TalkCatcher

임상·기술 명세서

28 개 알고리즘 · 7 Pillars · 3-Layer 자기개선 엔진

Marina (Alex)[1]

BCN · PhD · 전 Harvard GSE 객원학자 (Fischer Lab)

[1] *Boston Neuromind LLC, Canton, MA, USA*

Version 1.0 · 2026 년 5 월 13 일

초록 (Abstract)

TalkCatcher 는 성인 ADHD 를 위한 진단 특화 AI 동반자 시스템이며, 플러그인 아키텍처를 통해 9 개의 추가 DSM-5-TR 조건에 걸쳐 복제되도록 설계되었다. 본 문서는 완전한 기술 아키텍처를 명세한다: 3-Layer 자기개선 엔진, 4 단계 파이프라인으로 조직된 28 개의 전문화된 알고리즘, 7 개의 독점 임상 IP Pillar, 그리고 기저 대형 언어모델을 재훈련하지 않고도 벨트가 실세계 사용으로부터 성장하도록 하는 자동 성장 루프.

시스템은 교수설계, 정신건강 상담, 다이내믹 스킬 이론(Harvard GSE, Fischer Lab) 분야의 정식 훈련을 받은 단일 창립자/임상가에 의해 개발되고 있다. 범용 대화형 AI 와 달리 TalkCatcher 의 임상 콘텐츠는 큐레이션되고, 검색은 LLM 파라미터가 아닌 임상가가 저작한 벨트에 근거하며, 안전 오버라이드는 언어모델 자체와 독립적으로 작동한다. 8 개의 USPTO 임시 특허 명세서가 가장 새로운 구성요소를 다룬다.

이 문서는 완전한 알고리즘 명세가 필요한 임상가, 연구자, 규제 담당자, 투자자 및 기타 기술 독자를 대상으로 한다. 최종 사용자를 위한 별도의 비기술 문서가 제공된다.

목차 (Contents)

1. 서론 및 범위

1.1 문제

정신건강 분야에 적용된 범용 대화형 AI 는 반복되는 세 가지 실패에 직면한다. 첫째, 표면 수준에서는 유사하지만 정반대의 개입을 요구하는 장애들을 구별하지 못한다(안심시키기는 불안에는 도움이 되지만 OCD 에는 해롭다). 둘째, 콘텐츠가 큐레이션이 아닌 생성이므로 임상적 신뢰성을 보장할 수 없다. 셋째, 언어모델과 독립적으로 작동하는 내장 안전 오버라이드가 없으므로 적대적 입력이 안전을 완전히 우회할 수 있다.

진단 특화 시스템(예: ADHD 의 Inflow, OCD 의 NOCD, 불안/우울의 Wysa)은 범위를 좁혀 일부 문제를 다루지만, 각 회사는 그러한 시스템을 단 하나만 구축했다. 어떤 회사도 단일 엔진으로 여러 진단 특화 봇을 지원할 수 있는 통합 아키텍처를 구축하지 않았다.

1.2 TalkCatcher 접근법

TalkCatcher 는 엔진과 장애를 분리하여 이러한 실패를 다룬다. 엔진 — 4 단계 파이프라인으로 조직된 28 개 알고리즘 — 은 진단 비특이적이다. 각 장애는 플러그인이다: YAML 매니페스트, 임상가가 저작한 콘텐츠의 벨트, 검증된 임상 척도 세트, 개입 순서화 프로토콜. 플러그인을 교체하면 시스템이 ADHD 동반자에서 OCD 동반자로 바뀌며 엔진은 변경되지 않는다.

시스템은 한 명의 창립자/임상가(Marina), 한 명의 사용자/베타테스터/임상감독(Marina 의 파트너), 그리고 자동화된 합성 데이터 생성과 대형 언어모델 API 의 도움으로 구축되고 있다. 필요한 총 자본은 약 1 만 미국 달러 — 비교 가능한 VC 자금 지원 경쟁자보다 3 자릿수 낮다.

1.3 본 문서의 범위

본 문서는 시스템이 무엇인지와 알고리즘 수준에서 어떻게 작동하는지 명세한다. 사용자 대상 제품(자체 설계 문서 보유)이나 사업 전략(별도)은 다루지 않는다. 마케팅 없이 기술적 세부 사항을 원하는 독자를 대상으로 한다.

2. 시스템 아키텍처 — 3-Layer 자기개선 엔진

전체 시스템은 각각 다른 시간 척도에서 작동하는 세 개의 레이어로 조직된다. Layer 1 은 오프라인으로 합성 벨트 콘텐츠를 생성한다. Layer 2 는 사용자 대화를 처리하는 런타임 엔진이다. Layer 3 는 실 사용 데이터로부터 벨트를 성장시켜 루프를 닫는다.

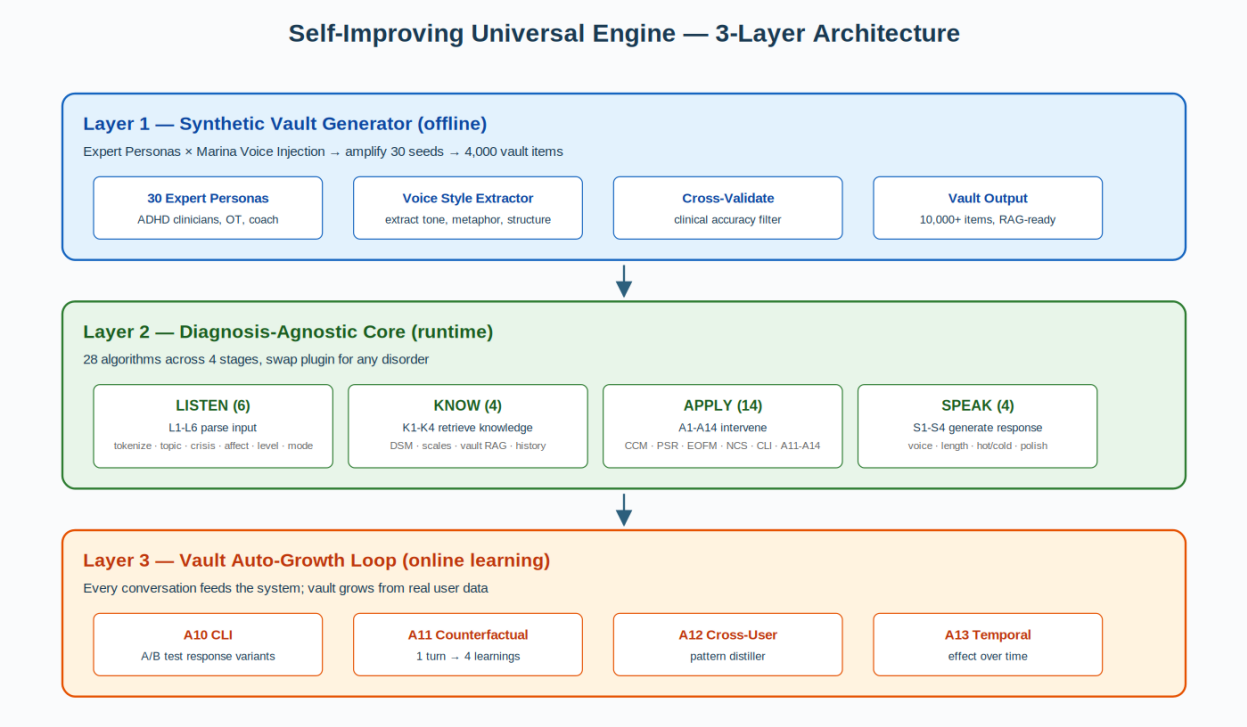

그림1. 3-Layer 자기개선 엔진.

2.1 Layer 1 — Synthetic Vault Generator (offline)

사용자 대화 이전에 생성기 파이프라인이 벨트 콘텐츠를 생산한다. LLM 이 30 개의 전문가 페르소나(임상가, 작업치료사, 코치, 경험자 기여자)를 시뮬레이션한다. Marina 의 보이스 패턴이 소수의 시드 샘플(현재 5 개)에서 추출된 후 시뮬레이션된 페르소나에 주입된다. 시스템은 임상 정확성을 위해 출력을 교차 검증하고 검증된 항목만 벨트에 병합한다. 약 30 개의 직접 큐레이션된 시드 입력은 보통 품질에서 약 4,000 개의 벨트 항목으로 증폭되며, 생산 품질에서는 10,000 개를 향해 올라간다.

2.2 Layer 2 — Diagnosis-Agnostic Core (runtime)

런타임 엔진은 28 개 알고리즘을 4 단계(LISTEN, KNOW, APPLY, SPEAK)로 거치며 각 사용자 메시지를 처리한다. 모든 28 개 알고리즘은 진단 비특이적이다; 그 동작은 활성 플러그인의 YAML 설정 파일에 의해 매개변수화된다. 교체를 지배하는 세 파일: registry.yaml 은 개별 알고리즘을 켜거나 끄고, book_library.yaml 은 지식 소스를 관리하고, plugin_loader 는 런타임에 장애별 벨트를 검색 풀에 병합한다.

2.3 Layer 3 — Vault Auto-Growth Loop (online learning)

각 사용자 대화 후, 네 개의 알고리즘(A10 CLI, A11 Counterfactual, A12 Cross-User Pattern Distiller, A13 Temporal Effect Tracker)이 벨트 개선을 위한 신호를 추출한다. 신호는 기존 벨트 항목의 가중치를 업데이트하고 인간 검토를 위한 새 항목을 제안한다. 언어모델 자체는 재훈련되지 않는다. 이것이 결정적이다: 변경되는 구성요소(벨트)가 인간 검토 가능하므로 임상 안전을 보장할 수 있다; LLM은 안정적으로 유지된다.

3. 4 단계 파이프라인

모든 사용자 메시지는 네 개의 순차 단계를 거쳐 처리된다.

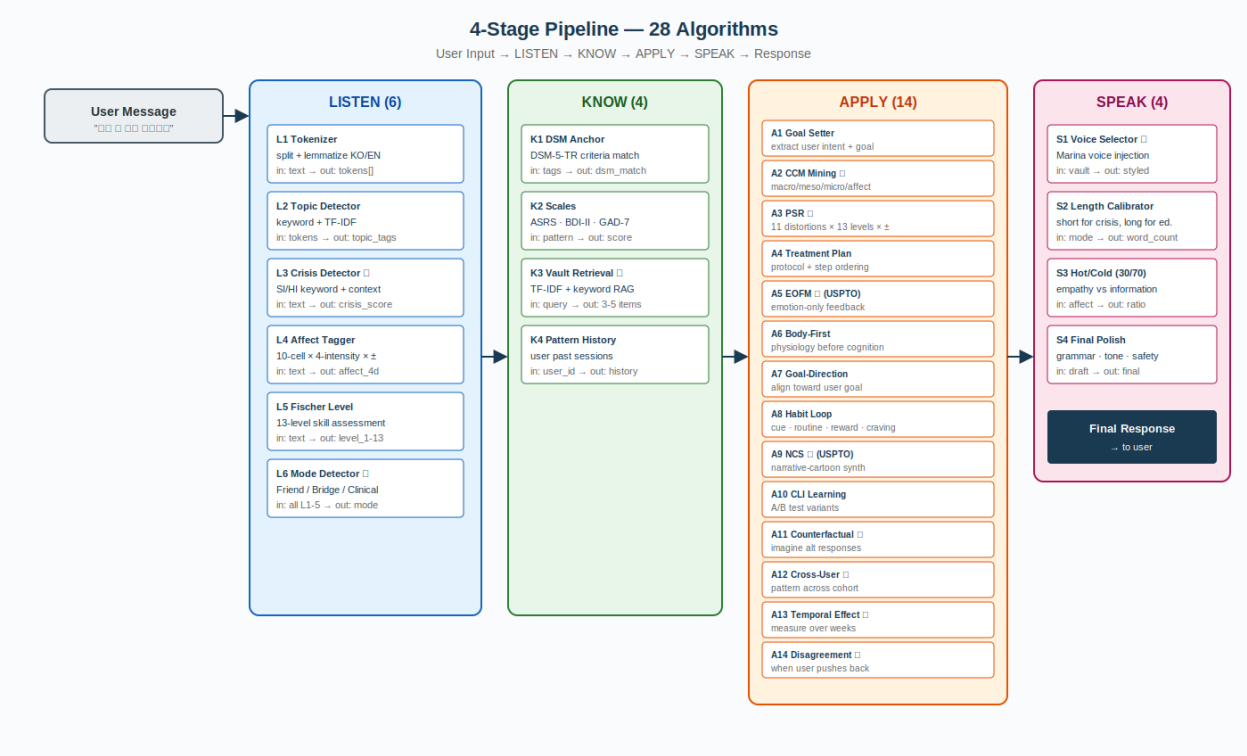

그림2. 28 개 알고리즘을 포함한 완전한4 단계 파이프라인.

LISTEN 은 입력을 파싱한다. KNOW 는 관련 지식을 검색한다. APPLY 는 개입을 선택하고 실행한다. SPEAK 는 응답을 생성하고 다듬는다. 각 단계는 정의된 입출력 계약을 가지며; 단계 내 알고리즘은 다른 단계에 영향을 미치지 않고 추가하거나 교체할 수 있다.

4. 28 개 알고리즘 상세 명세

이하 섹션은 28 개 알고리즘 각각을 다음 형식으로 명세한다: 요약, 입력, 출력, 의사코드, 임상적 근거.

4.1 LISTEN 단계

L1 — 토크나이저 (Tokenizer) [LISTEN] [V14 ✅]

한국어와 영어를 단일 토큰 스트림으로 처리하는 표제어(lemma) 기반 토크나이저. 한국어는 조사(은/는/이/가)와 어미(-다/-요/-습니다)를 제거하는 어간 추출기를 사용하며, 영어는 Porter 스템머(-ing/-ed/-s)를 사용한다.

입력 (Inputs)

- raw_text: string (한국어, 영어, 또는 혼합)

출력 (Outputs)

- tokens: Array<{lemma: string, lang: 'ko'|'en', pos?: string}>

의사코드 (Pseudocode)

```
function tokenize(text):
    lang = detect_language(text)             # ko / en / mixed
    if lang == 'ko':
        words = split_by_whitespace_and_punct(text)
        lemmas = [strip_particle_and_ending(w) for w in words]
    elif lang == 'en':
        words = split_by_whitespace(text.lower())
        lemmas = [porter_stem(w) for w in words]
    else:  # 혼합
        lemmas = lemmas_ko + lemmas_en
    return lemmas
```

임상적 근거 (Clinical Rationale)

ADHD 사용자는 코드 스위칭(언어 전환)과 부분 단어 입력이 빈번하다. K3 벨트 검색과 L2 토픽 감지에는 통일된 표제어 표현이 필요하다. 표제어 추출 없이는 '약속을'과 '약속이'가 서로 다른 토큰으로 인식되어 벨트 검색이 실패한다.

참고문헌: *Manning, C. & Schütze, H. (1999). Foundations of Statistical NLP.*

L2 — 토픽 감지기 (Topic Detector) [LISTEN] [V14 ✅]

10 개 토픽 라벨(work, family, romance, finance, health, school, friend, self, existential, somatic) 중 하나 이상으로 메시지를 태깅한다. 키워드 앵커와 사용자 최근 이력에 대한 TF-IDF 를 결합해 사용한다.

입력 (Inputs)

- tokens: TokenList
- user_recent_history: Array<TokenList> (최근 10 턴)

출력 (Outputs)

- topics: Array<{label: string, confidence: float}>

의사코드 (Pseudocode)

```
function topic_detect(tokens, history):
    keyword_hits = match_against_topic_anchors(tokens)
    tfidf_vec = compute_tfidf(tokens, history)
    candidates = []
    for topic in TOPIC_SET:
        score = 0.6 * keyword_hits[topic] + 0.4 * cosine(tfidf_vec,
topic_centroid[topic])
        if score > THRESHOLD: candidates.append((topic, score))
    return sort_desc(candidates)[:3]
```

임상적 근거 (Clinical Rationale)

같은 단어가 맥락에 따라 다른 의미를 갖는다 ('상사'는 직장; '옛 상사'는 실존적일 수 있음). 어휘 앵커와 TF-IDF 맥락을 결합하면 매 턴마다 비싼 LLM 호출을 하지 않고도 이를 처리할 수 있다.

참고문헌: *Blei, D. (2012). Probabilistic Topic Models. Comm. ACM.*

L3 — 위기 감지기 (Crisis Detector) [LISTEN] [V14 ✅ (항상 작동)]

모든 메시지에서 자살 사고, 자해, 타살 사고, 심각한 약물 위기, 아동 안전 관련 언어를 항상 스캔하는 안전 알고리즘. 두 단계: 키워드 정규식 + 맥락 명확화. 일반 파이프라인을 우회하고 988(미국) 또는 1393(한국) 등 지역 핫라인을 노출하는 라우팅 오버라이드를 발동시킨다.

입력 (Inputs)

- raw_text: string
- language_locale: string

출력 (Outputs)

- crisis_level: 0|1|2|3 (없음 / 수동적 / 능동적 / 임박)
- crisis_type: string
- route_override: bool

의사코드 (Pseudocode)

```
function crisis_detect(text, locale):
    matches = regex_match(text, CRISIS_PATTERNS[locale])
    if matches.empty: return (0, null, false)
    # 명확화: 'kill the deadline' vs 'kill myself'
    context_score = classify_intent(text, matches)
    level = score_to_level(context_score)
    if level >= 2:
        log_event(user_id, level, type, redacted=true)
        return (level, type, true)
    return (level, type, false)
```

임상적 근거 (Clinical Rationale)

범용 챗봇의 실패 양상은 능동적 위기 상황에서도 공감적 대화를 시도해 자원 연계 시점을 놓치는 것이다. 이 감지기는 다른 단계와 독립적으로 작동하므로 프롬프트 조작이나 LLM 환각에 의해 침묵될 수 없다.

참고문헌: *Posner, K. et al. (2011). C-SSRS. Am J Psychiatry, 168(12).*

L4 — 정서 태거 (Affect Tagger) [LISTEN] [V14 ✅]

메시지를 800-cell 정서 공간 — 4 차원 좌표 [Context 10] × [Alphabet A–J 정서 가족] × [Intensity 1–4] × [Polarity ±] — 으로 매핑한다. PSR 과 MEDA 의 하류 입력으로 사용된다.

입력 (Inputs)

- tokens: TokenList
- topics: L2 의 TopicList

출력 (Outputs)

- affect_cell: {context, alphabet, intensity, polarity}
- confidence: float

의사코드 (Pseudocode)

```
function affect_tag(tokens, topics):
    context = topics[0].label
    alphabet = classify_emotion_family(tokens)  # A..J
    intensity = score_intensity_markers(tokens)  # 1..4
    polarity = sign_from_lexicon(tokens)
    return {context, alphabet, intensity, polarity}
```

임상적 근거 (Clinical Rationale)

Russell 의 circumplex 는 2 차원이라 임상가가 필요로 하는 정보를 잃는다. 4D 800-cell 공간이 더 조밀하며 맥락을 보존한다 — 이는 맥락 변동성 자체가 장애인 ADHD 에서 결정적이다.

참고문헌: *Russell, J. (1980). Circumplex Model of Affect. JPSP, 39(6); Bradley & Lang (1999) ANEW.*

L5 — Fischer 레벨 (Fischer Level) [LISTEN] [V14 ✅]

Fischer 의 Dynamic Skill Theory 13 단계(감각운동 → 표상 → 추상 → 원리 체계) 척도로 인지발달 기술 수준을 평가한다. S2 의 응답 복잡도 조정에 사용된다.

입력 (Inputs)

- tokens: TokenList
- sentence_structure_features

출력 (Outputs)

- fischer_level: 1..13

의사코드 (Pseudocode)

```
function fischer_level(tokens, struct):
    abstraction = count_abstract_nouns(tokens) / len(tokens)
    hierarchy = parse_tree_depth(struct)
    integration = count_meta_terms(tokens)  # 'I notice that I...'
    raw = 0.4*abstraction + 0.3*hierarchy + 0.3*integration
    return map_to_level(raw)
```

임상적 근거 (Clinical Rationale)

Fischer 의 Dynamic Skill Theory(하버드 Kurt Fischer Lab)는 인지발달을 13 개의 재현 가능한 기술 수준으로 매핑한다. 사용자 현재 수준보다 높게 보정된 응답은 혼란이나 수치심을 유발하고, 낮게 보정된 응답은 무시당하는 느낌을 준다. 매칭이 응답이 '딱 맞다'고 느껴지게 만드는 요소다.

참고문헌: *Fischer, K. (1980). A theory of cognitive development. Psych Review, 87(6).*

L6 — 모드 감지기 ⭐ (Mode Detector) [LISTEN] [V14 ✅ (핵심)]

시스템이 진입할 세 가지 응답 모드 중 하나를 결정한다: 친구(Friend, 따뜻하게 함께함), 가교(Bridge, 감정을 다음 단계로 번역), 임상(Clinical, 구조화된 개입). L1–L5 출력을 종합한다.

입력 (Inputs)

- L1-L5 전체
- user_session_phase

출력 (Outputs)

- mode: 'friend' | 'bridge' | 'clinical'
- confidence: float

의사코드 (Pseudocode)

```
function mode_detect(L1, L2, L3, L4, L5, phase):
    if L3.crisis_level >= 2: return 'clinical'
    if L4.intensity >= 3 and L4.polarity == '-': return 'friend'
    if phase == 'goal_setting' or L1.has_action_request: return 'bridge'
    if L4.intensity <= 2 and phase == 'reflection': return 'friend'
    return 'bridge'
```

임상적 근거 (Clinical Rationale)

정신건강 챗봇의 가장 큰 실패는 레지스터 불일치 — 사용자가 들어주기를 원할 때 임상 구조를 제공하거나 사용자가 행동을 원할 때 공감만 제공하는 것이다. L6 는 어떤 내용도 생성되기 전에 레지스터를 선택하여 이를 방지한다.

참고문헌: *Bordin, E. (1979). The generalizability of the working alliance. Psychotherapy.*

4.2 KNOW 단계

K1 — DSM 앵커 (DSM Anchor) [KNOW] [V14 ✅]

감지된 패턴을 활성 플러그인 장애의 DSM-5-TR 진단 기준과 매칭한다. 진단을 내리지 않으며 — 다른 알고리즘이 사용하는 유사도 점수를 출력한다.

입력 (Inputs)

- L4.affect_cell
- L2.topics
- user_history

출력 (Outputs)

- dsm_match: Array<{criterion_id, similarity}>

의사코드 (Pseudocode)

```
function dsm_anchor(affect, topics, history):
    plugin_criteria = load_yaml('plugin.dsm.yaml')
    matches = []
    for crit in plugin_criteria:
        sim = jaccard(crit.markers, extract_markers(affect, topics, history))
        if sim > 0.2: matches.append((crit.id, sim))
    return matches
```

임상적 근거 (Clinical Rationale)

DSM-5-TR 에 앵커링하면 임상 검토자 전반에 걸쳐 개입의 타당성이 유지된다. 플러그인 이식성에 핵심적이다 — 같은 엔진, 장애별로 다른 기준 파일.

참고문헌: *American Psychiatric Association (2022). DSM-5-TR.*

K2 — 척도 (Scales) [KNOW] [V14 ✅]

검증된 임상 척도 점수(ADHD 의 ASRS, 우울증의 PHQ-9, 불안의 GAD-7, OCD 의 Y-BOCS, PTSD 의 PCL-5 등)를 유지한다. 단일 설문지로 한 번에 게이트하지 않고, 항목을 여러 세션에 분산시킨다.

입력 (Inputs)

- session_history
- active_plugin

출력 (Outputs)

- scale_scores: {name: current_value, items_completed, items_remaining}

의사코드 (Pseudocode)

```
function scales(history, plugin):
    scale_set = plugin.scales
    state = {}
    for scale in scale_set:
        items_seen = match_items_in_history(scale.items, history)
        state[scale.name] = {score: sum(items_seen.responses),
                             completed: len(items_seen),
                             remaining: scale.length - len(items_seen)}
    return state
```

임상적 근거 (Clinical Rationale)

사용자에게 임상 설문지를 강요하는 것은 대화를 깨뜨린다. 척도 항목을 자연스러운 대화 전반에 분산시키면 동맹 관계를 흐트러뜨리지 않으면서 동일한 데이터를 수집할 수 있다.

참고문헌: *Kessler, R. et al. (2005). ASRS validation. Psychol Med, 35.*

K3 — 벨트 검색 ⭐ (Vault Retrieval) [KNOW] [V14 ✅ → V15.1 ✅ (TF-IDF)]

벨트(활성 플러그인당 10,000+ 항목)를 검색하는 하이브리드 검색기. 표제어-키워드 매칭과 TF-IDF 코사인 유사도를 결합한다. 응답 후보 자료로 상위 K 개(기본 K=5) 항목을 반환한다.

입력 (Inputs)

- query_tokens
- topic_filter
- L6 의 mode_filter

출력 (Outputs)

- candidates: Array<{vault_id, score, item_body, voice_signal}>

의사코드 (Pseudocode)

```
function vault_retrieve(tokens, topic, mode):
    vault = load_active_plugin_vault()  # vault.jsonl + vault_ko.jsonl
    query_vec = tfidf(tokens, vault.idf_table)
    candidates = []
    for item in vault:
        if topic and topic not in item.topics: continue
        if mode and mode not in item.modes: continue
        kw = jaccard(tokens, item.tokens)
        cs = cosine(query_vec, item.tfidf)
        score = 0.4*kw + 0.6*cs
        candidates.append((item.id, score, item.body, item.voice))
    return top_k(candidates, K=5)
```

임상적 근거 (Clinical Rationale)

순수 LLM 응답은 일반적이며 임상 맥락에서 안전하지 않다. 임상가가 직접 저작한 큐레이션 벨트에 대한 RAG 가 TalkCatcher 의 목소리와 임상적 신뢰성을 제공한다. 해자는 LLM 이 아니라 벨트다.

참고문헌: *Lewis, P. et al. (2020). Retrieval-Augmented Generation. NeurIPS.*

K4 — 패턴 이력 (Pattern History) [KNOW] [V14 stub → V17]

사용자별 종단적 패턴 저장소: 어떤 개입이 효과 있었는지, 어떤 기분이 재발했는지, 하루 중 어떤 시간이 가장 힘든지. 기법 선택을 개인화하기 위해 APPLY 단계에서 읽는다.

입력 (Inputs)

- user_id

출력 (Outputs)

- history_summary: {effective_techniques, recurring_themes, time_patterns, prior_interventions}

의사코드 (Pseudocode)

```
function pattern_history(user_id):
    sessions = db.fetch_sessions(user_id, limit=50)
    eff = group_by_technique(sessions).filter(rating >= 3)
    themes = top_k_topics(sessions, k=5)
    time_pat = histogram(sessions, by='hour_of_day', filter='intensity>=3')
    prior = list_recent_interventions(sessions, n=10)
    return {eff, themes, time_pat, prior}
```

임상적 근거 (Clinical Rationale)

종단적 기억이 없으면 모든 대화가 0 에서 다시 시작한다. ADHD 사용자는 특히 시스템이 무엇을 시도했고 무엇이 잘 맞았는지 기억해주는 것에서 큰 이득을 본다; 그렇지 않으면 '나를 기억하지 못하는 또 다른 챗봇' 경험으로 퇴화한다.

참고문헌: *Bickmore, T. (2010). Health dialog systems. Patient Educ Counsel.*

4.3 APPLY 단계

A1 — 목표 설정자 (Goal Setter) [APPLY] [V14 ✅]

사용자의 이 턴 목표를 식별한다 — 토로 / 명료화 / 결정 / 계획 / 실행. 출력은 후속 알고리즘 선택을 결정한다.

입력 (Inputs)

- L1.tokens
- L2.topics
- L6.mode
- K4.history

출력 (Outputs)

- turn_goal: enum
- explicit: bool

의사코드 (Pseudocode)

```
function goal_setter(tokens, topics, mode, history):
    if mode == 'friend' and has_emotional_release(tokens): return ('vent',
explicit=false)
    if has_question_marker(tokens): return ('clarify', explicit=true)
    if has_choice_markers(tokens): return ('decide', explicit=true)
    if mode == 'bridge': return ('plan', explicit=false)
    return ('clarify', explicit=false)
```

임상적 근거 (Clinical Rationale)

사용자의 실제 목표와 시스템이 선택한 응답 전략 간 불일치가 단일 최대 불만족 요인이다. A1 은 사용자가 단지 토로하고 싶을 때 시스템이 계획을 제공하는 것을 방지한다.

A2 — CCM 마이닝 ⭐ (Context-Cascaded Mining) [APPLY] [V14 ✅]

맥락 단계 마이닝(Pillar 1). 네 개의 중첩 스케일로 맥락을 추출한다: 거시(생애 상황), 중간(이번 주/이번 달), 미시(오늘의 사건), 정서(지금 이 순간). 각 레벨이 다음을 제약한다.

입력 (Inputs)

- session_history
- L4.affect
- L2.topics

출력 (Outputs)

- context: {macro, meso, micro, affect}

의사코드 (Pseudocode)

```
function ccm_mine(history, affect, topics):
    macro = extract_long_term_themes(history, window='90d')
    meso = extract_active_stressors(history, window='30d')
    micro = extract_current_event(history.last_turn)
    return {macro, meso, micro, affect}
```

임상적 근거 (Clinical Rationale)

범용 챗봇은 각 메시지를 고립된 것으로 다룬다. 인간 치료사는 네 개의 스케일을 동시에 보유한다: '이 순간, 이번 주, 이 계절, 이 인생'. CCM 은 동일하게 작동하며 — 어떤 개입이 적절한지를 극적으로 바꾼다.

참고문헌: *Marina (founder, BCN, PhD), 임상 큐레이션.*

A3 — PSR ⭐ (Polarity-Selective Replacement) [APPLY] [V14 ✅ (영업비밀)]

극성-선택적 대체(Pillar 2). 429-셀 매핑 표: 11 개 인지 왜곡 × 13 Fischer 레벨 × 3 극성 구간. 사용자의 정확한 레벨에 맞춘 대체 사고를 선택한다.

입력 (Inputs)

- A2.context
- L4.affect_cell
- L5.fischer_level

출력 (Outputs)

- replacement_offer: {distortion, original_pattern, suggested_reframe, level_match}

의사코드 (Pseudocode)

```
function psr(context, affect, level):
    distortion = classify_distortion(context, affect)  # 11개 중 1개
    polarity = affect.polarity
    key = (distortion, level, polarity)
    reframe = PSR_TABLE[key]  # Marina 큐레이션
    return {distortion, pattern=context.micro, reframe, level_match=true}
```

임상적 근거 (Clinical Rationale)

표준 CBT 는 왜곡당 하나의 재구성을 제공한다. PSR 은 13 개를 제공한다 — 기술 수준에 맞춰 보정된. Level-9 사용자에게 Level-3 재구성은 자칫 거들먹거리는 것이고, Level-3 사용자에게 Level-9 재구성은 이해 불가다. 429-셀 표가 그 간극을 메운다.

참고문헌: *Beck, A. (1976). Cognitive Therapy and the Emotional Disorders; Fischer (1980).*

A4 — 치료 계획 (Treatment Plan) [APPLY] [V14 stub → V17]

치료 호(보통 8–12 주)에 걸쳐 개입을 순서화한다. 플러그인별 프로토콜을 로드하고 현재 주차에 적절한 기법을 선택한다.

입력 (Inputs)

- plugin.protocol

- user_progress_state
- K4.history

출력 (Outputs)

- plan: {current_phase, current_step, next_steps}

의사코드 (Pseudocode)

```
function treatment_plan(protocol, progress, history):
    phase = progress.phase  # 예: 'psychoed' / 'skill' / 'consolidation'
    step = progress.step_in_phase
    candidate_steps = protocol[phase][step:step+3]
    filtered = remove_recently_used(candidate_steps, history)
    return {phase, step, next=filtered}
```

임상적 근거 (Clinical Rationale)

무작위 개입 선택은 도구상자처럼 느껴지지 도 치료처럼 느껴지지 않는다. 순서화된 계획은 기술을 누적적으로 쌓아 올리며, 각 주가 이전 주에 의지한다.

참고문헌: *Beck, J. (2011). Cognitive Behavior Therapy, 2nd ed.*

A5 — EOFM ⭐ (USPTO) [APPLY] [V14 stub → V18]

정서 단독 피드백 모드(Pillar 4). 순수하게 정서 신호(HRV, 음성 운율, 응답 지연)에만 기반하여 세션을 조정하는 세계 최초 시스템 — 언어적/인지적 부담을 우회한다. 알렉시티미아 사용자에게 결정적이다.

입력 (Inputs)

- L4.affect
- biosignal_stream (선택)
- response_latency

출력 (Outputs)

- session_adjust: {speed, depth, register, pause_offer}

의사코드 (Pseudocode)

```
function eofm(affect, bio, latency):
    if bio.hrv_lf_hf > THRESH_AUTONOMIC: speed='slow'
    if latency > THRESH_OVERLOAD: depth='surface'
    if affect.intensity >= 3: pause_offer=true
    register = derive_register_from_affect(affect)
    return {speed, depth, register, pause_offer}
```

임상적 근거 (Clinical Rationale)

많은 ADHD 와 트라우마 이력 사용자는 정서를 언어로 표현하는 데 어려움을 겪는다(알렉시티미아). 감정 이름을 요구하는 것 자체가 장벽이다. EOFM 은 세션이 단어가 아닌 몸과 함께 호흡하도록 한다.

참고문헌: *Taylor, G. (1997). Disorders of Affect Regulation.*

A6 — 신체 우선 (Body-First) [APPLY] [V14 stub → V18]

정서 강도가 인지 대역폭을 초과할 때, 어떤 언어적/인지적 개입보다 먼저 생리적 그라운딩으로 세션을 라우팅한다. 다미주신경(polyvagal) 정보에 기반한 순서화를 구현한다.

입력 (Inputs)

- L4.affect
- L6.mode

출력 (Outputs)

- grounding_protocol: {type, duration_sec, instruction_text}

의사코드 (Pseudocode)

```
function body_first(affect, mode):
    if affect.intensity < 3 or mode == 'clinical': return null
    protocols = ['orient_5senses', 'cold_water', 'box_breathing', 'feet_floor']
    chosen = select_by_context(affect.context, protocols)
    return {type: chosen, duration_sec: 60, instruction: load_text(chosen)}
```

임상적 근거 (Clinical Rationale)

다미주신경 이론과 수십 년의 트라우마 연구는 정서가 높을 때 인지 재구성만으로는 ventral-vagal 활성화를 달성할 수 없음을 확인한다. 신체 우선은 강도 3 이상에서 생리가 인지를 선행해야 한다는 점을 존중한다.

참고문헌: *Porges, S. (2011). The Polyvagal Theory.*

A7 — 목표 방향 (Goal-Direction) [APPLY] [V14 ✅]

선택된 기법을 사용자의 명시된 목표 방향과 정렬시킨다. 시스템이 목표 외 개입(예: 사용자가 계획을 원할 때 정서 처리)으로 표류하는 것을 방지한다.

입력 (Inputs)

- A1.turn_goal
- user.long_term_goal
- candidate_interventions

출력 (Outputs)

- filtered_interventions: Array<intervention>

의사코드 (Pseudocode)

```
function goal_direction(turn_goal, long_term, candidates):
    filtered = []
    for c in candidates:
        if c.serves(turn_goal) and c.compatible_with(long_term):
            filtered.append(c)
    return filtered
```

임상적 근거 (Clinical Rationale)

명시적 목표 정렬이 없으면 AI 시스템은 학습 데이터에 정서 언어가 가장 풍부하기 때문에 정서 처리 쪽으로 표류한다. A7 은 이러한 표류를 막는다.

A8 — 습관 고리 (Habit Loop) [APPLY] [V14 stub → V19]

네 가지 습관 고리 요소(단서, 루틴, 보상, 갈망)를 사용자 하루의 시간 정렬된 순간에 매핑한다. 어떤 행동 변화 개입에도 사용된다.

입력 (Inputs)

- A2.context
- user_daily_log

출력 (Outputs)

- habit_map: {cue, routine, reward, craving, time_anchor}

의사코드 (Pseudocode)

```
function habit_loop(context, daily):
    target = context.micro.target_behavior
    cue = identify_antecedent(daily, target)
    routine = describe(target)
    reward = identify_reinforcer(daily, target)
    craving = infer_underlying_need(reward)
    time = anchor_to_day(daily, cue)
    return {cue, routine, reward, craving, time_anchor: time}
```

임상적 근거 (Clinical Rationale)

네 가지 고리 요소 전체를 포함하지 않는 행동 변화 개입은 유지율이 나쁘다. 시간 앵커링은 ADHD 사용자가 실제로 개입을 수행하게 만드는 핵심이다 — 모호한 시점은 순응도를 죽인다.

참고문헌: *Duhigg, C. (2012). The Power of Habit; Wood & Neal (2007), Psych Review.*

A9 — NCS ⭐ (USPTO) [APPLY] [V14 stub → V19]

내러티브-카툰 합성기(Pillar 5). Freytag 의 5 막 구조를 따르는 SVG 카툰을 생성하며, 인지 편향 수정-해석(CBM-I) 내용을 내장한다.

입력 (Inputs)

- A3.replacement_offer
- user.preferred_character_style

출력 (Outputs)

- svg_cartoon: string
- cbm_i_target: string

의사코드 (Pseudocode)

```
function ncs(reframe, style):
    arc = build_freytag_arc(reframe)   # 5 막
    char = render_character(style, pose=affect_to_pose(reframe.affect))
    panels = render_panels(arc, char)
    cbm_i_target = reframe.suggested_reframe
    return {svg: panels_to_svg(panels), cbm_i_target}
```

임상적 근거 (Clinical Rationale)

텍스트 중심 심리교육은 ADHD 인구에서 회상률이 낮다. 내러티브 카툰을 통해 전달된 CBM-I 는 이중 부호화(시각 + 언어)를 증가시키고, 회상을 개선하며, 읽기 부담 장벽을 낮춘다.

참고문헌: *Hertel & Mathews (2011). Cognitive Bias Modification, Perspect Psychol Sci.*

A10 — CLI 학습 (Continuous Learning) [APPLY] [V14 ✅ (Pillar 7)]

지속적 학습 인프라(Pillar 7). 동의를 받은 실제 사용자에 대해 응답 변형을 A/B 테스트한다. 벨트 항목 가중치에 대한 그래디언트 업데이트는 임상 안전 필터로 게이트된다.

입력 (Inputs)

- K3 의 response_candidates
- user_feedback_signal

출력 (Outputs)

- selected_variant: vault_id
- weight_update: Map<vault_id, delta>

의사코드 (Pseudocode)

```
function cli_learn(candidates, feedback):
    if random() < EXPLORE_RATE: chosen = random_pick(candidates)
    else: chosen = argmax(c.score for c in candidates)
    if feedback received:
        delta = compute_gradient(chosen.id, feedback.rating)
        # 안전 점검: 검토자가 안전하지 않다고 표시한 경우 절대 가중치 증가 금지
        if not safety_flag(chosen.id):
            vault.weight[chosen.id] += delta * LR
    return chosen, weight_update
```

임상적 근거 (Clinical Rationale)

정적 프롬프트는 품질에서 정체된다. CLI 는 LLM 을 재훈련하지 않고도 실세계 신호로부터 시스템을 개선시킨다 — 비용을 낮게 유지하고 임상 감독을 단단하게 유지한다(학습하는 것은 모델이 아닌 벨트이므로).

참고문헌: *Sutton & Barto (2018). Reinforcement Learning, 2nd ed.*

A11 — 반사실 생성 ⭐ 🆕 (Counterfactual) [APPLY] [V15.1 ✅]

생성된 각 응답에 대해 시스템은 조용히 3 개의 대안 응답 전략을 생성하고, 각각을 대리 결과 모델로 점수화하며, 델타를 K3 와 A10 의 학습 데이터로 저장한다. 사실상 턴당 4 배의 학습 신호.

입력 (Inputs)

- selected_response
- context
- K3.candidates

출력 (Outputs)

- counterfactual_set: Array<{alt, predicted_outcome, regret}>

의사코드 (Pseudocode)

```
function counterfactual(chosen, ctx, candidates):
    alternatives = sample_top_k(candidates, k=3, exclude=chosen)
    predicted = [proxy_outcome_model(alt, ctx) for alt in alternatives]
```

```
    chosen_pred = proxy_outcome_model(chosen, ctx)
    regrets = [p - chosen_pred for p in predicted]
    log_for_training(chosen, alternatives, predicted, regrets)
    return {alternatives, predicted, regrets}
```

임상적 근거 (Clinical Rationale)

표준 A/B 테스트는 사용자당 턴당 1 개 변형에서 학습한다. 반사실 생성은 시스템이 턴당 4 개의 암묵적 변형에서 학습하게 한다 — 거의 0 에 가까운 한계 비용으로 4 배의 신호.

참고문헌: *Pearl, J. (2009). Causality, 2nd ed. (반사실 프레임워크).*

A12 — 교차 사용자 패턴 추출 🆕 (Cross-User Distiller) [APPLY] [V15 stub → V17]

PII 제거 후, 여러 사용자에 걸쳐 반복되는 패턴을 식별한다(예: '파트너 대화 주변의 시간 맹점이 사례의 73%에서 수치심을 유발'). 출력은 새 벨트 항목으로 흘러들어가며, 모델 가중치로는 절대 흘러가지 않는다.

입력 (Inputs)

- anonymized_session_summaries (n>200)

출력 (Outputs)

- distilled_patterns: Array<{pattern_signature, frequency, suggested_vault_entries}>

의사코드 (Pseudocode)

```
function cross_user_distill(summaries):
    stripped = [strip_pii(s) for s in summaries]
    clusters = topic_cluster(stripped, min_size=20)
    patterns = []
    for cl in clusters:
        sig = extract_signature(cl)
        freq = len(cl) / len(stripped)
        vault_suggestions = generate_candidate_entries(sig)
        patterns.append((sig, freq, vault_suggestions))
    return patterns
    # 벨트 병합 전 인간 검토 필수
```

임상적 근거 (Clinical Rationale)

집단 수준 패턴은 단일 사용자 수준에서는 보이지 않는다 . A12 는 프라이버시를 보존하면서 이를 표면화한다. 출력은 벨트 병합 전에 항상 인간 검토를 거친다.

참고문헌: *Dwork & Roth (2014). Algorithmic Foundations of Differential Privacy.*

A13 — 시간적 효과 추적 🆕 (Temporal Effect) [APPLY] [V15 stub → V18 (USPTO 후보)]

단일 턴이 아닌 주(weeks) 단위로 개입의 잠재 효과를 측정한다. 기법이 지속적 안도를 주었는지 일시적 안도를 주었는지 감지한다. 더 긴 기간에 걸쳐 K3 검색을 가중하는 데 사용된다.

입력 (Inputs)

- user.intervention_log

- user.symptom_self_report_series

출력 (Outputs)

- effect_estimates: Map<intervention_id, {1wk_effect, 4wk_effect, decay_rate}>

의사코드 (Pseudocode)

```
function temporal_effect(log, series):
    estimates = {}
    for iv in log.unique_interventions:
        windows = align_pre_post(iv, series, windows=[7, 14, 28])
        eff_1wk = mean(post[0:7]) - mean(pre[-7:0])
        eff_4wk = mean(post[21:28]) - mean(pre[-7:0])
        decay = (eff_1wk - eff_4wk) / max(eff_1wk, 1e-6)
        estimates[iv.id] = (eff_1wk, eff_4wk, decay)
    return estimates
```

임상적 근거 (Clinical Rationale)

단기 만족도는 종종 지속적 임상 이득과 갈라진다. A13 은 시스템을 정직하게 유지한다: 즉각적으로 기분 좋게 만들지만 4 주 이득이 없는 기법은 가중치가 낮춰진다.

참고문헌: *Kazdin, A. (2017). Single-Case Research Designs.*

A14 — 불일치 감지 🆕 (Disagreement Detector) [APPLY] [V15 stub → V18]

사용자가 시스템의 프레이밍을 거부할 때 감지한다. 불일치를 실패가 아닌 데이터로 다룬다 — 많은 임상적 돌파가 정확히 불일치 지점에서 일어난다.

입력 (Inputs)

- user.recent_messages
- system.recent_responses

출력 (Outputs)

- disagreement_signal: {present, type, severity}

의사코드 (Pseudocode)

```
function disagreement(user_msgs, sys_resps):
    if not any_negation_marker(user_msgs[-1]): return {present: false}
    type = classify_disagreement(user_msgs[-1], sys_resps[-1])
    severity = score_intensity(user_msgs[-1])
    return {present: true, type, severity}
```

임상적 근거 (Clinical Rationale)

표준 챗봇은 거부를 실패로 해석하고 사과한다. 임상적으로 거부는 더 깊은 작업에 대한 준비도를 신호하는 경우가 많다. A14 는 시스템이 안심시키기로 무너지지 않고 불일치와 머무를 수 있게 한다.

참고문헌: *Safran & Muran (2000). Negotiating the Therapeutic Alliance.*

4.4 SPEAK 단계

S1 — 보이스 선택 ⭐ (Voice Selector) [SPEAK] [V14 ✅ → V16 ✅ (Marina voice)]

검색된 벨트 콘텐츠에 Marina 의 보이스 시그니처를 주입한다. 추출된 스타일 특성에는 운율, 비유 밀도, 시작 문구, 따뜻함 마커가 포함된다.

입력 (Inputs)

- K3.candidates
- marina.voice_pattern
- L6.mode

출력 (Outputs)

- styled_draft: string

의사코드 (Pseudocode)

```
function voice_select(candidates, pattern, mode):
    base = candidates[0].body
    styled = apply_voice_pattern(base, pattern, mode)
    # voice_pattern = {opener_options, metaphor_examples, cadence,
warmth_markers}
    return styled
```

임상적 근거 (Clinical Rationale)

보이스는 복제할 수 없는 해자다. 경쟁자가 벨트를 본다 해도 특정 임상 보이스는 복제할 수 없다. S1 은 모든 응답이 그 시그니처를 지니도록 보장한다.

S2 — 길이 보정 (Length Calibrator) [SPEAK] [V14 ✅]

모드와 정서 강도에 기반해 목표 응답 길이를 설정한다. 위기에서는 짧게, 심리교육에서는 길게.

입력 (Inputs)

- L6.mode
- L4.affect.intensity
- user.length_preference

출력 (Outputs)

- word_count_target: int

의사코드 (Pseudocode)

```
function length_calibrate(mode, intensity, pref):
    if mode == 'clinical' and intensity >= 3: return 30
    if mode == 'friend' and intensity >= 3: return 50
    if mode == 'bridge': return 120
    return pref.default or 80
```

임상적 근거 (Clinical Rationale)

정서가 높은 순간의 긴 응답은 시스템이 듣고 있지 않은 것처럼 느껴진다. 심리교육 중의 짧은 응답은 불충분하게 느껴진다. 길이는 내용만큼 중요하다.

S3 — Hot/Cold (30/70) [SPEAK] [V14 stub → V19]

공감 콘텐츠('hot') 대 정보/구조 콘텐츠('cold')의 비율을 설정한다. 모드가 다르게 지시하지 않는 한 기본값은 30/70.

입력 (Inputs)

- L6.mode
- L4.affect

출력 (Outputs)

- mix: {hot: float, cold: float}

의사코드 (Pseudocode)

```
function hot_cold(mode, affect):
    if mode == 'friend': return {hot: 0.7, cold: 0.3}
    if mode == 'clinical' and affect.intensity >= 3: return {hot: 0.5, cold: 0.5}
    if mode == 'bridge': return {hot: 0.3, cold: 0.7}
    return {hot: 0.3, cold: 0.7}
```

임상적 근거 (Clinical Rationale)

순수 정보는 차갑게 느껴진다; 순수 공감은 쓸모없게 느껴진다. 30/70 기본값은 검증과 행동 양쪽을 원하는 성인 ADHD 인구에 경험적으로 sweet spot 이다.

S4 — 최종 마감 (Final Polish) [SPEAK] [V14 ✅]

출력 전 문법, 톤, 안전 검사를 실행한다. 금지 문구(창립자 규칙에 따른 rest/break 등)를 플래깅하고, 위기 감지기 오버라이드 상태를 확인하며, 최종 양식 보정을 적용한다.

입력 (Inputs)

- styled_draft
- S2.length_target
- L3.crisis_status

출력 (Outputs)

- final_response: string

의사코드 (Pseudocode)

```
function final_polish(draft, target_len, crisis):
    if crisis.level >= 2: return crisis_response_template(crisis)
    cleaned = remove_banned_phrases(draft)
    trimmed = adjust_length(cleaned, target_len)
    polished = grammar_check(trimmed)
    return polished
```

임상적 근거 (Clinical Rationale)

출력 전 마지막 방어선. 표류, 길이 초과를 잡아내고 안전 오버라이드가 존중되도록 보장한다. S4 없이는 파이프라인의 어떤 오류든 사용자에게 도달한다.

5. 7 Pillars 임상 IP

Pillar 는 TalkCatcher 를 범용 대화형 AI 와 구별시키는 독점 임상 자산이다. 각 Pillar 는 알고리즘, 매핑 표, 그리고 Marina 가 저작한 임상 가중치의 조합이다.

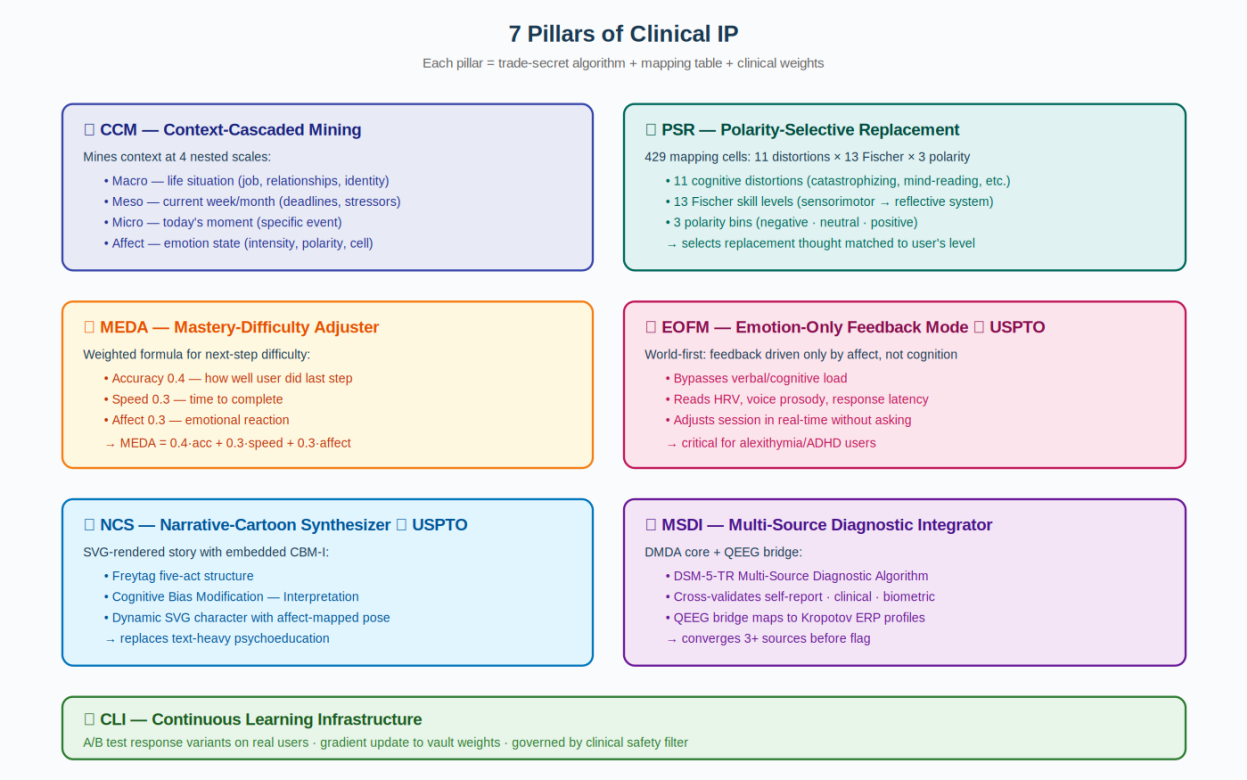

그림3. 7 개의 임상IP Pillar.

① CCM — Context-Cascaded Mining

네 개의 중첩 스케일(macro/meso/micro/affect)에서 맥락을 마이닝한다. 알고리즘 A2 로 구현. 차별화: 범용 챗봇이 할 수 없는 네 개의 시간 스케일을 동시에 보유한다.

```
context = {macro: f_macro(history_90d), meso: f_meso(history_30d), micro:
f_micro(last_turn), affect: L4.cell}
```

② PSR — Polarity-Selective Replacement

429-셀 매핑 표(11 왜곡 × 13 Fischer 레벨 × 3 극성 구간). A3 로 구현. 핵심: 재구성을 사용자의 발달 기술 수준에 맞춘다.

```
reframe = PSR_TABLE[(distortion_11, level_13, polarity_3)]   # 429 cells, 영업비밀
```

③ MEDA — Mastery-Difficulty Adjuster

다음 단계 난이도를 계산하는 가중 공식. 등급화된 순서가 필요한 곳마다 사용(노출, 행동 활성화, 습관 사다리).

```
MEDA = 0.4·accuracy + 0.3·speed + 0.3·affect_reaction
```

④ EOFM — Emotion-Only Feedback Mode (USPTO)

세계 최초: 언어/인지 부담을 우회하고 순수하게 정서 신호에 기반해 세션을 조정. A5 로 구현. 알렉시티미아 인구에 결정적.

```
session_adjust = f(HRV_lf_hf, voice_prosody, response_latency)
```

⑤ NCS — Narrative-Cartoon Synthesizer (USPTO)

Freytag 의 5 막 구조를 따르는 SVG 렌더링 카툰에 CBM-I 내장. A9 로 구현.

```
cartoon = Freytag_5act(arc) ∘ render_character(pose=affect_to_pose(affect))
```

⑥ MSDI — Multi-Source Diagnostic Integrator

DMDA 코어 + QEEG 브릿지. 어떤 진단 플래그 이전에 자기 보고, 임상 관찰, 그리고 가능한 경우 생체 데이터를 교차 검증. K1 과 K2 로 구현.

```
diagnostic_score = converge(self_report, clinical_obs, biometric)   # ≥3 sources
required
```

⑦ CLI — Continuous Learning Infrastructure

실제 사용자에게 응답 변형을 A/B 테스트하고, 벨트 가중치를 업데이트하며, 임상 안전 필터로 게이트. A10 으로 구현.

```
vault.weight[id] += LR · gradient(feedback, response_id)   # gated by safety_flag
```

6. 800-Cell 정서 공간

800-cell 공간은 정서 태깅을 위한 4 차원 이산 좌표계다. PSR 매핑 표(Pillar 2)의 입력 공간으로, 그리고 Mastery-Difficulty Adjuster(Pillar 3)의 상태 공간으로 사용된다.

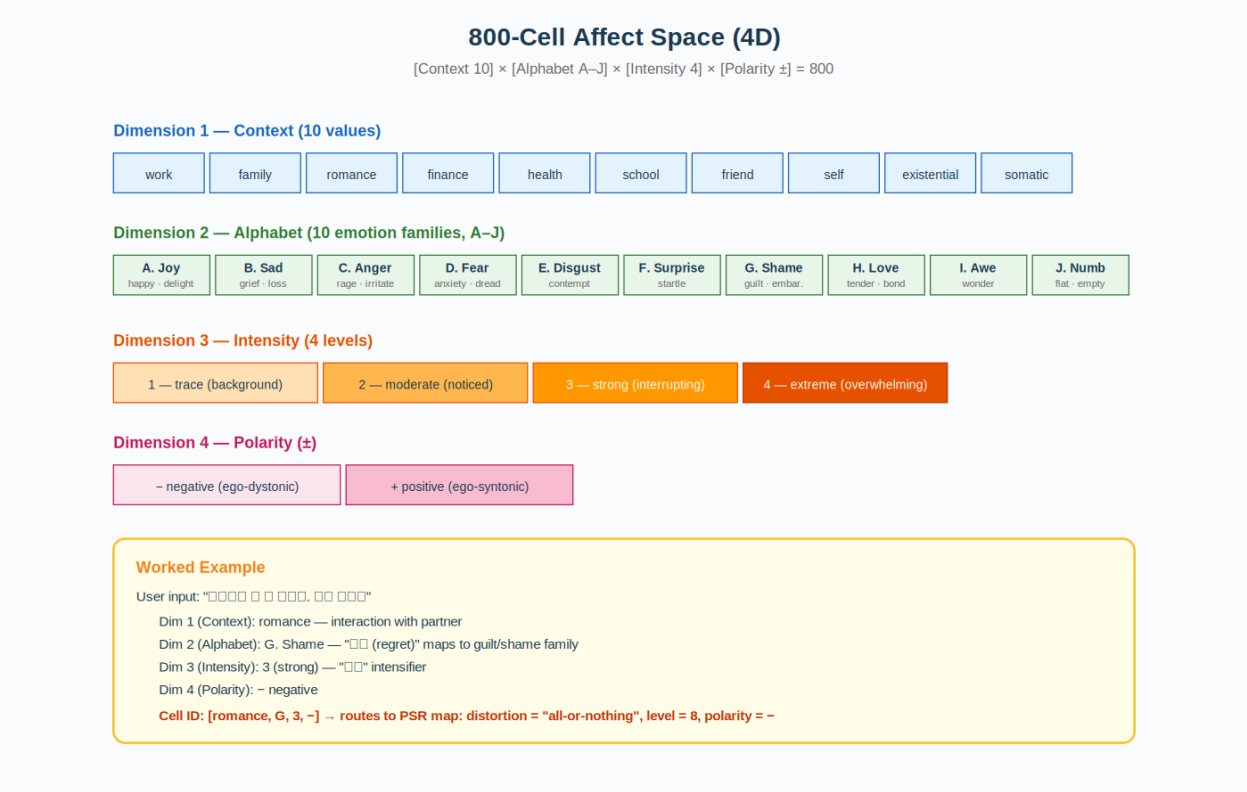

그림 4. 800-Cell 정서 공간과 실제 예시.

6.1 차원

- Context (10 값): work, family, romance, finance, health, school, friend, self, existential, somatic.
- Alphabet (10 개 정서 가족, A–J): Joy, Sad, Anger, Fear, Disgust, Surprise, Shame, Love, Awe, Numb.
- Intensity (4 단계): trace, moderate, strong, extreme.
- Polarity (2 부호): 음성(ego-dystonic)과 양성(ego-syntonic).

총: 10 × 10 × 4 × 2 = 800 cells.

6.2 Russell 의 Circumplex 를 사용하지 않는 이유

Russell 의 고전 circumplex 는 2 차원(valence × arousal)이다. 중요한 임상적 구별을 무너뜨린다 — 예를 들어, 수치심과 슬픔은 circumplex 영역에서 유사한 위치를 차지하지만 매우 다른 개입을 요구한다. 800-cell 공간은 임상가들이 실제로 사용하는 구별을 보존한다.

7. Synthetic Vault Generation

단일 임상가가 10,000 개 항목의 벨트를 손으로 구축하는 것은 비현실적이다. 합성 생성기는 다섯 단계를 통해 소규모 시드 샘플을 증폭한다.

7.1 Stage 1 — Expert Persona Definition

30 개 페르소나가 expert_personas/<plugin>.yaml 에 정의된다. 각 페르소나는 임상 전문 분야(예: 임원진 대상 ADHD 코치, 작업치료사, 가족 시스템 관점), 글쓰기 스타일, 허용 가능한 기법 목록을 가진다.

7.2 Stage 2 — Voice Style Extraction

Marina 가 저작한 소규모 시드 샘플(현재 5 개)로부터 보이스 패턴이 추출된다: 시작 문구, 비유 밀도, 문장 운율, 따뜻함 마커, 일반적 전환구.

7.3 Stage 3 — Generation

각 (페르소나 × 맥락) 쌍에 대해 LLM 이 후보 벨트 항목을 생성한다. 페르소나 전반에 걸친 양식 일관성을 유지하기 위해 Marina 의 보이스 패턴이 각 항목에 주입된다.

7.4 Stage 4 — Cross-Validation

각 후보 항목은 임상 정확성, 안전(금지 문구 없음, OCD 플러그인에 대해 안심시키기 없음 등), 활성 플러그인 프로토콜과의 일관성을 점검하기 위해 다른 페르소나로 재프롬프팅하여 검증된다. 검증 실패 항목은 폐기된다.

7.5 Stage 5 — Merge

검증된 항목은 메타데이터(주제, 모드, 극성, 강도 범위, 소스 페르소나)와 함께 벨트에 병합된다. 벨트는 그 후 TF-IDF 검색을 위해 재인덱싱된다.

8. Plugin Architecture and Scale-Out

플러그인 아키텍처는 단일 창립자에게 프로젝트를 확장 가능하게 만든다.

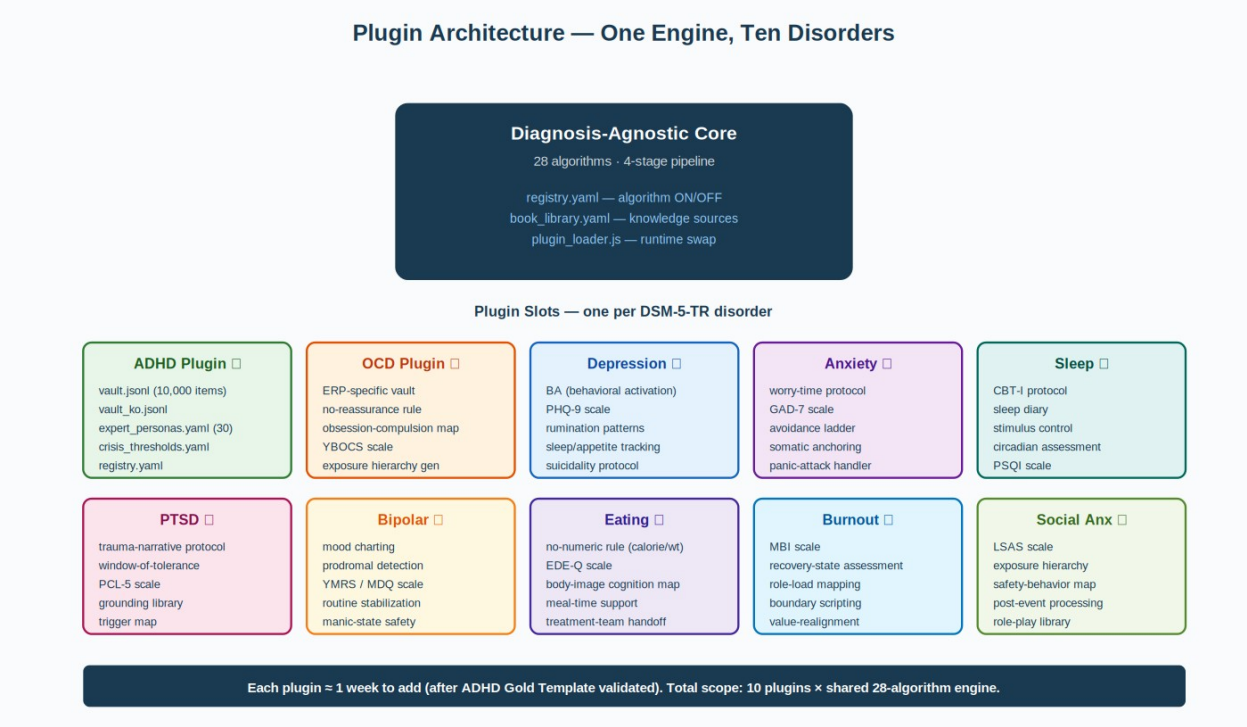

그림5. 10 개DSM-5-TR 조건에 걸친 플러그인 아키텍처.

8.1 Plugin Manifest

각 플러그인은 plugins/<disorder>/ 아래에 위치하며 다음을 포함한다:

- vault.jsonl — 주 영어 벨트
- vault_<locale>.jsonl — 지역화된 벨트(예: 한국어의 경우 vault_ko.jsonl)
- expert_personas.yaml — 스타일과 전문분야가 포함된 페르소나 목록
- dsm.yaml — 관련 DSM-5-TR 기준
- scales.yaml — 대화 전반에 분산된 임상 척도 항목
- registry.yaml — 이 플러그인의 알고리즘 on/off 플래그
- protocol.yaml — 순서화된 개입 계획(주간 호)
- crisis_thresholds.yaml — 장애별 안전 임계값

8.2 Replication Plan

ADHD 가 검증 템플릿이다. ADHD 플러그인이 임상적으로 검증되면, OCD, 우울증, 불안, 수면, PTSD, 양극성, 섭식장애, 번아웃, 사회불안에 동일한 스캐폴드가 적용된다. 각 플러그인은 1 주의 저작 작업과 임상 검토로 추정된다.

9. USPTO Patent Portfolio

8 개의 임시 특허 명세서가 가장 새로운 구성요소를 다룬다.

#	제목	핵심 청구항	Pillar / Algorithm
1	Fischer DSL — Dynamic Skill Language Mapping	사용자 언어를 Fischer 13 단계 기술 척도에 매핑	Pillar 지원 — A3, L5
2	QEEG-Conv — QEEG to Conversational State Mapping	Kropotov ERP 프로파일을 챗봇 상태에 매핑	MSDI 브릿지 지원
3	Temporal Congruence	시간 전반에 걸쳐 cue/routine/reward/craving 정렬	A8 Habit Loop 지원
4	DMDA — DSM-5-TR Multi-Source Diagnostic Algorithm	진단 플래그 이전 3+ 소스 수렴	Pillar 6 MSDI; 대표 작품
5	CCM + MEDA Combined Specification	Pillar 1 과 3 의 결합 명세	Pillars 1 + 3
6	EOFM — Emotion-Only Feedback Mode	정서 신호만으로 세션 조정	Pillar 4; A5
7	NCS — Narrative-Cartoon Synthesizer	SVG 카툰 + Freytag + CBM-I	Pillar 5; A9
8	A11 Counterfactual Generation	학습을 위한 턴당 대안 응답 생성	A11; 신규

10. Validation Status and Roadmap

10.1 현재 빌드 상태

2026 년 5 월 13 일 기준:

- V14: 24 개 알고리즘 스캐폴딩, 12/12 단위 테스트 통과.
- V15.1: 시맨틱 검색과 합성 벨트 생성기 추가; A11 Counterfactual 구현. 17/17 테스트 통과.
- V16: Voice 추출기, 전문가 페르소나 라이브러리(ADHD 용 30 개 페르소나), Admin UI, 한국어 벨트 시드(10 개 항목), 통합 i18n. 22/22 테스트 통과. Marina 저작 보이스 샘플: 5 개.
- 벨트 총량(V16): 20 개 항목(시드). ADHD V20 목표: 10,000 개 항목.

10.2 로드맵

- V17 — talkcatcher.com 통합, GitHub/S3 배포, Supabase 영속화, A12 Cross-User Pattern Distiller.
- V18 — A5 EOFM, A13 Temporal Effect Tracker, A14 Disagreement Detector.
- V19 — A8 Habit Loop, A9 NCS (Narrative-Cartoon Synthesizer), S3 Hot/Cold 비율.
- V20 — Auto-Growth Loop 완전 통합; 28 개 알고리즘 모두 프로덕션.
- Post-V20 — ADHD 임상 검증; 그 다음 9 개 추가 장애로의 복제, 장애당 1 주의 플러그인 저작 추정.

11. 안전·윤리·한계

11.1 안전 아키텍처

위기 감지(L3)는 LLM 과 독립적으로 작동한다. 결정론적 정규식과 작은 의도 분류기로 구현된다. LLM 이 침묵하거나 조작되더라도 L3 는 여전히 응급 자원으로 라우팅한다.

11.2 실무 범위

TalkCatcher 는 비의사 BCN+PhD 실무 범위 내에서 작동한다. 진단하거나 처방하거나 임상적 돌봄을 대체하지 않는다. 마케팅 언어는 수행 최적화와 자기 탐색 프레이밍에 맞춰져 있다; 진단 및 치료 언어는 면허 있는 맥락에 한정된다.

11.3 프라이버시

사용자 수준 데이터는 사용자 유래 키로 암호화되어 저장된다(Privacy v2 설계). 교차 사용자 패턴 추출(A12)은 스트립된 익명 요약에서만 작동한다. 언어모델 API 는 zero-retention 정책으로 계약되며; 대화는 외부 모델 훈련에 사용되지 않는다.

11.4 한계

시스템은 아직 무작위 대조 시험을 거치지 않았다. 합성 벨트 생성은 교차 검증되었지만, 임상가 단독 벨트와 비교하여 맹검 평가되지 않았다. ADHD 외 장애로의 복제는 아직 시작되지 않았다. 이는 명시적 한계이며 로드맵은 각각을 순서화하여 다룬다.

12. 참고문헌

아키텍처와 임상 콘텐츠를 형성한 주요 출처(선별):

- American Psychiatric Association (2022). Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition, Text Revision.
- Beck, A. T. (1976). Cognitive Therapy and the Emotional Disorders.
- Fischer, K. W. (1980). A theory of cognitive development: The control and construction of hierarchies of skills. Psychological Review, 87(6), 477–531.
- Hertel, P. T., & Mathews, A. (2011). Cognitive Bias Modification: Past Perspectives, Current Findings, and Future Applications. Perspectives on Psychological Science.
- Kessler, R. C., et al. (2005). The World Health Organization Adult ADHD Self-Report Scale (ASRS). Psychological Medicine, 35(2), 245–256.
- Kropotov, J. D. (2009). Quantitative EEG, Event-Related Potentials and Neurotherapy.
- Lewis, P., et al. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. NeurIPS.
- Pearl, J. (2009). Causality: Models, Reasoning and Inference, 2nd Edition.
- Porges, S. W. (2011). The Polyvagal Theory.
- Posner, K., et al. (2011). The Columbia-Suicide Severity Rating Scale. American Journal of Psychiatry, 168(12), 1266–1277.
- Russell, J. A. (1980). A circumplex model of affect. Journal of Personality and Social Psychology, 39(6), 1161–1178.
- Safran, J. D., & Muran, J. C. (2000). Negotiating the Therapeutic Alliance.
- Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction, 2nd Edition.
- Taylor, G. J. (1997). Disorders of Affect Regulation: Alexithymia in Medical and Psychiatric Illness.