간 개선이나 즉각적인 코칭이 어렵습니다.

특히 한국어 상담 환경은 단순한 내용 전달을 넘어, 존댓말 사용, 완곡어법, 공감 표현 등 복합적인 사회언어학적 요소가 서비스 만족도에 지대한 영향을 미칩니다. 이러한 문화적, 언어적 뉘앙스는 기존의 범용 분석 도구로는 정확히 포착하기 어렵습니다. Call 모델은 이러한 배경 하에, 한국어 상담 환경의 특수성을 깊이 이해하고 이를 정밀하게 분석할 수 있는 특화된 솔루션을 제공함으로써, 기존 QA 방식의 한계를 극복하고자 개발되었습니다.

1.2. 모델 목표

본 모델은 다음의 네 가지 핵심 목표를 달성하는 것을 지향합니다.

- 완전 자동화: 음성 파일 입력부터 최종 18개 지표가 포함된 분석 보고서 생성까지, 인간의 개입이 없는 100% 자동화된 파이프라인을 구축합니다.

- 데이터 기반 객관성(Data-driven Objectivity): 평가자의 주관적 판단을 배제하고, 사전에 정의된 18개의 정량적 지표를 통해 모든 상담을 일관된 기준으로 평가합니다.
- 준실시간 처리: 평균 50초 내외의 빠른 처리 속도를 통해 상담 종료 후 거의 즉각적으로 분석 결과를 제공하여, 신속한 피드백과 운영 개선을 가능하게 합니다.
- 한국어 환경 최적화: 존댓말 사용률, 공감 표현의 진정성 등 한국의 독특한 상담 문화를 반영한 지표를 개발하여, 분석의 깊이와 실효성을 극대화합니다.

Call 모델의 핵심 가치는 '한국어 문화에 최적화된 하이브리드 AI 아키텍처'에 있습니다. 이는 단순히 데이터를 분석하는 것을 넘어, 한국 비즈니스 환경에서 실질적인 가치를 창출하는 의미 있는 통찰력을 제공하는 데 중점을 둡니다. 본 모델은 기술적 정교함과 상업적 실용성을 겸비한 솔루션으로서, 고객 경험 관리의 새로운 표준을 제시합니다.

2. 모델 아키텍처와 선정 근거

2.1. 하이브리드 AI 아키텍처의 당위성

Call 모델의 아키텍처는 단일 기술에 의존하는 '순수 LLM' 또는 '순수 규칙 기반' 접근법을 의도적으로 지양합니다. 대신, 각 과업의 특성에 가장 적합한 도구를 사용하는 실용주의적 철학에 기반한 하이브리드 AI 아키텍처를 채택했습니다. 이 구조는 정확성, 속도, 그리고 비용 효율성 간의 최적의 균형을 달성하기 위해 전략적으로 설계되었습니다.

- 규칙 기반(Rule-based) 접근의 역할: 규칙 기반 시스템은 속도가 빠르고, 결과가 결정론적이며, 계산 비용이 저렴하다는 장점이 있습니다. 이는 "죄송합니다", "감사합니다"와 같은 특정 키워드를 탐지하거나, '-습니다', '-세요'와 같은 명확한 문법적 패턴(존댓말 어미)을 식별하는 작업에 이상적입니다. Call 모델에서는 이러한 고빈도, 저모호성 작업을 규칙 기반으로 우선 처리하여 전체 파이프라인의 효율성을 극대화합니다.
- LLM 기반(LLM-based) 접근의 역할: 대형 언어 모델(LLM)은 문맥, 뉘앙스, 그리고 숨겨진 의미를 파악하는 데 있어 타의 추종을 불허하는 능력을 보입니다. 이는 사과의 진정성을 평가하거나, 문제 해결 제안의 논리적 타당성을 분석하는 등 복잡하고 정성적인 판단이 요구되는 작업에 필수적입니다. 규칙만으로는 포착할 수 없는 대화의 깊이를 이해하는 데 LLM이 활용됩니다.
- 시너지 창출: '필터링 및 검증' 모델: 본 아키텍처의 핵심적인 작동 원리는 '규칙 기반 필터링 후 LLM 검증'이라는 2단계 프로세스입니다. 모델은 먼저 계산적으로 저렴한 규칙 기반 방식을 사용해

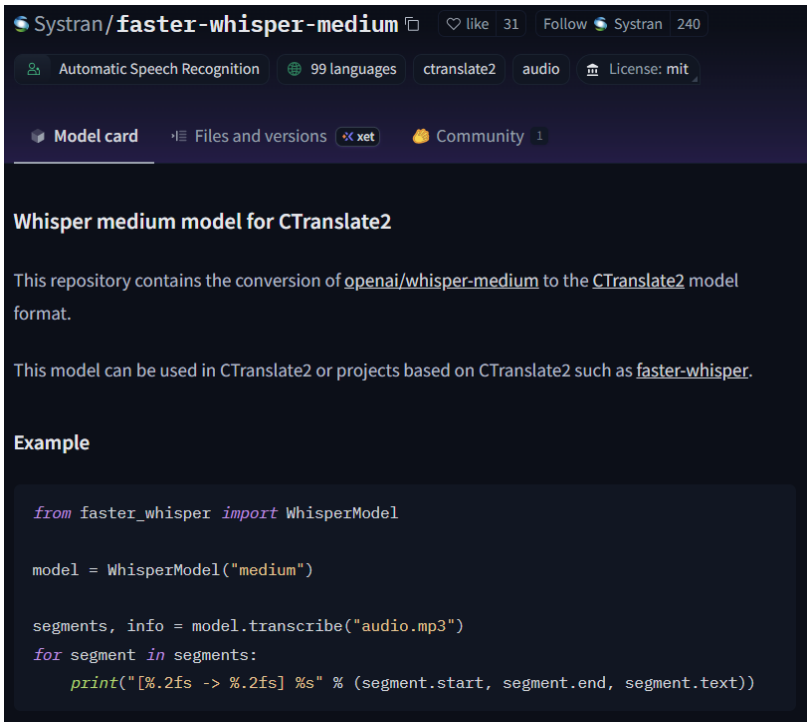
분석이 필요한 후보군(예: 공감 표현으로 추정되는 문장)을 신속하게 식별합니다. 그 후, 이 후보군만을 선별하여 계산 비용이 높은 LLM에 전달, 심층적인 문맥 분석과 검증을 수행합니다. 이 접근법은 LLM의 강력한 분석 능력을 꼭 필요한 곳에만 집중적으로 사용함으로써, 전체 모델의 처리 속도와 비용 효율성을 혁신적으로 개선합니다.

2.2. 구성 모델 선정 근거

Call 모델을 구성하는 각 AI 모델은 상기한 하이브리드 아키텍처 철학에 따라 신중하게 선정되었습니다.

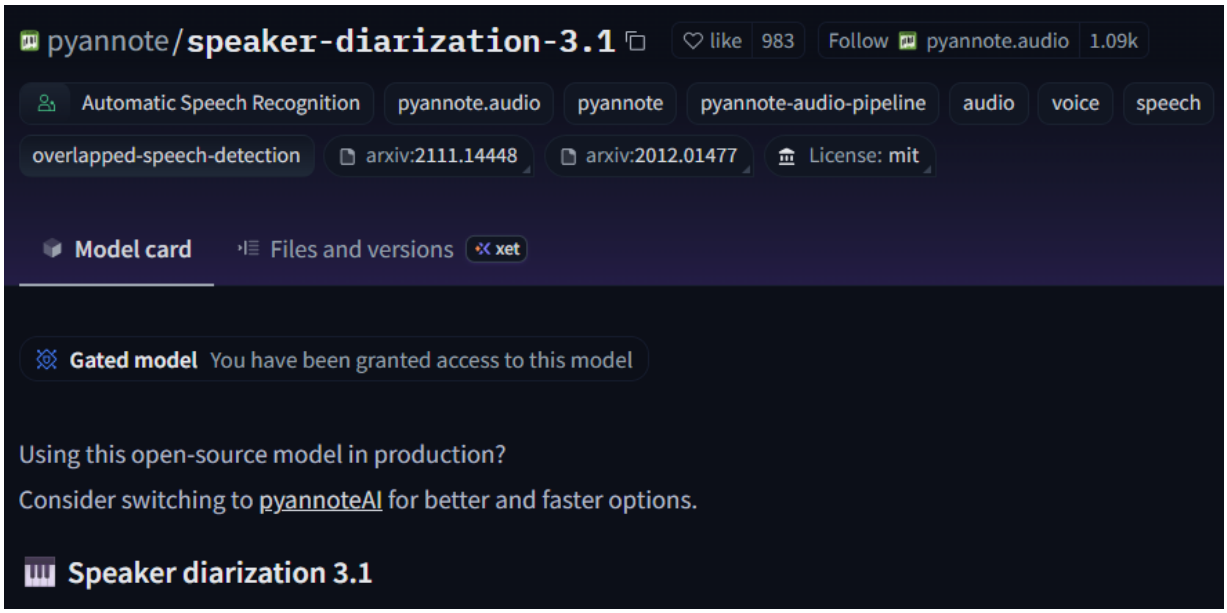
- 음성 인식 (STT): faster-whisper (medium)

- 선정 근거: faster-whisper는 OpenAI의 Whisper 모델을 CTranslate2로 재구현하여 속도와 효율성을 크게 개선한 버전입니다. 특히 한국어 음성 데이터에 대해 현존 최고 수준의 인식 정확도를 제공하며, 이는 모든 후속 자연어 처리 분석의 품질을 결정하는 가장 중요한 전제 조건입니다. 'medium' 크기 모델은 최상의 정확도와 합리적인 리소스 사용량 사이의 최적의 균형점을 제공합니다.
- 기술적 타당성: int8 정수 양자화(quantization) 기술을 적용하여 모델의 메모리 점유율을 줄이고 GPU에서의 추론 속도를 추가적으로 가속화했습니다. 이는 정확도 저하를 최소화하면서 상용 서비스에 필수적인 처리 효율성을 확보하기 위한 전략적 선택입니다.



- 화자 분리 (Speaker Diarization): pyannote/speaker-diarization-3.1

- 선정 근거: pyannote.audio의 3.1 버전은 상담사와 고객이라는 2인 대화 시나리오에서 매우 높은 분리 정확도를 자랑합니다. 이는 각 발화의 주체를 명확히 구분하여, 상담사 평가 지표와 고객 감정 지표를 정확하게 계산하기 위한 필수적인 단계입니다.
- 기술적 타당성: pyannote 모델이 다른 PyTorch 기반 모델과 동일한 런타임에서 실행될 때 발생할 수 있는 CUDA 컨텍스트 충돌 문제를 해결하기 위해, 멀티프로세싱 래퍼를 구현했습니다.



- 대화 내용 분석 (LLM): OpenAI GPT-4.1-nano

- 선정 근거: 상담 주제 분류, 결과 평가, 비속어 탐지, 문제 해결 능력 평가, 감정의 뉘앙스 분석, 그리고 공감/사과 표현의 진정성 검증과 같은 고차원적인 분석 과업들은 단순 키워드 매칭이나 감성 사전을 넘어선 깊은 문맥 이해와 추론 능력을 요구합니다. GPT-4 클래스의 모델은 이러한 복잡한 정성적 판단을 안정적으로 수행할 수 있는 능력을 갖추고 있습니다.
- 기술적 타당성: 모델명에 포함된 'nano'는 대규모 GPT-4 모델 대비 낮은 지연 시간과 비용에 최적화된 버전임을 시사합니다. 이는 모델의 준실시간 처리 목표와 상업적 운영의 경제성을 동시에 만족시키기 위한 전략적 선택입니다.

- 한국어 자연어 처리 (Korean NLP): kiwipiepy, kss

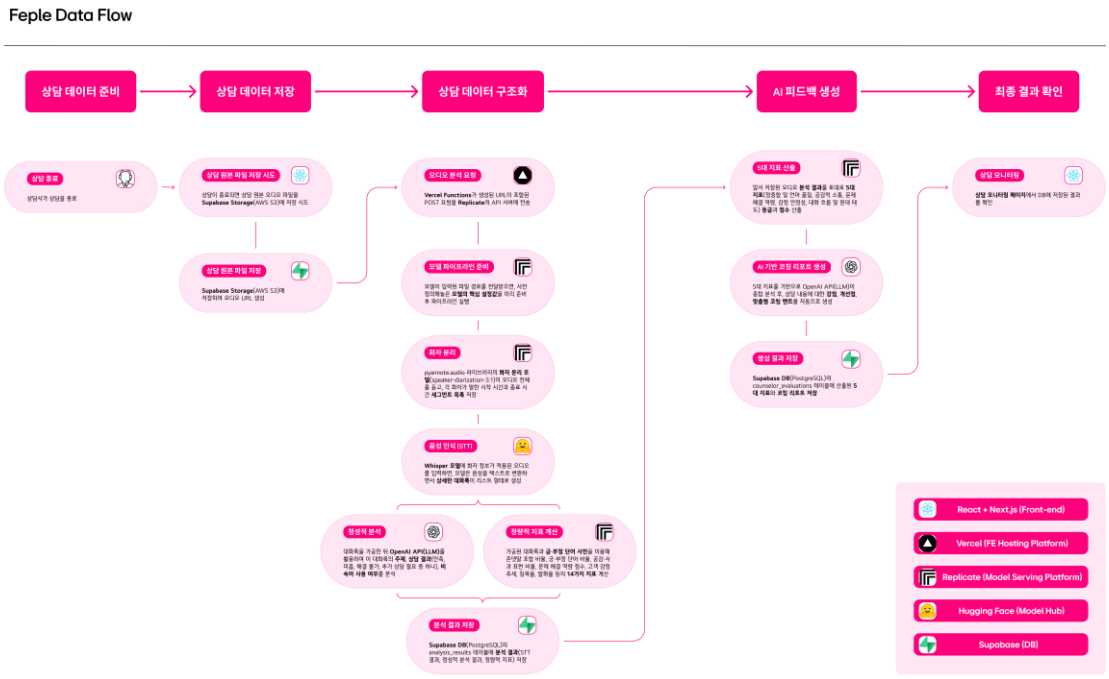
- 선정 근거: 한국어의 교착어적 특성과 문맥 의존적인 문장 구조를 정확하게 처리하기 위해, 범용 다국어 라이브러리 대신 한국어에 특화된 전문 라이브러리를 채택했습니다. 형태소 분석기인

kiwipiepy와 문장 분리 라이브러리인 kss(Korean Sentence Splitter)는 존댓말 어미 분석, 감성 사전 매핑 등에서 월등한 정확도를 제공하며, 이는 honorific_ratio와 같은 핵심 지표의 신뢰도를 직접적으로 보장합니다.

3. 데이터 처리 파이프라인

Call 모델은 원본 음성 파일이 입력된 순간부터 최종 분석 결과인 JSON 객체가 반환될 때까지, 다음과 같은 체계적인 단계를 거쳐 데이터를 처리합니다.

3.1. 전체 처리 흐름도



3.2. 단계별 상세 처리 과정

1. 화자 분리(Speaker Diarization): pyannote 모델이 전체 오디오를 분석하여, 어떤 화자(SPEAKER_00 또는 SPEAKER_01)가 언제부터 언제까지 발화했는지에 대한 시간 세그먼트 목록을 생성합니다. 이때 clustering_threshold=0.6과 같은 핵심 파라미터는 2인 대화 환경에 최적화된 설정값으로, 분리 정확도를 높이는 데 결정적인 역할을 합니다.

```
# 1. 화자 분리
diarization_start_time = time.time()
result_queue = Queue()
diarization_process = Process(target=run_diarization_in_process, args=(audio_path, self.hf_token, result_queue))
diarization_process.start()
speaker_turns = result_queue.get()
diarization_process.join()

if isinstance(speaker_turns, Exception):
    raise speaker_turns
```

2. 음성 인식(STT): faster-whisper 모델이 오디오를 텍스트로 변환하면서, 스크립트의 각 단어에 대한 정확한 시작 및 종료 시간 정보를 함께 출력합니다. 이 단어 수준의 타임스탬프는 다음 통합 단계의 정확성을 위한 핵심 데이터입니다.

```
# 2. 음성 인식 (STT)
stt_start_time = time.time()
audio_waveform, sr = librosa.load(audio_path, sr=16000, mono=True)
total_duration = len(audio_waveform) / sr
word_segments = self._run_stt(audio_waveform)
processing_times['stt'] = time.time() - stt_start_time
print(f"음성 인식(STT) 완료. (소요 시간: {processing_times['stt']:.2f}초)")
```

3. 결과 통합(Diarization & STT Merge): 이 단계는 두 모델의 출력을 결합하는 모델의 핵심 로직 중 하나입니다.
    - 단어 중심점 매핑: STT 결과로 나온 각 단어의 시간적 중심점(tmidpoint=(tstart+tend)/2)을 계산합니다.
    - 화자 할당: 계산된 중심점 시간(tmidpoint)이 화자 분리 결과로 나온 시간 세그먼트 중 어느 화자의 구간에 속하는지를 확인하여 해당 단어의 화자를 결정합니다. 이 방식은 두 모델 간의 미세한 시간 불일치가 발생하더라도 각 단어에 대한 화자 할당을 매우 강건하게 만듭니다.
    - 발화 병합: 동일한 화자에게 연속적으로 할당된 단어들을 하나의 의미 있는 발화 단위로 병합합니다.

```
# 3. 결과 종합
merge_start_time = time.time()
structured_transcript = self._merge_results(speaker_turns, word_segments)
processing_times['merge'] = time.time() - merge_start_time
print(f"결과 종합 완료. (소요 시간: {processing_times['merge']:.2f}초)")
```

4. 후처리 및 역할 할당 (Post-processing and Role Assignment):
    - 역할 식별: 모델은 SPEAKER_00과 SPEAKER_01에 'Agent'(상담사)와 'Customer'(고객) 역할을 자동으로 부여합니다. 이때 사용되는 핵심 휴리스틱(heuristic)은 통계적 특성입니다. 일반적으로 상담사는 고객보다 발화량이 많고(talk_ratio), 존댓말 사용 빈도(honorific_ratio)가 높다는 특성을 이용합니다. 이는 시스템이 자신의 분석 결과를 내부 처리에 재활용하는 지능적인 설계입니다.
    - 텍스트 정규화: 후속 NLP 분석의 정확도를 높이기 위해, STT 과정에서 발생할 수 있는 인식 오류나 불필요한 기호를 제거하는 정제 작업을 수행합니다.

```
# 4. 후처리
postprocess_start_time = time.time()
final_transcript = self._postprocess_transcript(structured_transcript)
processing_times['post_processing'] = time.time() - postprocess_start_time
print(f"후처리 완료. (소요 시간: {processing_times['post_processing']:.2f}초)")
```

3.3. 최종 데이터 구조

모델의 모든 처리 결과는 구조화된 JSON 형식으로 제공됩니다. 이 형식은 기계가 읽기 용이하며, 다른 모델과의 연동을 위한 API 응답으로 최적화되어 있습니다. 최종 출력은 처리 시간, 전체 대화 스크립트, 그리고 18개 핵심 지표의 세 부분으로 구성됩니다.

4. 통합 분석 지표 체계

Call 모델의 핵심 자산은 6개 영역으로 구성된 18개의 정량·정성적 분석 지표입니다. 이 프레임워크는 상담사의 성과를 언어적 품질, 감성 지능, 운영 효율성 등 다각적인 측면에서 종합적으로 평가하기 위해 설계되었습니다. 각 지표는 단순한 수치를 넘어, 구체적인 개선 포인트를 도출할 수 있는 진단 도구로서의 역할을 합니다.

4.1. 상담 메타데이터 (Session Metadata) - 4개 지표

이 지표들은 개별 상담 세션을 식별하고 전체적인 맥락을 요약합니다.

지표	컬럼명	데이터 타입	설명	계산 방식
세션 고유 ID	session_id	string	각 상담 세션을 유일하게 식별하는 번호	파일 처리 시점의 타임스탬프와 난수 조합으로 생성
주제 분류	mid_category	string	상담의 핵심 주제 (총 11개 카테고리)	LLM이 전체 대화 내용을 분석하여 사전 정의된 11개 카테고리 중 가장 적합한 것을 선택 (Zero-shot Classification)
상담 결과	result_label	string	상담의 최종 마무리 상태에 대한 평가	LLM이 대화의 종결부와 고객의 반응을 바탕으로 '만족', '미흡', '해결불가', '추가상담필요' 4단계로 분류
비속어 사용	profane	int	고객의 비속어 또는 공격적 언어 사용 여부	LLM이 고객 발화 전체에서 비속어 및 공격적 표현의 존재 유무를 탐지하여 이진값(0 또는 1)으로 반환

4.2. 정중함 및 언어 품질 (Politeness & Language Quality) - 4개 지표

상담사의 언어적 전문성과 고객 응대 태도의 기본을 측정합니다.

지표	컬럼명	데이터 타입	설명	계산 방식
존댓말 사용률	honorific_ratio	float	상담사 발화 문장 중 존댓말이 사용된 비율 (%)	kiwipiepy 형태소 분석을 통해 존댓말을 나타내는 어미('습니다', 'ㅂ니다', '세요', '셔요', '까요') 및 선어말어미('시') 를 탐지. 계산식: Ratio=(Ntotal_agent_sentences Nhonorific_sentences)×100
긍정어 비율	positive_word_ratio	float	상담사 발화 형태소 중 긍정적 단어의 비율 (%)	KNU 한국어 감성사전을 기반으로, 극성값(polarity) 이 0보다 큰 긍정 형태소의 수를 전체 형태소 수로 나누어 계산. 계산식: Ratio=(Ntotal_morphemes Npositive_morphemes) ×100
부정어 비율	negative_word_ratio	float	상담사 발화 형태소 중 부정적 단어의 비율 (%)	KNU 한국어 감성사전을 기반으로, 극성값이 0보다 작은 부정 형태소의 수를 전체 형태소 수로 나누어 계산. 계산식: Ratio=(Ntotal_morphemes Nnegative_morphemes)×100
완곡어 사용률	euphonious_word_ratio	float	부드럽고 정중한 인상을 주는 표현의 사용 비율 (%)	규칙 기반 패턴 매칭. 쿠션어("실례지 만", "죄송하지만", "혹시")와 완곡 표현("~인 것 같습니다", "~ㄹ 수 있습니다")을 포함하는 문장의 비율을 계산. 계산식: Ratio=(Ntotal_agent_sentences Neuphonious_sentences)×100

4.3. 공감적 소통(Empathy) - 2개 지표

고객의 감정을 이해하고 이에 적절히 반응하는 능력을 평가합니다.

지표	컬럼명	데이터 타입	설명	계산 방식
공감 표현률	empathy_ratio	float	진정성 있는 공감 표현이 사용된 문장의 비율 (%)	2단계 '필터링 및 검증' 방식 1. 규칙 기반 후보 탐지: 키워드("공감", "이해", "걱정") 및 패턴("~하셨겠어요", "힘드셨죠")으로 공감 표현 후보 문장 식별. 2. LLM 검증: 후보 문장을 LLM에 제시하여, "이 표현이 고객의 상황에 대한 진심 어린 공감인가, 아니면 기계적인 동의인가?"를 판단하게 하여 진정성을 검증.
사과 표현률	apology_ratio	float	진정성 있는 사과 표현이 사용된 문장의 비율 (%)	2단계 '필터링 및 검증' 방식 1. 규칙 기반 후보 탐지: 키워드("죄송", "사과", "미안", "양해")로 사과 표현 후보 문장 식별. 2. LLM 검증: 후보 문장을 LLM에 제시하여, "이 표현이 진정한 사과의 의도를 담고 있는가, 아니면 책임을 회피하는 조건부 사과인가?"를 판단하게 하여 진정성을 검증.

4.4. 문제 해결 역량(Problem Solving) - 1개 지표

상담의 핵심 목적인 고객 문제 해결의 효율성을 측정합니다.

지표	컬럼명	데이터 타입	설명	계산 방식
해결 제안력	suggestions	float	구체적인 해결 방안을 제시하여 문제를 해결하는 능력 점수	LLM 기반 단계별 평가: LLM이 전체 대화의 흐름을 분석하여, 고객의 문제가 상담사의몇번째 제안으로 해결되었는지를 추론. • 1.0점: 최초 제안으로 문제 해결 • 0.6점: 두 번째 제안으로 문제 해결 • 0.2점: 세 번 이상의 제안으로 문제 해결 • 0.0점: 문제 미해결

4.5. 감정 안정성(Emotional Stability) - 3개 지표

상담 과정에서 고객의 감정 변화를 추적하여 서비스의 질적 영향을 평가합니다.

지표	컬럼명	데이터 타입	설명	계산 방식
초반 고객 감정	customer_sentiment_early	float	상담 시작 후 첫 33% 구간의 고객 감정 평균	kss로 분리된 고객의 각 문장을 LLM이 -1(부정), 0(중립), 1(긍정)로 평가. 초반 33%에 해당하는 문장들의 감정 점수 평균값.
후반 고객 감정	customer_sentiment_late	float	상담 마지막 33% 구간의 고객 감정 평균	후반 33%에 해당하는 고객 문장들의 감정 점수 평균값.
감정 개선 추세	customer_sentiment_trend	float	상담을 통한 고객 감정의 긍정적 변화 정도	계산식: Trend=customer_sentiment_late −customer_sentiment_early. 양수일 경우 감정 개선, 음수일 경우 감정 악화를 의미.

4.6. 대화 흐름 및 응대 태도(Flow & Stability) - 4개 지표

대화의 속도, 주도권, 안정성 등 운영 효율성과 관련된 요소를 측정합니다.

지표	컬럼명	데이터 타입	설명	계산 방식
평균 응답 시간	avg_response_la tency	float	고객 발화 종료 후 상담사가 응답하기까지 걸린 평균 시간(초)	타임스탬프 기반 계산. 고객 발화 종료 시점과 다음 상담사 발화 시작 시점 간의 시간 차이(Δt>0)들의 평균을 산출.
끼어들기 횟수	interruption_cou nt	int	상담사가 고객의 말을 중간에 끊은 횟수	타임스탬프 기반 겹침 감지. 상담사의 발화 시작 시간이 고객의 발화 종료 시간보다 이전인 경우(Agentstart_time<Customerend_time)를 카운트. (단, 200ms 미만의 자연스러운 겹침은 제외)
침묵 비율	silence_ratio	float	전체 대화 시간 중 아무도 말하지 않는 침묵 구간의 비율	계산식: Ratio=TtotalTtotal−(Tagent_talk+Tcustomer_talk). 0과 1 사이의 값으로 표현.
발화 시간 비율	talk_ratio	float	전체 발화 시간 중 상담사가 차지하는 시간의 비율	계산식: Ratio=Tagent_talk+Tcustomer_talkTagent_talk. 0.5보다 크면 상담사의 발화량이 많음을 의미.

5. 모델 성능과 기술 사양

5.1. 모델 처리 속도와 효율성 분석

Call 모델은 상용 환경에서의 실용성을 담보하기 위해 처리 속도와 효율성에 대한 면밀한 분석을 거쳤습니다. 평균적인 5분 길이의 상담 통화 파일을 기준으로 한 성능 벤치마크 결과는 다음과 같습니다.

처리 단계	평균 소요 시간	전체 대비 비중	분석
화자 분리 (Diarization)	약 25초	50%	현재 모델의 주요 성능 병목(bottleneck) 구간. 전체 처리 시간의 절반을 차지하며, 향후 최적화의 최우선 순위 대상.
음성 인식(STT)	약 16초	32%	GPU 가속 및 int8 양자화 적용을 통해 높은 처리 효율성을 달성.
지표 계산(Metrics)	약 9초	18%	규칙 기반 필터링과 선택적 LLM 호출을 통해 복잡한 분석을 효율적으로 수행.
총 처리 시간	약 50초	100%	준실시간 분석이라는 프로젝트 목표를 성공적으로 달성.

이 분석을 통해, 현재 모델의 성능은 화자 분리 단계에 가장 크게 의존하고 있음을 명확히 알 수 있습니다. STT 단계에서의 성공적인 최적화 경험을 바탕으로, 향후 화자 분리 모델의 경량화 또는 파이프라인 개선을 통해 전체 처리 속도를 더욱 단축할 수 있는 잠재력이 존재합니다.

5.2. 기술 스택과 배포 환경

Call 모델은 안정적이고 확장 가능한 운영을 위해 검증된 최신 기술들로 구성되었습니다.

구분	기술 / 라이브러리	버전	주요 역할
언어	Python	3.12	모델 핵심 개발 언어
AI/ML 프레임워크	PyTorch + CUDA	2.5.1	모든 딥러닝 모델의 기반 프레임워크 및 GPU 가속
음성 인식(STT)	faster-whisper	1.1.1	음성-텍스트 변환 및 단어 타임스탬프 생성
화자 분리 (Diarization)	pyannote.audio	3.1.1	2인 대화 화자 분리
LLM 인터페이스	openai	1.93.0	GPT-4.1-nano 모델 API 연동
한국어 NLP	kiwipiepy, kss	4.5.4	형태소 분석 및한국어 특화 문장 분리
오디오 처리	librosa	0.11.0	오디오 파일 로딩 및 전처리
배포 환경	Replicate (GPU)	-	컨테이너 기반의 서버리스 GPU 플랫폼을 통한 모델 서빙

배포 환경으로 Replicate를 선택한 것은 복잡한 GPU 인프라 관리, 오토스케일링, 모델 버전관리등의 MLOps 부담을 최소화하고, 핵심 AI 모델개발에 역량을 집중하기 위한 전략적 결정입니다. 모델은 표준 REST API 엔드포인트를 통해 외부에 서비스를 제공하며, 이는 고객사의 CRM, QA 대시보드 등 기존 시스템과 유연하게 통합될 수 있는 구조를 보장합니다.

6. 모델 활용 방안 및 향후 개선 계획

6.1. 주요 활용 분야

Call 모델은 단순한 기술 시연을 넘어, 실제 비즈니스 현장에서 구체적인 가치를 창출하도록 설계

- 콜센터 품질 관리(QA) 혁신: 100%의 통화를 실시간으로 분석하여, 샘플링 기반의 수동 평가를 대체합니다. 이를 통해 공정하고 일관된 평가 체계를 확립하고 QA 인력은 고부가가치 업무(코칭, 프로세스 개선)에 집중할 수 있습니다.
- 데이터 기반 상담사 코칭: 18개 지표로 구성된 다면적 대시보드를 통해 상담사 개개인의 강점과 약점을 정확히 진단합니다. "평균 응답 시간을 0.3초 단축해 보세요" 또는 "공감 표현 사용 빈도를 높여 고객 감정 개선 추세를 긍정적으로 만들어 보세요"와 같은 구체적이고 실행 가능한 피드백을 제공할 수 있습니다.
- 고객 경험(CX) 관리 고도화: customer_sentiment_trend 지표를 팀별, 상품별, 기간별로 추적하여 고객 불만을 야기하는 근본적인 원인을 파악하고 선제적으로 대응할 수 있습니다.
- 상담 프로세스 최적화: silence_ratio, talk_ratio 등의 지표를 분석하여 비효율적인 상담 스크립트나 업무 절차를 발견하고 개선함으로써, 평균 통화 시간(AHT) 단축 등 운영 효율성을 증대시킬 수 있습니다.

6.2. 향후 계획

Call 모델은 현재의 완성도에 머무르지 않고, 지속적인 발전을 통해 최고의 통화 분석 솔루션으로 성장하는 것을 목표로 합니다.

1. 성능 최적화: 현재 모델의 가장 큰 병목인 화자 분리 모델의 처리 속도 개선을 최우선 과제로 삼습니다. 더 가볍고 빠른 대체 모델을 연구하거나, 현재 모델의 파라미터를 미세 조정하여 성능 저하 없이 속도를 향상시키는 방안을 탐색할 계획입니다.
2. 자체 LLM 미세조정(Fine-tuning): 장기적으로는 외부 상용 API에 대한 의존도를 줄이고 비용 효율성을 높이기 위해, 경량화된 오픈소스 LLM(예: Llama, Polyglot-Ko 등)을 상담 도메인 특화 데이터로 미세조정하는 프로젝트를 추진합니다. 이는 '진정성'과 같은 특정 도메인 과업에 대한 정확도를 높이고 운영 비용을 절감하는 두 가지 효과를 기대할 수 있습니다.
3. 분석 지표 확장: 현재 18개 지표 외에, 상담 품질을 더욱 입체적으로 분석할 수 있는 신규 지표를 개발합니다. 예를 들어, 고객의 말을 경청하고 있음을 나타내는 '적극적 경청 지표'(예: "네, 고객님", "아, 그러셨군요" 등 맞장구 표현 탐지)나 '상향/교차 판매 시도 탐지'와 같은 비즈니스 기여도 관련 지표를 추가할 수 있습니다.
4. 사용자 인터페이스(UI) 개발: 현재의 JSON 출력 형식을 넘어, QA 관리자나 센터장이 직관적으로

데이터를 탐색하고 인사이트를 얻을 수 있는 웹 기반 시각화 대시보드를 구축합니다. 이 대시보드는 상담사별 성과 비교, 기간별 추이 분석, 특정 통화 상세 조회(스크립트-점수 연동) 등의 기능을 제공하여 데이터 활용성을 극대화할 것입니다.

이러한 로드맵을 통해 Call 모델은 정교한 기술 플랫폼을 넘어, 고객 서비스 산업의 성과를 견인하는 핵심적인 비즈니스 인텔리전스 도구로 자리매김할 것입니다.