



コーブルク応用科学大学
電気情報学部

専攻：情報工学

インターンシップ報告書

サルマ・エル・ワルギ・エル・マイジ

メール：salma.elouarghielmaizi@stud.hs-coburg.de

企業名	奈良先端科学技術大学院大学 ソーシャルコンピューティング研究室 〒 630-0192 奈良県生駒市高山町 8916-5 奈良県生駒市
期間	01.10.2025 ~15.02.2026

提出日：28.02.2026

コーブルク応用科学大学への報告書の提出許可：

指導教員	荒牧英治教授	
職位	研究室責任者	15.02.2026
電話番号	+81743725250	
メール	aramaki@is.naist.jp	荒牧英治教授

目次

図目次	II
表目次	III
記号一覧	VI
略語一覧	VI
1 緒論	1
1.1 動機、課題設定、および目的	1
2 理論的背景	2
2.1 医療ドキュメント作成とトリアージ (SOAP & ESI)	2
2.2 AI 支援型情報抽出 (Whisper & LLM)	2
2.3 人間工学的要因と認知負荷	2
3 システムアーキテクチャと実装	3
3.1 システム構成とバックエンドロジック	3
3.2 ユーザーインターフェースとローカライゼーション戦略	6
3.2.1 メイン・ダッシュボード	6
3.2.2 透過モニター	7
3.2.3 デュアルモード・モニタリング	8
3.2.4 ハイブリッドシステム：多言語対応戦略	9
3.2.5 ペイシエント・ジャーニー・ダッシュボードとレポート作成	11
3.2.6 臨床レポート・エクスポート (PDF)	11
3.2.7 認知分析のための音声マーカー	12
3.2.8 品質指標と検証	13
4 技術的課題と解決策	15
4.1 低遅延オーディオセグメンテーション (ピンポン録音)	15
4.2 マルチウィンドウ環境における状態同期	15
4.3 部分的永続化による可用性の向上	15
4.4 非同期パイプラインにおけるスレッドセーフティ	15
4.5 API レートリミットとコスト	16
5 評価と結果	17
5.1 評価手法	17
5.2 トリアージ評価：事例分析とパフォーマンス統計	17
5.3 情報抽出：アイデンティティ、SOAP、翻訳	18

目次

6 結論と展望	20
6.1 プロジェクトの成果とリフレクション (NAIST)	20
6.2 倫理的枠組みと将来性	20
用語集	X
参考文献	XV
宣誓書	XIX

図目次

Fig. 1:	アーキテクチャ概要	3
Fig. 2:	システム・フロー	4
Fig. 3:	SOAP 詳細表示	4
Fig. 4:	異なるローカライゼーション段階におけるメイン・ダッシュボード .	7
Fig. 5:	透過モニター：ダッシュボードと待機画面表示	8
Fig. 6:	同時表示（デュアルモード）.	9
Fig. 7:	ペイシェント・ジャーニー・ダッシュボード	11
Fig. 8:	PDF エクスポートにより生成された臨床レポート	12
Fig. 9:	言語的マーカーと品質分析	13
Fig. 10:	転記精度バリデーター：BLEU と CUDA による評価	14
Fig. 11:	評価ワークフロー：CALL_110 のための Python 構造と生成された 看護師評価用 PDF。	17

表目次

Tab. 1:	SOAP ドキュメントの構造	2
Tab. 2:	ESI トリアージレベル	2
Tab. 3:	ローカライゼーション戦略の概要 (EN / JA)	9
Tab. 4:	システム評価ダッシュボード：REDSS vs 看護師グラウンド・トゥルース (n=15)	18
Tab. 5:	情報抽出：属性フィールド、臨床的 SOAP 評価、翻訳品質	19

コード一覧

1	処理パイプラインのオーケストレーション	3
2	SOAP 抽出プロンプト（抜粋）	4
3	インテリジェントなフィールド抽出	5
4	Evidence-Based Hybrid Urgency Classifier (ESI v4 + AI)	5
5	API の主要エンドポイント	6
6	React ダッシュボード - 状態管理	6
7	翻訳キャッシュの検証	7
8	統合ローカライゼーション	8
9	Statische Lokalisierung im Frontend	9
10	医療ローカライゼーションのための AI 指示	10
11	言語純度のバリデーション	10
12	自動同期メカニズム	11
13	PDF エクスポートの実装	11
14	バリデーションロジックと多言語正規化	13
15	隙間のない音声キャプチャのためのピンポン・バッファ	15
16	明示的な変換によるスレッドセーフなセッション管理	16
17	評価用データ CALL_110	17

記号一覧

記号	意味	単位
α	クリッペンドルフの α - データの信頼性を評価するための係数	[\cdot]
κ	コッパ係数 (Cohen's Kappa) - 評価者間信頼性の統計的尺度	[\cdot]
$BLEU$	Bilingual Evaluation Understudy - 翻訳精度の評価指標	[\cdot]
F_1	F1 スコア - 適合率と再現率の調和平均	[\cdot]
G	ギロー指数 (Guiraud-Index) - 語彙多様性の尺度	[\cdot]
n	サンプルサイズ	[\cdot]
P	適合率 (Precision) - 情報抽出の正確性の尺度	[\cdot]
R	再現率 (Recall) - 情報抽出の網羅性の尺度	[\cdot]
t_{ASR}	音声認識 (Whisper) の所要時間	[s]
t_{LLM}	情報抽出および分類 (LLM) の所要時間	[s]
t_{total}	パイプラインの総処理時間	[s]
TTR	Type-Token-Ratio - 語彙変異性の尺度	[\cdot]
v	オーディオパイプラインの処理速度	[s_{Audio}/s_{Real}]
WER	単語誤り率 (Word Error Rate) - 音声認識における誤り率	[$\%$]

略語一覧

AI	人工知能 (Artificial Intelligence)
ALS	二次救命処置 (Advanced Life Support)
API	アプリケーション・プログラミング・インターフェース
ASGI	非同期サーバー・ゲートウェイ・インターフェース
ASR	自動音声認識 (Automatic Speech Recognition)
BLEU	Bilingual Evaluation Understudy (翻訳精度の指標)
BLS	一次救命処置 (Basic Life Support)
CAD	コンピュータ支援指令システム (Computer-Aided Dispatch)
CJK	中国語、日本語、韓国語
CORS	オリジン間リソース共有 (Cross-Origin Resource Sharing)
CPU	中央処理装置
CSS	カスケーディング・スタイル・シート
CSV	カンマ区切り値
CUDA	Completion, Understanding, Detail, Accuracy (SOAP 評価指標)
DNP	デジタル神経心理学的評価 (Digital Neuropsychological)
DOM	ドキュメント・オブジェクト・モデル
DP	動的計画法
DTO	データ転送オブジェクト
EMD	救急医療通信員 (Emergency Medical Dispatcher)
EMS	救急医療サービス
ESI	緊急度判定指数 (Emergency Severity Index)
FastAPI	Python Web フレームワーク
FHIR	Fast Healthcare Interoperability Resources (医療情報交換規格)
GDPR	一般データ保護規則 (General Data Protection Regulation)
GPT	Generative Pre-trained Transformer
GPU	画像処理装置
HIS	病院情報システム (Hospital Information System)

HL7	Health Level 7 (医療情報交換標準規格)
HTML	ハイパーテキスト・マークアップ・ランゲージ
HTTP	ハイパーテキスト・トランスファー・プロトコル
HTTPS	Hypertext Transfer Protocol Secure
IDE	統合開発環境
JSON	JavaScript Object Notation
LLM	大規模言語モデル (Large Language Model)
ML	機械学習
NAIST	奈良先端科学技術大学院大学
NLP	自然言語処理
NLTK	Natural Language Toolkit
OpenAI	OpenAI Inc.
ORM	オブジェクト関係マッピング
PDF	ポータブル・ドキュメント・フォーマット
RAM	ランダム・アクセス・メモリ
React	ユーザーインターフェース用 JavaScript ライブライ
REDSS	リアルタイム救急意思決定支援システム
Regex	正規表現 (Regular Expressions)
REST	Representational State Transfer
SOAP	主訴 (S), 客観的所見 (O), 評価 (A), 計画 (P)
SQL	構造化クエリ言語
SQLite	構造化クエリ言語ライト
TTR	Type-Token Ratio (語彙多様性指標)
UI	ユーザー・インターフェース
URL	Uniform Resource Locator
WER	単語誤り率 (Word Error Rate)

1 緒論

1.1 動機、課題設定、および目的

救急指令センターにおいて、救急医療通信員（EMDs）は、救助を求める人々と病院前救急医療との間の重要なインターフェースである。極度の時間的制約の下で、不完全であったり感情的であったりする会話から、正確な医療情報を抽出、記録、優先順位付けしなければならない。[KVS18] この際、通信員は限られた作業記憶容量により高い認知負荷にさらされる [Mil56; Zap+25]。現代の音声認識および自然言語処理技術は、この負担を軽減する新たな可能性を提供している [Blo+19; Luo+25]。

現在の手動による通知処理には、構造的な欠陥がある。第一に、通話とドキュメント作成を逐次的に行うことで遅延が生じる。これは、関連情報が通話終了後に確定されることが多いいためである [Mel+25; Sco+16]。また、マニュアルを逐次的に参照することも要因となる [CDM08, p. 45–52]。第二に、医療的初期評価（トリアージ）は個人間のばらつきが大きく、不一致な判断につながる可能性がある [Mis+18; Sax+23]。第三に、非構造的な自由記述のメモは、標準化された臨床的引き継ぎやデータの追跡可能性を困難にする [PLG23; Wee68]。第四に、高いストレス負荷は通信員の作業記憶を著しく損ない、情報喪失のリスクを高める [KVS18; LeB09; Pin17]。コンピュータ支援システムは、この文脈において、より高い診断の安定性とプロセス上の安全性の向上を約束するものである。

このような背景から、本研究では以下の中心的な研究課題を調査する：AI 支援によるリアルタイムシステムは、自動化された SOAP 抽出と標準化されたトリアージを通じて、救急通報処理の質をどの程度向上させることができるか？

この問い合わせるために答えるため、以下の部分目標を追求する：

1. **システム開発**：自動転記（Whisper）、LLM ベースの SOAP 抽出、および ESI ベースのトリアージからなるリアルタイムパイプラインの概念設計と実装。
2. **評価**：転記精度およびシステムパフォーマンスの定量的評価。
3. **臨床的検証**：医療専門職 ($n = 15$) によるシステムの定性的評価。
4. **多言語対応**：英語と日本語の間の双方向翻訳の実装。

2 理論的背景

2.1 医療ドキュメント作成とトリアージ（SOAP & ESI）

医療ドキュメント作成は SOAP メソッド（表 1）に基づいて行われ [PLG23; Wee68]、トリアージは Emergency Severity Index (ESI) を用いて緊急度の優先順位を決定する [Gil+20]（表 2）。手動トリアージの誤トリアージ率は最大 32% に達するため [Mis+18; Sax+23; Tan+07]、自動化によって信頼性を高めることが目標である。これは ANKUTRIAGE などのコンピュータ支援システムによって既に実証されている [Koc+22]。

表 1: SOAP ドキュメントの構造

セクション	内容と説明
主訴 (S)	患者の訴え、症状、痛みの記述（ナラティブ）。
客観 (O)	測定可能な事実：氏名、年齢、バイタルデータ、意識、住所。
評価 (A)	医療的な作業仮説および臨床的評価。
計画 (P)	治療措置および派遣される救急手段。

表 2: ESI トリアージレベル

ESI	基準 / 緊急度	資源
1 (極めて緊急)	生命の危険があり、即時の生命維持措置が必要。	即時
2 (緊急)	高い緊急性；不安定な状態または高いリスク。	高
3 (中等度)	安定しているが、多くの資源が必要（診断等）。	≥ 2
4 (低)	安定しており、最大 1 つの医療資源が必要。	1
5 (非緊急)	安定しており、医療資源を必要としない（相談等）。	0

システム適応： REDSS では、ESI プロトコルに従い、ESI 4 と 5 を「LOW」に統合している。これは両レベルとも直接的な資源投入を必要としないためである [Gil+20]。

2.2 AI 支援型情報抽出（Whisper & LLM）

OpenAI Whisper は、音響的に困難な緊急事態における転記のための堅牢な音声認識モデルとして機能する [Luo+25; Rad+23; Zec+25]。本システムでは、低遅延のために「base」モデルを使用している。GPT-4o は、91% 以上の精度で構造化情報の抽出を担当し [Gup+25; Per+24]、コンテキスト把握を向上させるために逐次推論 (Chain-of-Thought) を利用している [Ada+24; Wei+22]。

2.3 人間工学的要因と認知負荷

急性のストレスは作業記憶の容量を著しく低下させ [Mil56]、ドキュメント作成とアセスメントなどの並行作業を困難にする [LeB09; Pin17]。これによりエラー率が高まり、バーンアウトのリスクが増加する [KVS18]。コンピュータ支援ツールは、ドキュメント作成の負担を引き受けることでスタッフを支援し、コミュニケーションへのより強い集中を可能にする [Mel+25; Sco+16]。

3 システムアーキテクチャと実装

3.1 システム構成とバックエンドロジック

REDSSシステムは、3層アーキテクチャを採用している [BCK03]。React フロントエンドは REST および WebSocket を介して FastAPI バックエンドと通信し、データはSQLiteに保存される（図 1）。パイプラインは、録音から保存までのすべてのステップを調整する（図 2）。

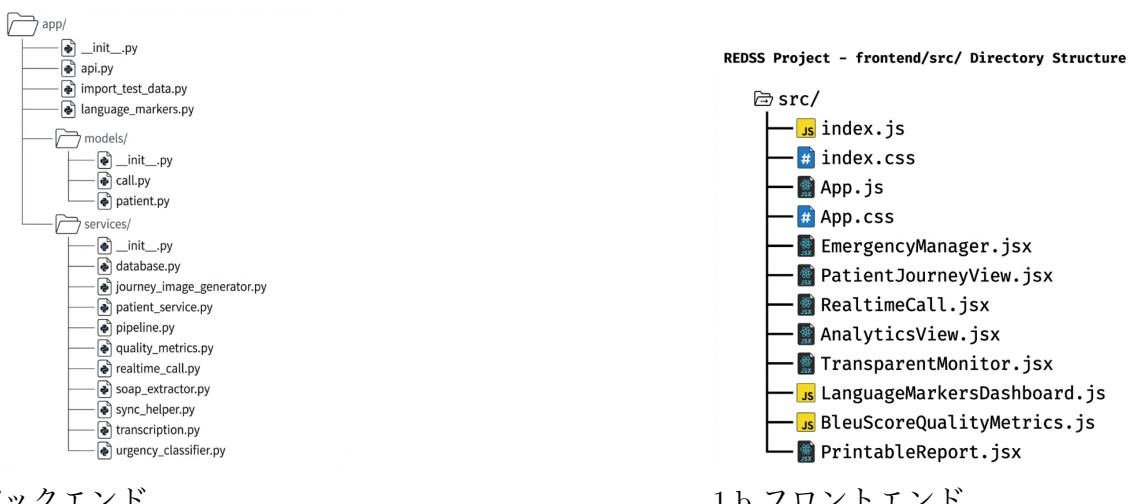


図 1: アーキテクチャ概要

処理パイプライン： 非同期パイプラインは、音声録音から構造化データ出力までの中心的な処理ステップをオーケストレートする（コード 1）。

```

1  async def process_call(self, audio_path: str) -> dict:
2      call_id = f"CALL_{datetime.now().strftime('%Y%m%d_%H%M%S')}"
3      try:
4          # ステップ 1: 音声を転記
5          logger.info(f"{call_id} の転記を開始")
6          transcription = await transcription_service.transcribe(audio_path)
7          transcript = transcription['text']
8          # ステップ 2: SOAP ノートを抽出
9          logger.info(f"{call_id} の SOAP ノートを抽出")
10         soap = await soap_extractor.extract(transcript)
11         # ステップ 3: 緊急度を分類
12         urgency = await urgency_classifier.classify(transcript, soap)
13         # ステップ 4: データベースに保存（アトミック）
14         with get_db() as db:
15             call = save_call(db, call_id, audio_path, transcript)
16             update_soap(db, call_id, soap)
17             update_urgency(db, call_id, urgency)
18             db.commit()
19         return self._build_response(call_id, transcript, soap, urgency)
20     except Exception as e:
21         logger.error(f"処理エラー: {str(e)}")
22         raise ProcessingError(f"パイプラインエラー: {str(e)}") from e

```

コード 1: 処理パイプラインのオーケストレーション

3 システムアーキテクチャと実装

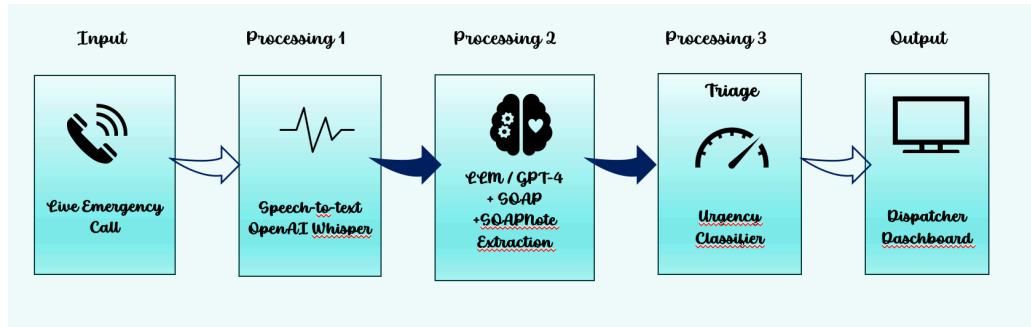


図 2: システム・フロー

音声転記は、遅延と精度の最適なトレードオフを実現するために Whisper の「base」モデルを使用している [Rad+23; Zec+25]。SOAP 抽出は、Chain-of-Thought アプローチを用いたプロンプトエンジニアリングによって行われる [Wei+22] :

```

1 You are a medical AI assistant. TASK: Extract structured SOAP notes.
2 GUIDELINES: Identify symptoms, vitals, patient identifiers, and clinical reasoning.
3 OUTPUT FORMAT (STRICT):
4 <S>- Subjective observations and patient complaints</S>
5 <O>- Name: [extracted name]
6   - Age: [number]
7   - Address: [full address]
8   - Symptoms: [list]
9 </O>
10 <A>- Clinical assessment and suspected condition</A>
11 <P>- Emergency response plan and instructions</P>
  
```

コード 2: SOAP 抽出プロンプト（抜粋）

The screenshot shows a user interface for 'SOAP Clinical Analysis'. It displays four main sections: SUBJECTIVE, OBJECTIVE, ASSESSMENT, and PLAN. The SUBJECTIVE section contains a narrative about a teenager named Jake who drank excessively at a party. The OBJECTIVE section lists Jake's details: Name: Jake Morrison, Age: 18, Address: 8923 University Boulevard, Phone: [Not provided], Blood: [Not provided]. The ASSESSMENT section notes signs of potential alcohol poisoning. The PLAN section states that paramedics are being dispatched. Below these, there is a 'System Performance (BLEU)' section with an 'EXTRACTION QUALITY' icon showing 'System Validated' and a 'BLEU OK' button. At the bottom, there is a 'Clinical Transcription' section with a transcription of a conversation about adults at the party.

図 3: SOAP 詳細表示

3 システムアーキテクチャと実装

LLM ベースの抽出アプローチは、構造化された臨床情報の抽出に関する確立された手法に従っている [Ada+24; Gup+25]。

患者データの堅牢な抽出のために、LLM による構造化を補完し、潜在的なフォーマットエラーをキャッチする追加の正規表現パターンが使用されている：

```
1 def parseField(text, fieldName):
2     # 多言語ラベルのサポート
3     translations = {
4         'Name': ['Name', '氏名', '氏名', 'Nom'],
5         'Age': ['Age', '年齢', '年齢', 'Edad'],
6         'Address': ['Address', '住所', '住所', 'Location']
7     }
8     keys = translations.get(fieldName, [fieldName])
9
10    for key in keys:
11        patterns = [
12            f"{key}\n\s*: \s*(\n|\r|\r\n)+", # 標準
13            f"\n*\n{s*{key}}\n\s*: \s*(\n|\r|\r\n)+", # 簡条書き
14            f"^{key}\n\s*: \s*(\n|\r|\r\n)+" # 行頭
15        ]
16        for pattern in patterns:
17            match = re.search(pattern, text, re.IGNORECASE)
18            if match and is_valid_value(match.group(1)):
19                return match.group(1).strip()
20
21    return 'N/A'
```

コード 3: インテリジェントなフィールド抽出

緊急度判定は、キーワードマッチングと ESI 基準に基づく AI 文脈分析を統合したハイブリッド手法を採用している [Gil+20]。この戦略では、明確な ESI 基準（ルールベース）と、AI による確率的な文脈解析を組み合わせている：

1. **ESI Level 1/2 (極めて緊急/高リスク)**： ESI を信頼する（エビデンスに基づき、常にエスカレート）。
 2. **ESI Level 4/5 (低/最小限)**： ESI を信頼する。ただし、AI がより高い緊急度を検出した場合は例外（安全優先）。
 3. **ESI Level 3 (中等度)**： AI による精緻化を利用（コンテキスト認識）。
 4. **安全優先**：いずれかの方法がそれを示唆する場合、常により高い緊急度にエスカレートする。
5. **特殊ケース**： AI はテスト通話や無意味な内容を認識し、適切に格下げする。

```
1 class HybridUrgencyClassifier:
2     # ESI Level 1: Immediate Lifesaving Criteria
3     CRITERIA = {
4         "en": ["intubated", "cardiac arrest", "unconscious", "apneic"],
5         "ja": ["挿管", "心停止", "意識不明", "無呼吸"]
6     }
7     def classify(self, transcript, soap, lang="en"):
8         # 1. Rule-based ESI Check (Speed & Safety)
9         if any(term in transcript for term in self.CRITERIA.get(lang, [])):
10             return {"level": "CRITICAL", "esi": 1}
11         # 2. AI によるコンテキスト分析 (インストラクションの最適化)
12         prompt = f"""
13             Role: Emergency MD. Task: ESI Triage.
14             Input: {transcript} | Context: {json.dumps(soap)}
```

3 システムアーキテクチャと実装

```
15 Format: {{ "level": "ESI_LEVEL", "logic": "Rationale in {lang}" }}
16 """
17     return self.call_ai(prompt)
```

コード 4: Evidence-Based Hybrid Urgency Classifier (ESI v4 + AI)

クライアントとの通信は、以下の REST エンドポイントを介して行われる：

```
1 # 音声のアップロードと処理
2 POST /api/upload
3 # テキストベースの入力
4 POST /api/process-text
5 # 通報リストの取得（翻訳付き）
6 GET /api/calls?lang=en|ja
7 # 個別の通報データの取得
8 GET /api/calls/{call_id}?lang=en|ja
9 # リアルタイム通話用 WebSocket
10 WS /ws/realtime-call
11 # 患者管理
12 GET/POST/PUT /api/patients
```

コード 5: API の主要エンドポイント

3.2 ユーザーインターフェースとローカライゼーション戦略

ユーザーインターフェースは、英語と日本語の同時利用に最適化されている。React ベースのフロントエンドには、運用管理、ライブ通話、治療経過のためのダッシュボードが含まれている。中央のダッシュボードがグローバルな状態を管理する：

```
1 const EmergencyManager = () => {
2     const [calls, setCalls] = useState([]);
3     const [language, setLanguage] = useState('en');
4     const [urgencyFilter, setUrgencyFilter] = useState('ALL');
5     // リアルタイム更新のためのポーリング
6     useEffect(() => {
7         const fetchCalls = async () => {
8             const response = await fetch('/api/calls?lang=${language}');
9             const data = await response.json();
10            setCalls(data);
11        };
12        fetchCalls();
13        const interval = setInterval(fetchCalls, 5000); // 5 秒ごと
14        return () => clearInterval(interval);
15    }, [language]);
16    return (
17        <div className="dashboard">
18            <Header language={language} setLanguage={setLanguage} />
19            <StatsCards calls={calls} />
20            <CallList calls={calls} urgencyFilter={urgencyFilter} />
21        </div>
22    );
23};
```

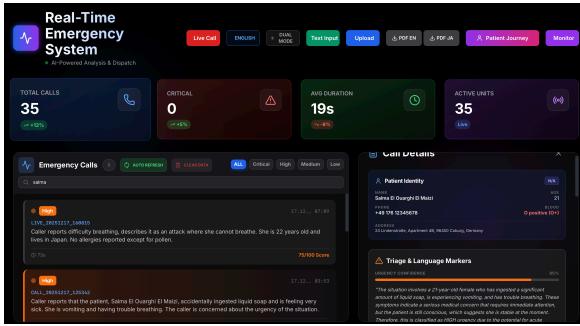
コード 6: React ダッシュボード - 状態管理

5 秒のポーリング間隔は、リアルタイム更新とサーバー負荷の妥協点である。これより短い間隔は、API のレートリミットに不必要的負荷をかけることになる [BG21]。

3.2.1 メイン・ダッシュボード

運用管理用の中央ダッシュボードは、すべての緊急通報の概要を提供し、緊急度によるフィルタリングやシステム言語の切り替えが可能である（図 4）。

3 システムアーキテクチャと実装



4.a 英語表示



4.b 日本語表示

図 4: 異なるローカライゼーション段階におけるメイン・ダッシュボード

動的ローカライゼーション・パイプライン (バックエンド): バックエンドのインテリジェンス (`app/api.py`) は 2 段階で動作する。まず、遅延とコストを最小限に抑えるために、既存の翻訳がないかキャッシュ・チェックを行う。キャッシュミスが発生した場合は、AI 翻訳 (GPT-4o) が、臨床用語や文化的な背景 (例: 日本の住所形式) を考慮した医学的に正確なローカライゼーションを実行する。

3.2.2 透過モニター

このセカンダリディスプレイは、リアルタイム同期を可能にし [WHA23]、メインデバイスでの操作を妨げることなく、スタッフに最も重要なデータを一目で確認させる。ローカライゼーションは単一のリクエストに統合され、応答時間を約 20 秒から 1 秒未満 (キャッシュヒット時) に短縮している (図 5)。多言語システム (EN/JA) は双方向翻訳をサポートし、冗長な API 呼び出しを避けるためにキャッシュを利用する：

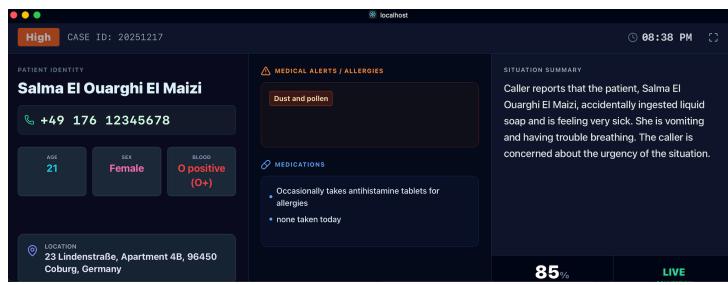
```

1 CACHE_VERSION = "v2"
2
3 def is_valid_cache(cached_data, target_lang):
4     # バージョンチェック
5     if cached_data.get("_version") != CACHE_VERSION:
6         return False
7     # 言語の純度チェック
8     if target_lang == "ja":
9         # 英語のラベルが含まれていないか確認
10        english_indicators = ["Name:", "Age:", "not provided"]
11        if any(ind in cached_data["soap_objective"] for ind in english_indicators):
12            return False
13    elif target_lang == "en":
14        # 日本語の文字が含まれていないか確認
15        japanese_pattern = r'[\u3040-\u309F\u30A0-\u30FF\u4E00-\u9FFF]'
16        if re.search(japanese_pattern, cached_data["soap_objective"]):
17            return False
18    return True

```

コード 7: 翻訳キャッシュの検証

3 システムアーキテクチャと実装



5.a 透過ダッシュボード



5.b 時刻 (EN)



5.c 時刻 (JA)

図 5: 透過モニター：ダッシュボードと待機画面表示

```
1 def localize_call_data(transcript, soap_notes, metadata, target_lang):
2     """
3     3つ以上の個別コールではなく、1つの API リクエストで実行
4     文字起こし、SOAP (全 4 セクション)、メタデータを翻訳
5     """
6     prompt = f"""
7     Translate and localize the following emergency call data to {target_lang}.
8     IMPORTANT:
9     - Translate field labels (Name, Age)
10    - Preserve medical accuracy
11    - Maintain SOAP structure
12    TRANSCRIPT: {transcript}
13    SOAP NOTES: {json.dumps(soap_notes)}
14    METADATA: {json.dumps(metadata)}
15    OUTPUT: Return JSON with all translated fields
16    """
17    response = openai.ChatCompletion.create(
18        model="gpt-4-mini",
19        messages=[{"role": "user", "content": prompt}]
20    )
21    return parse_localized_response(response)
```

コード 8: 統合ローカライゼーション

3.2.3 デュアルモード・モニタリング

デュアルモードボタンにより、品質管理のために両方の言語を同時に表示することができる。これは、通信員がいつでも AI 翻訳を確認し、必要に応じて修正できるという Human-in-the-Loop 設計コンセプトを象徴している。図 6 に示すように、インターフェース全体が隙間なくローカライズされている。

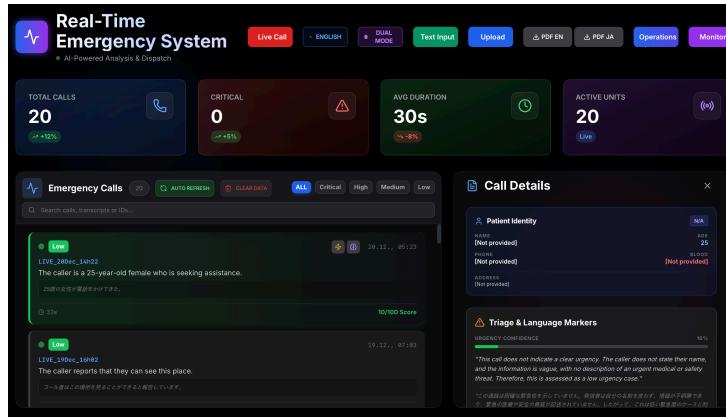


図 6: 同時表示 (デュアルモード)

3.2.4 ハイブリッドシステム：多言語対応戦略

二言語の実装は、ユーザーインターフェースの精密な制御と AI によるコンテンツの柔軟性を組み合わせたハイブリッドアプローチに従っている。ここでは、静的（ハードコード）コンテンツと動的（AI 駆動）コンテンツを厳密に区別している。

実装マトリックス： 以下の表は、システムにおける責任の分担をまとめたものである：

表 3: ローカライゼーション戦略の概要 (EN / JA)

機能	タイプ	コンポーネント / 場所
ボタン・メニュー	静的	EmergencyManager.jsx (TRANSLATIONS)
テーブルヘッダー	静的	フロントエンド UI レイヤー
通報文字起こし	動的	soap_extractor.py (OpenAI)
SOAP ノート	動的	OpenAI GPT-4o-mini
緊急度の理由	動的	OpenAI GPT-4o-mini
バリデーション	ロジック	app/api.py (言語混合チェック)

このハイブリッドアーキテクチャにより、UI の応答性と安定性を維持しつつ、複雑な医療データを動的かつスケーラブルに翻訳することが可能となる。

レベル 1：UI 翻訳（静的） 誤操作やメニュー表示における「AI ハルシネーション」を防ぐため、固定 UI テキストはすべて辞書システムで管理されている。言語切り替え時には、UI 全体の状態が即座に更新される。

```

1 // frontend/src/EmergencyManager.jsx
2 const TRANSLATIONS = {
3   en: {
4     title: "Real-Time Emergency System",
5     liveCall: "Live Call",
6     patientIdentity: "Patient Identity",
7     urgency: "Urgency Level",

```

3 システムアーキテクチャと実装

```
8     critical: "Critical"
9   },
10  ja: {
11    title: "救急意思決定支援システム",
12    liveCall: "ライブ通話",
13    patientIdentity: "患者基本情報",
14    urgency: "緊急度レベル",
15    critical: "最優先(重症)"
16  }
17};
```

コード 9: Statische Lokalisierung im Frontend

レベル 2: AI インストラクションと最適化 (AI 駆動): 医療ローカライゼーションの核心は、プロンプトの動的な生成にある。ここでは、AI に単にテキストを渡すだけでなく、対象文化の正確な医療構造（例：日本の住所形式）も学習させている。

```
1 # app/services/soap_extractor.py
2 def localize_call_data(self, transcript, soap_notes, metadata, target_language):
3     is_japanese = target_language in ["ja", "jp", "japanese"]
4     lang_name = "JAPANESE" if is_japanese else "ENGLISH"
5     # 対象言語固有のラベルを定義
6     if is_japanese:
7         labels = "氏名:, 年齢:, 住所:, 電話:, 血液型:"
8         missing = "[不明]"
9     else:
10        labels = "Name:, Age:, Address:, Phone:, Blood:"
11        missing = "[Not provided]"
12    # 医療的正確性のためのインストラクション最適化
13    prompt = f"""
14        You are a medical localization expert.
15        Localize the following emergency call data into {lang_name}.
16        EXTREMELY IMPORTANT: In the Objective section, you MUST use these exact labels: {labels}.
17        If information is missing, use "{missing}".
18        Maintain the clinical tone in the Reasoning section.
19        """
20    return self.call_openai_api(prompt, transcript, soap_notes)
```

コード 10: 医療ローカライゼーションのための AI 指示

レベル 3: 混合防止と品質保証: 対象言語の純度を確保するために、自動化された品質管理が実装されている。このアルゴリズムは、AI が誤って元の言語のラベルを残していないか（「ジャパングリッシュ」）を検出し、そのような場合には再翻訳を強制する。

```
1 # app/api.py
2 def validate_localization(lang, obj_text):
3     # 言語混合の高度な検出
4     if lang == "ja":
5         # 日本語には英語のラベルが含まれるべきではない
6         has_english_labels = any(label in obj_text for label in ["Name:", "Age:", "Address:])
7         if has_english_labels:
8             print(f"警告: 日本語キャッシュ内に英語の標記があります。再翻訳を実行します... ")
9             return False # 再翻訳を強制
10    elif lang == "en":
11        # 英語には日本語の文字が含まれるべきではない
12        import re
13        has_japanese_chars = bool(re.search(r'[\u3040-\u309F\u30A0-\u30FF\u4E00-\u9FFF]', obj_text))
14        if has_japanese_chars:
15            print(f"警告: 英語キャッシュ内に日本語の文字があります。再翻訳を実行します... ")
16            return False
17    return True
```

コード 11: 言語純度のバリデーション

3.2.5 ペイシェント・ジャーニー・ダッシュボードとレポート作成

自動同期： ペイシェント・ジャーニー・トラッキング・プロセスにより、新しい救急通報を既存のデータセットと自動的に同期させることができる：

```

1 def sync_emergency_call_to_patient_journey(call_id):
2     call = db.query(EmergencyCall).filter_by(id=call_id).first()
3     patient = db.query(Patient).filter_by(name=call.patient_name).first()
4     if not patient:
5         patient = Patient(
6             name=call.patient_name,
7             primary_condition=call.disease,
8             medical_history=call.soap_subjective,
9             journey_events=json.dumps([{"segment": 1, "date": call.created_at.date(),
10                                         "description": f"初回: {call.disease}", "status": "completed"}])
11     )
12     db.add(patient)
13 else:
14     events = json.loads(patient.journey_events or "[]")
15     events.append({"segment": len(events) + 1, "date": call.created_at.date(),
16                     "description": f"フォローアップ: {call.disease}", "status": "completed"})
17     patient.journey_events = json.dumps(events)
18 db.commit()

```

コード 12: 自動同期メカニズム

患者の経過を追跡することで、最初の救急通報から治療に至るまでの継続的な監視が可能になる。



図 7: ペイシェント・ジャーニー・ダッシュボード

3.2.6 臨床レポート・エクスポート (PDF)

生成されたレポートは、緊急通報受領から臨床的後続加療への架け橋となる。ここではコンパクトな情報の受け渡しに焦点を当てており、エクスポートには初期アヌマネーゼを簡潔に表すための S (主観) と O (客観) のデータのみが含まれる。

```

1 const exportToPDF = async () => {
2     const { default: jsPDF } = await import('jspdf');
3     const doc = new jsPDF('p', 'mm', 'a4');
4     // ヘッダーと患者メタデータ

```

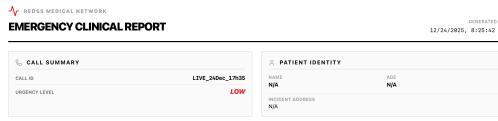
3 システムアーキテクチャと実装

```

5 doc.setFontSize(22);
6 doc.text('Patient Journey Report', 20, 20);
7 doc.text(`Name: ${selectedPatient.name}`, 20, 45);
8 // ジャーニーイベントの反復処理
9 journeyEvents.forEach((event, i) => {
10   doc.text(`#${event.date}: ${event.description}`, 20, 70 + i*10);
11 });
12 doc.save(`Patient_Journey_${selectedPatient.name}.pdf`);
13

```

コード 13: PDF エクスポートの実装



8.a 英語レポート

8.b 日本語レポート

図 8: PDF エクスポートにより生成された臨床レポート

3.2.7 認知分析のための音声マーカー

LanguageMarkerAnalyzer は、学術的に確立されたバージョンへと進化し、転記された内容を構造化された閾値 (MarkerThresholds) を用いて認知機能の低下について調査する。分析は、発話の流暢さ (Speech Fluency)、認知、言語、および意味論的マーカーに基づいてモジュール式に行われる。核となる機能の一つは、Type-Token-Ratio (TTR) および Guiraud 指数を用いた語彙の多様性の分析である。TTR は語彙の変動性の尺度であり ($TTR = Types/Tokens \in [0, 1]$) [CM10]、Guiraud 指数 ($G = Types/\sqrt{Tokens}$) は、テキストの長さに依存しない偏りのない分析を可能にするために採用されている。本システムは、パッシブな音声分析を通じて、緊急時における微妙な神経学的変化 (例: 認知症の兆候) を客観的に評価し、臨床支援を行うことを目的としている [Roa+11]。

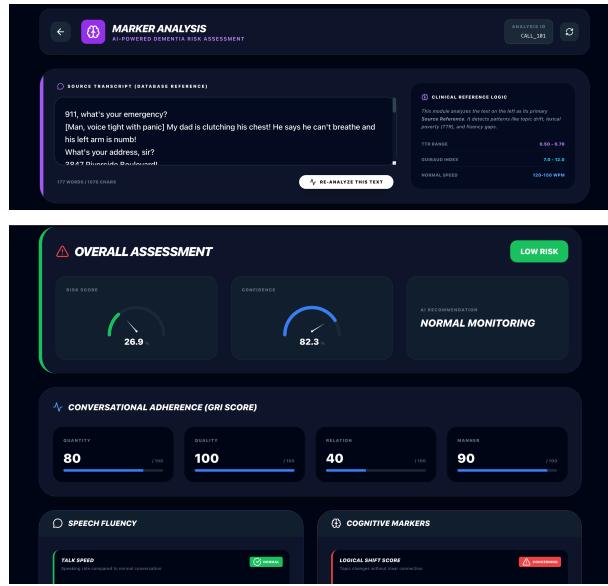


図 9: 言語的マーカーと品質分析

3.2.8 品質指標と検証

医療現場での音声認識システムを評価するには、機械的な認識率 (WER) と意味的な正確性 (BLEU スコアおよび臨床向け指標 CUDA) を明確に区別する必要がある [Gup+25; Pap+02] :

$$WER = \frac{S+D+I}{N} \quad BLEU = BP \cdot \exp \left(\sum_{n=1}^N w_n \ln p_n \right) \quad (3.1)$$

これらを補完するため、医療的な整合性を評価する CUDA モデルを採用した。この臨床 SOAP ノート評価システムは、以下の 4 つの次元を統合している：Completion (参照情報の網羅性)、Understanding (単語の重複に基づくジャカード類似度)、Detail (情報を失わないための情報密度の比率)、および Accuracy (ハルシネーションを避けるための精度)。CUDA は、BLEU スコアとともに、自動抽出された情報が医療スタッフにとってどの程度正確かつ有用であるかをデータに基づいて推定する。

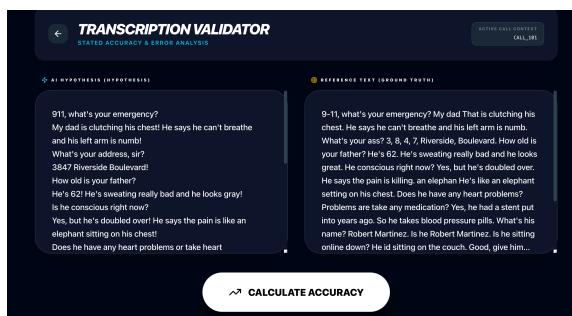
```

1 def clean_text(self, text):
2     # 1. ノイズ除去 (テキスト正規化)
3     text = re.sub(r'\[.*?\]', '', text.lower())
4     # 2. 漢字・かな等の Unicode 範囲の保護
5     text = re.sub(r'[\w\ufe0e\u3040-\u309f\u30a0-\u30ff\u4e00-\u9fff]', ' ', text)
6     return text.strip()
7 def validate_transcription(self, ref, hyp):
8     # 機械的評価 (WER) : 動的計画法による編集距離
9     wer = calculate_levenshtein(ref, hyp) / len(ref.split())
10    # 意味的評価 (CUDA) : 臨床的なデータ整合性 [Gup+25]
11    cuda = (completion + understanding + detail + accuracy) / 4
12    return {"wer_acc": max(0, (1-wer)*100), "cuda": cuda}

```

コード 14: バリデーションロジックと多言語正規化

3 システムアーキテクチャと実装



10.a テキスト比較 (参照 vs 推論)



10.b BLEU スコアと CUDA メトリクス

図 10: 転記精度バリデーター：BLEU と CUDA による評価

この開発ループにより、開発者はバリデーター（図 10）を用いて、色分けされた不一致箇所に基づいて AI の結果を確認できる。システムの信頼性をテストし、転記が実際にどれほど正確か、あるいは誤っているかを検証するために、このダッシュボードが実装された。

4 技術的課題と解決策

4.1 低遅延オーディオセグメンテーション（ピンポン録音）

リアルタイム転記における重大な課題の一つは、チャンク生成中のデータ損失である。従来の `MediaRecorder` の実装では、セグメントが終了して次が開始される際に音声ストリームに隙間が生じてしまう。

この解決策として、クライアント側でのピンポン・バッファリング手法を採用した。ここでは、2つの並列な `MediaRecorder` インスタンスを時間差で制御することで、隙間のないキャプチャを保証している：

```
1 const switchRecorders = () => {
2   const nextIndex = (activeIndex + 1) % 2;
3   const current = recorders[activeIndex];
4   const next = recorders[nextIndex];
5   next.start(); // 古い録音が止まる前に新しい録音を開始
6   if (current.state === 'recording') {
7     current.stop(); // チャンクを閉じて送信
8   }
9   activeIndex = nextIndex;
10};
```

コード 15: 隙間のない音声キャプチャのためのピンポン・バッファ

4.2 マルチウィンドウ環境における状態同期

データベースへの継続的な問い合わせ（ポーリング）に頼ることなく、ダッシュボードと独立したモニター（透過モニター）を同時に更新するという要件は、`BroadcastChannel API` を通じて解決された。これにより、ブラウザのコンテキスト間で最小限の遅延でイベント駆動型の通信が可能になった。

4.3 部分的永続化による可用性の向上

API タイムアウトなどによるシステム障害に対処するため、**Partial Results** を保存するパイプラインパターンを組み込んだ。AI による解析が失敗した場合でも、医療用ドキュメント作成の義務を果たすために、転記内容と音声ファイルは永続化される。

4.4 非同期パイプラインにおけるスレッドセーフティ

FastAPI の非同期コンテキストにおいて、`SQLAlchemy` セッションの管理による `DetachedInstanceError` 例外の回避が大きな課題であった。これは、セッションが閉じられた後にオブジェクトの属性にアクセスしようとした際に発生する [Fow03]。解

4 技術的課題と解決策

決策として、アクティブなセッション内でデータを Python の辞書形式に変換する「Data Transfer Object」パターンを適用した：

```
1 def get_call_data(call_id: str) -> dict:
2     """セッション管理を伴う安全なデータ抽出"""
3     with get_db() as db:
4         call = db.query(EmergencyCall).filter_by(id=call_id).first()
5         # アクティブなセッション、つまり内部で辞書に変換
6         call_dict = {
7             'id': call.id,
8             'transcript': call.transcript,
9             'soap_subjective': call.soap_subjective,
10            'soap_objective': call.soap_objective,
11            'urgency_level': call.urgency_level,
12            'created_at': call.created_at.isoformat()
13        }
14        # セッションは閉じられるが、辞書は安全に使用可能
15    return call_dict
```

コード 16: 明示的な変換によるスレッドセーフなセッション管理

データベース層を非同期パイプラインから分離することで、クライアントにおけるスレッドセーフティとデータの一貫性が確保される [BG21; Fow03]。

4.5 API レートリミットとコスト

Google Gemini 1.5 Flash の不安定さ（接続エラー）から、OpenAI GPT-4o-mini へと切り替えた。API コストは、キャッシュ（コード 7）、プロンプトの最小化、およびリクエストの統合（コード 8）によって最適化されている。

5 評価と結果

5.1 評価手法

評価は、3つの段階を経て検証された15の合成救急通報シナリオに基づいている：(1) 初期化とREDSSによる転記処理、(2) 結果の検証可能なPDFレポートへの集約、(3) リッカート尺度(1-5)を用いた資格を持つ専門看護師による臨床的検証である。これらの専門家による修正が、すべての比較における臨床的な正解(グラウンド・トゥルース、expected_urGENCY)となる。図11はこのワークフローを示している。左側の技術的データ構造が、右側のPDF形式の臨床アセスメントと対比されている。

```

1 EVALUATED_CALLS = [
2     "call_id": "CALL_110",
3     "text": "***911, what's your emergency?
4 [Elderly woman, calm] Hello dear, I'm calling because I've had a
5 nosebleed for about 45 minutes and it won't stop.
6 What's your address, ma'am?
7 3421 Sycamore Boulevard, apartment 12C.
8 How old are you?
9 I'm 81. My name is Helen Reed.
10 Are you on any blood thinners?
11 """
12     "expected_urGENCY": "medium",
13     "expected_type": "medical",
14     "expected_location": "3421 Sycamore Boulevard, apartment 12C",
15     "expected_agent": "Helen Reed",
16     "expected_soap": {
17         "subjective": "81-year-old female with unilateral epistaxis (right naris) for 45 minutes, on warfarin for mechanical heart valve, reports lightheadedness, history of hypertension, lives alone",
18         "objective": "Patient alert and conversational, applying appropriate first aid (pinching and forward lean), prolonged bleeding despite measures, anticoagulated patient, hypertensive history, mild orthostatic symptoms reported",
19         "assessment": "Prolonged epistaxis in anticoagulated elderly patient, possible posterior bleed, mild hypovolemia",
20         "plan": "BLS transport for ENT evaluation, likely nasal packing needed, check INR/PT levels, blood pressure monitoring, assess for posterior vs anterior source, possible cauterization or packing, ensure family support, monitor for continued bleeding"
21     }
]

```

コード 17: 評価用データ
CALL_110

The PDF report contains the following sections:

- Case 10 of 15: CALL_110**
- Patient:** Helen Reed
- Urgency:** HIGH
- AI TRIAGE CLASSIFICATION**
- AI Predicted Urgency:** HIGH
- Triage Logic:** The patient is an 81-year-old woman with a prolonged nosebleed lasting 45 minutes, which is concerning given her use of warfarin and her age. The combination of anticoagulation therapy and the duration of the bleeding increases her risk of significant complications. Although she is not in immediate distress, the lightheadedness she reports, along with her medical history, indicates a potential for deterioration. Therefore, she requires urgent evaluation and intervention, placing her at ESI Level 2. (AI-enhanced from ESI Level 3)
- Nurse Assessment:**
 - ACCURATE
 - SHOULD BE HIGHER
 - SHOULD BE LOWER
 - Comments: medium
- CLINICAL SUMMARY (S + O)**
- For Context:** The caller is an 81-year-old woman named Helen Reed who has been experiencing a nosebleed for about 45 minutes that won't stop. She reports that she has had occasional nosebleeds in the past, which usually resolve in 10 to 15 minutes. She feels a little lightheaded but attributes it to her age. She takes warfarin and has an artificial heart valve. The nosebleed started while she was reading her book, and she has been pinching her nose and leaning forward as instructed. Name: Helen Reed Age: 81 Address: 3421 Sycamore Boulevard, apartment 12C Phone: [Not provided] Blood: [Not provided]
- Assessment:** The patient is experiencing a prolonged nosebleed, likely exacerbated by the use of blood thinners (warfarin) and her age. She is lightheaded but does not report severe dizziness or weakness.
- Plan:** Paramedics will be dispatched to check on the patient due to the duration of the nosebleed and her use of blood thinners. She is advised to continue pinching her nose until help arrives.

看護師評価用 PDF 出力

図11: 評価ワークフロー: CALL_110 のための Python 構造と生成された看護師評価用 PDF。

5.2 トリアージ評価: 事例分析とパフォーマンス統計

表4は、15の緊急通報シナリオの詳細な個別結果をまとめたものである。この表では、医療スタッフによる臨床評価(GT)と、REDSSシステムによる自動分類を直接比較し、安全性(ステータスと分析)の観点から偏差を分析している。全体としてシステムは良好な結果を示しているが、特に患者の安全を確保するために、過小トリアージ(15例中2例)の割合については慎重な検討が行われた。基礎となる処理時間はMacBook Air M2で測定され、パイプライン全体でリアルタイム機能(6~20秒)が維持されていることが示された。

5 評価と結果

表 4: システム評価ダッシュボード：REDSS vs 看護師グラウンド・トゥルース (n=15)

通報 ID	シナリオ	Expected (GT)	System	ステータス・分析
CALL_101	心筋梗塞（胸痛）	CRITICAL	HIGH	! 過小トリアージ (-1)
CALL_102	チーンソーによる負傷	CRITICAL	CRITICAL	✓一致
CALL_103	小児の痙攣 (>5 分)	CRITICAL	CRITICAL	✓一致
CALL_104	喘息（重度）	HIGH	HIGH	✓一致
CALL_105	頭部外傷とワーファリン服用	HIGH	HIGH	✓一致
CALL_106	出産（発露）	CRITICAL	CRITICAL	✓一致
CALL_107	低血糖（糖尿病）	CRITICAL	HIGH	! 過小トリアージ (-1)
CALL_108	腕の骨折	LOW	MEDIUM	↑ 安全側へ (+1)
CALL_109	胃腸炎	LOW	HIGH	↑ 安全側へ (+2)
CALL_110	鼻出血（抗凝固剤服用）	LOW	HIGH	↑ 安全側へ (+2)
CALL_111	COPD 増悪	MEDIUM	HIGH	↑ 安全側へ (+1) / 氏名：該当なし
CALL_112	腎臓感染症	MEDIUM	HIGH	↑ 安全側へ (+1)
CALL_113	安否確認	LOW	HIGH	↑ 安全側へ (+2)
CALL_114	鼻に真珠が入った（子供）	MEDIUM	MEDIUM	✓一致
CALL_115	アルコール中毒	HIGH	HIGH	✓一致
集計パフォーマンス指標 (ESI レベル毎) :				
レベル	精度	GT 数	正解数	過大 / 過小
極めて緊急 (ESI-1)	60 %	5	3	0 / 2
緊急 (ESI-2)	100 %	3	3	- / 0
中等度 (ESI-3)	33 %	3	1	2→H / 0
低 (ESI-4/5)	0 %	4	0	4→M/H / 0
全体 (4 レベル)	46,7 %	15	7	6 / 2
±1 レベル 以内	80,0 %	-	12	-

トリアージは「安全第一」の原則に従っている（過大トリアージ 40.0%、過小トリアージ 13.3%）。CALL_101/107 (CRITICAL の代わりに HIGH) での誤分類はあるものの、保守的な格上げによって重大な適応症をカバーしている [Tan+07]。

5.3 情報抽出：アイデンティティ、SOAP、翻訳

表 5 は、フィールド抽出（複数パターン vs 正規表現）、臨床的 SOAP 評価（リッカート尺度 1-5）、および多言語対応（BLEU 英語 → 日本語）をまとめたものである：

5 評価と結果

表 5: 情報抽出: 属性フィールド、臨床的 SOAP 評価、翻訳品質

カテゴリ	フィールド/基準/メトリクス	正規表現	複数パターン	Δ	平均	σ	評価
属性	年齢	42 %	89 %	+47 PP	—	—	非常に良い
	氏名	68 %	93 %	+25 PP	—	—	非常に良い (一部欠落あり)
	住所	55 %	100 %	+45 PP	—	—	完璧
	電話番号	71 %	94 %	+23 PP	—	—	非常に良い
SOAP	網羅性 (S)	—	—	—	3.6	0.9	普通
	正確性 (S)	—	—	—	4.1	0.7	良い
	網羅性 (O)	—	—	—	3.4	1.0	普通
	正確性 (O)	—	—	—	3.9	0.8	良い
	臨床的妥当性	—	—	—	4.0	0.8	良い
	合計	—	—	—	3.8	0.8	良い
翻訳	BLEU (客観データ)	—	—	—	0.81	—	非常に良い
	BLEU (主観データ)	—	—	—	0.73	—	良い
	BLEU (アセスメント/計画)	—	—	—	0.69	—	普通
	キャッシュヒット率 (>90 %)	—	—	—	0.83s	—	リアルタイム

主な知見：複数パターンマッチングによりフィールド抽出が大幅に改善された（住所：100%）。臨床的妥当性（4.0/5）と翻訳品質（BLEU 0.81）は、実用性を裏付けている。

6 結論と展望

本研究で開発されたREDSSシステムは、コンピュータ支援救急医療における重要な進歩を示している。音声認識（Whisper）、大規模言語モデル（GPT-4o）、および臨床的に確立されたトリアージ基準（ESI）の相乗効果が、技術的に実現可能であるだけでなく、医療専門職にとって測定可能なプロセス上の負担軽減をもたらすことが実証された。ソースコード、デモビデオ、および検証データを含むプロジェクト全体は、GitHub で公開されている (<https://github.com/salmaelouar/REDSS-Emergency-AI>)。

6.1 プロジェクトの成果とリフレクション (NAIST)

当初定義されたすべての研究課題に対し、良好な結果が得られた。自動化されたSOAP 抽出が実装され、高い臨床的妥当性が証明された。ESI トリアージにおいて、システムは医学的に許容される許容範囲内で高い的中率を示し、価値のあるセカンドオピニオン（「保守的なセーフティネット」）としての適格性を備えていることが確認された。さらに、処理パイプラインの継続的なリアルタイム機能も正常に実証された。

日本の奈良先端科学技術大学院大学 (NAIST) での滞在は、世界トップクラスの研究環境において、フルスタック開発と LLM エンジニアリングに関する深い知見を得る貴重な機会となった。特に、プロンプト・チェーンの最適化はやりがいのある課題であり、異文化交流を通じて、グローバルに展開可能な医療ソフトウェア (EN/JA) の重要性が改めて浮き彫りとなった。

6.2 倫理的枠組みと将来性

本システムは **Human-in-the-Loop** の原則に厳格に従っており、最終的な決定権とアルゴリズムの透明性は、臨床的な信頼の前提条件として常に人間に委ねられている。将来の発展に向けて、具体的には以下のステップが推奨される：(1) 音声拡張（例：環境ノイズの注入）を用いた、10,000 件以上の実際のアノテーション付き救急通報データセットによるファインチューニング、(2) トリアージ決定に対する信頼度スコアの統合、(3) ドイツ語ローカライゼーションの実装、およびドイツと日本の救急指令センターでの A/B テストによる、実環境下での多言語対応の検証。

用語集

BLEU スコア

Bilingual Evaluation Understudy の略。参照テキストとの比較における n-gram の適合率に基づき、機械翻訳の品質を自動評価するための指標。

BroadcastChannel API

同じオリジンの異なる閲覧コンテキスト（ウィンドウ、タブ）同士が、リアルタイムで相互に通信することを可能にする Web インターフェース。

CUDA (指標)

AI が生成した内容の医療的正確性と網羅性を評価するための品質モデル。Completion (網羅性)、Understanding (理解度)、Directness (直接性)、Accuracy (正確性) から成る。

FastAPI

標準的な Python の型ヒントに基づき、非同期プログラミングをサポートする、API 作成のための現代的で高性能な Python 用 Web フレームワーク。

JSON

JavaScript Object Notation の略。人間とマシンの両方にとって読みやすく、アプリケーション間でのデータ交換のためのコンパクトなデータ形式。

n-gram (n グラム)

テキストシーケンスから抽出された n 個の連続した要素（例：単語）の列。統計的なテキスト分析や BLEU スコアなどの指標に使用される。

Pydantic

Python の型アノテーションに基づいた、データバリデーションおよび設定管理用のライブラリ。FastAPI において API リクエストの検証に使用される。

React

コンポーネントベースのアーキテクチャに基づき、仮想 DOM による効率的なレンダリングを可能にする、ユーザーインターフェース作成用の JavaScript ライブラリ。

React フック

関数コンポーネントにおいて状態（ステート）やライフサイクルメソッドを使用できるようにする、React 特有の関数（例：useState, useEffect）。

REST

Representational State Transfer の略。主に HTTP メソッド（GET, POST など）を介して通信する、システム間のステートレスなインターフェースのためのアーキテクチャスタイル。

SQLAlchemy

アプリケーションと SQL データベースの間に柔軟な抽象化レイヤーを提供する、強力な Python 用オブジェクト関係マッパー（ORM）。

SQLite

独立したサーバーインフラを必要としない、軽量でファイルベースのリレーションナルデータベース管理システム。

TTR

Type-Token-Ratio の略。単語の総数（トークン）に対する異なる単語の数（タイプ）の比率として計算される、テキストの語彙多様性の尺度。

WebSocket

クライアントとサーバー間の持続的な双方向通信を可能にするプロトコル。リアルタイム伝送（例：ライブ文字起こし）に使用される。

Whisper

OpenAI によって開発された、背景ノイズに対する堅牢性に特化した、トランスフォーマーベースの自動音声認識（ASR）モデル。

アルゴリズム

問題または一連の問題を、有限のステップで解決するための明確に定義された手順。

エンドポイント

URL によって識別される API のアクセスポイントで、特定のデータや機能を照会するために使用される。

オーバートリアージ（過大判定）

危険な判定不足（アンダートリアージ）のリスクを最小限に抑えるために、患者を医学的な客観的必要性よりも高い緊急度カテゴリーに用心深く分類すること。

コッパ係数（Cohen's Kappa）

偶然に一致する可能性を補正して、2人の評価者（例：人間と AI）の間の一致度を評価するための統計的尺度 (κ)。

コンテキスト・ウィンドウ

LLM が同時に処理できるトークンの最大数。これを超えると、会話の始まりの情報が失われる。

ゼロショット学習

学習中に獲得した一般知識に基づき、プロンプト内に具体的な例を事前に提示することなくタスクを解決するモデルの能力。

ディスパッチ (派遣・調整)

救急システムにおいて、出動車両や人員を現場に送り、調整するプロセス。

データマッパー・パターン

マッパーオブジェクトがオブジェクトとテーブル間のデータ転送を担当することで、ビジネスロジックをデータベース構造から厳密に分離するアーキテクチャパターン。

トランسفォーマー

アテンション（注意）メカニズムに基づいたニューラルネットワーク構成であり、GPT や Whisper などの現代的な言語モデルの基礎となっている。

トリアージ

救急状況において効率的なリソース配分を確保するために、患者の医療的治療の緊急性を判定する体系的な手順。

トークン

言語モデルが数学的に処理するために、テキストを分解した最小の情報単位（例：単語の一部や文字）。

ハルシネーション（幻覚）

生成型言語モデルにおいて、文法的には正しく聞こえるが、事実に反している、あるいは入力データに基づかないテキストをシステムが生成してしまう現象。

パイプライン

一連の処理ステップ（例：文字起こし、分析、分類）であり、あるステップの出力が次のステップの入力として機能するもの。

フレームワーク

アプリケーションの開発を加速・標準化するために、事前に定義された構造やツールを提供するプログラミング基盤。

プロンプト

AI モデルに対し、どのようなタスクを実行すべきかを指示するテキストによる入力命令。

プロンプト・エンジニアリング

大規模言語モデル（LLM）の出力品質を特定のタスクに合わせて向上させるために、入力命令（プロンプト）を意図的に設計および最適化するプロセス。

リッカート尺度

アンケートや臨床研究において、態度や満足度を測定するために頻繁に使用される、双極の尺度（多くは 5 段階）。

レイテンシ（遅延）

入力（例：緊急通報の終了）から結果の利用可能化（例：完成した SOAP ノート）までの間の時間的な遅延。

ワンショット/フューショット学習

プロンプトエンジニアリングにおける手法の一つで、モデルのコンテキストウインドウ内に、実行したいタスクの例を全く、あるいは非常に少数だけ提供すること。

仮想 DOM

React における実際の DOM のメモリ上のコピー。UI の変更を効率的に計算し、実際のブラウザ表示に対して必要な更新のみを行うことを可能にする。

再現率（Recall）

実際の正のケース全体に対する、正しく予測された割合（見落としの回避）。

堅牢性（ロバストネス）

背景の騒音や低い音声品質などの困難な条件下でも、システムやアルゴリズム（例：Whisper）が安定した結果を提供する能力。

待機モード（Idle Mode）

アクティブな緊急通報がない状態の透過モニターの待機状態。このモードでは、現在の日付と時刻がローカライズ（例：日本語または英語）されて表示される。

思考の連鎖（Chain-of-Thought）

最終的な回答を出力する前に議論の中間ステップを生成するようにモデルを導くプロンプティング手法。複雑なタスク（例：トリアージの理由説明）の品質を向上させる。

推論 (インフェレンス)

学習済みの AI モデルを新しいデータに対して運用し、予測、分類、または抽出を行うこと。

救急 / 緊急事態

生命や健康に対する差し迫った危険を示し、救急隊による迅速な介入を必要とする突然の出来事。

救急医療通信員 (EMD)

救急指令センターで、緊急通報の受付、医療用語を用いた初期判定、および救急隊の調整を担当する特別に訓練されたスタッフ。

文字起こし (トランスクリプション)

音声ファイルから話された言語を書き言葉のテキストに変換するプロセス。

正規表現 (Regex)

検索パターンを形成する文字列。コンピュータサイエンスにおいて、特定のパターン（例：電話番号、郵便番号）に沿ってテキストを検索または検証するために使用される。

緊急度分類器 (Urgency Classifier)

臨床基準とコンテキスト分析に基づいて、患者の医療的緊急度を自動的に分類するためのソフトウェアモジュール。

評価者間信頼性

同じ評価システムを使用する際に、異なる観察者または評価者（例：人間と AI）の間でどの程度一致しているかを示す統計的尺度。

適合率 (Precision)

AI の評価において、正の予測全体に対する正しく予測された割合（誤報の回避）。

非同期処理

オペレーションがメインのプログラムフローから独立して実行される形式であり、システムが AI 推論などの時間のかかるタスクの完了を待つ必要がないようにする。

参考文献

- [Ada+24] H. Adam, Y. Ming, A. J. Butte und L. A. Nathanson. „Clinical Information Extraction with Large Language Models: A Case Study on Organ Procurement“. In: *AMIA Annual Symposium Proceedings*. Bd. 2024. American Medical Informatics Association, 2024, S. 115–123.
- [BCK03] Len Bass, Paul Clements und Rick Kazman. *Software Architecture in Practice*. 2nd. Boston, MA: Addison-Wesley Professional, 2003.
- [BG21] Srinivas Bangalore und Sai S. Gullapalli. „Performance Analysis of Asynchronous Server Gateway Interface (ASGI) in Python Web Frameworks“. In: *International Journal of Advanced Computer Science and Applications* 12.8 (2021), S. 298–304.
- [Blo+19] Stig Nikolaj Blomberg, Fredrik Folke, Annette Kjær Ersbøll, Helle Collatz Christensen, Christian Torp-Pedersen, Michael R. Sayre, Jim R. Countiss und Freddy K. Lippert. „Real-time machine learning helps dispatchers identify out-of-hospital cardiac arrest“. In: *PLOS ONE* 14.1 (2019), e0210385. DOI: [10.1371/journal.pone.0210385](https://doi.org/10.1371/journal.pone.0210385).
- [CDM08] J. J. Clawson, K. B. Dernocoeur und C. Murray. *Principles of Emergency Medical Dispatch*. 4th. Priority Press, 2008.
- [CM10] Michael A. Covington und Joe D. McFall. „Cutting the TTR: Type-token ratio, language impairment, and language acquisition“. In: *Journal of Biomedical Informatics* 43.3 (2010), S. 471–476.
- [Fow03] Martin Fowler. *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2003.
- [Gil+20] Nancy Gilboy, Paula Tanabe, Debbie Travers und Anna Marie Rosenau. *Emergency Severity Index (ESI): A Triage Tool for Emergency Department Care, Version 4, Implementation Handbook, 2020 Edition*. Techn. Ber. Publication No. 12-0014. Rockville, MD: Agency for Healthcare Research und Quality (AHRQ), 2020.
- [Gup+25] R. Gupta, A. Gupta, R. Singh, A. Prasad und P. Bansal. „Large language models for data extraction from unstructured and semi-structured electronic health records: a multiple model performance evaluation“. In: *BMJ Health & Care Informatics* 32.1 (2025), e101139. DOI: [10.1136/bmjhci-2024-101139](https://doi.org/10.1136/bmjhci-2024-101139).

- [Koc+22] Ayca Koca, Onur Polat, Ahmet Burak Oguz und Mesut Sevindik. „Reliability and Validity of a New Computer-Based Triage Decision Support Tool: ANKU-TRIAGE“. In: *Disaster Medicine and Public Health Preparedness* 16.6 (2022), S. 2441–2445. doi: [10.1017/dmp.2022.101](https://doi.org/10.1017/dmp.2022.101).
- [KVS18] K. E. Klimley, V. B. Van Hasselt und A. M. Stripling. „Posttraumatic Stress Disorder in Police, Firefighters, and Emergency Dispatchers“. In: *Aggression and Violent Behavior* 43 (2018), S. 33–44. doi: [10.1016/j.avb.2018.08.005](https://doi.org/10.1016/j.avb.2018.08.005).
- [LeB09] V. R. LeBlanc. „The effects of acute stress on performance: implications for health professions education“. In: *Academic Medicine* 84.10 Suppl (2009), S25–S33. doi: [10.1097/ACM.0b013e3181b37b8f](https://doi.org/10.1097/ACM.0b013e3181b37b8f).
- [Luo+25] Xiao Luo, Le Zhou, Kathleen M. Adelgais und Zhan Zhang. „Assessing the Effectiveness of Automatic Speech Recognition Technology in Emergency Medicine Settings: A Comparative Study of Four AI-powered Engines“. In: *Journal of Healthcare Informatics Research* 9 (2025), S. 494–512. doi: [10.1007/s41666-025-00171-8](https://doi.org/10.1007/s41666-025-00171-8).
- [Mel+25] S. S. Meloy, E. Woltman, A. Martinez und K. Duane. „It’s time to talk to emergency medical dispatchers: survey study on performance feedback and patient outcome follow-up to EMDs“. In: *BMC Emergency Medicine* 25.1 (2025), S. 13. doi: [10.1186/s12873-025-01332-7](https://doi.org/10.1186/s12873-025-01332-7).
- [Mil56] George A. Miller. „The magical number seven, plus or minus two: Some limits on our capacity for processing information“. In: *Psychological Review* 63.2 (1956), S. 81–97.
- [Mis+18] Binoy Mistry, Sarah Stewart De Ramirez, Gabor Kelen, Paulo S. K. Schmitz, Kamna S. Balhara, Scott Levin, Diego Martinez, Kevin Psoter, Xavier Anton und Jeremiah S. Hinson. „Accuracy and Reliability of Emergency Department Triage Using the Emergency Severity Index: An International Multicenter Assessment“. In: *Annals of Emergency Medicine* 71.5 (2018), 581–587.e3.
- [Pap+02] Kishore Papineni, Salim Roukos, Todd Ward und Wei-Jing Zhu. „BLEU: a Method for Automatic Evaluation of Machine Translation“. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2002, S. 311–318.
- [Per+24] E. A. Perez-Alday, T. D. Li-Pershing, A. Bender, C. Hamilton, J. A. Thomas, K. Johnson, T. Lee, E. R. Gonzales, K. Taylor und L. G. Tereshchenko. „Leveraging Large Language Models for Accurate Retrieval of Patient Information From Medical Reports: Systematic Evaluation Study“. In: *Journal of Medical Internet Research* 26 (2024), e59803. doi: [10.2196/59803](https://doi.org/10.2196/59803).

- [Pin17] Jesse M. Pines. „What Cognitive Psychology Tells Us About Emergency Department Physician Decision-making and How to Improve It“. In: *Academic Emergency Medicine* 24 (2017), S. 1–5.
- [PLG23] Vivek Podder, Valerie Lew und Sassan Ghassemzadeh. „SOAP Notes“. In: *StatPearls [Internet]*. PMID: 29489268. Treasure Island (FL): StatPearls Publishing, 2023. URL: <https://www.ncbi.nlm.nih.gov/books/NBK482263/>.
- [Rad+23] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey und I. Sutskever. „Robust speech recognition via large-scale weak supervision“. In: *Proceedings of the 40th International Conference on Machine Learning*. Bd. 202. PMLR, 2023, S. 28492–28518.
- [Roa+11] Brian Roark, Margaret Mitchell, John-Paul Hosom, Kristy Hollingshead und Jeffrey Kaye. „Spoken Language Derived Measures for Detecting Mild Cognitive Impairment“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7 (2011), S. 2081–2090. DOI: [10.1109/TASL.2011.2112351](https://doi.org/10.1109/TASL.2011.2112351).
- [Sax+23] David R. Sax, E. Margaret Warton, Dustin G. Mark, David R. Vinson und Mamata V. Kene. „Evaluation of Version 4 of the Emergency Severity Index in US Emergency Departments for the Rate of Mistriage“. In: *JAMA Network Open* 6.3 (2023), e232404.
- [Sco+16] Greg Scott, Christopher Olola, Corike Toxopeus, Jeff J. Clawson, Adam Johnson, Brad Schultz, Kim Miller, Nicole Richmond, David Robinson, Matt Zavadsky, Leslie Burnette, Tracy Barron und Brian Patterson. „Characterization of Call Prioritization Time in a Medical Priority Dispatch System“. In: *Annals of Emergency Dispatch & Response* 4.1 (2016), S. 27–33. URL: <https://www.aedrjournal.org/characterization-of-call-prioritization-time-in-a-medical-priority-dispatch-system>.
- [Tan+07] Paula Tanabe, Richard Gimbel, Paul R. Yarnold, Demetrios N. Kyriacou und James G. Adams. „Reliability and validity of scores on the Emergency Severity Index version 3“. In: *Academic Emergency Medicine* 14.3 (2007), S. 213–218.
- [Wee68] Lawrence L. Weed. „Medical records that guide and teach“. In: *New England Journal of Medicine* 278.11 (1968), S. 593–600. DOI: [10.1056/NEJM196803142781105](https://doi.org/10.1056/NEJM196803142781105).
- [Wei+22] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le und D. Zhou. „Chain of Thought Prompting Elicits Reasoning in Large Language Models“. In: *Advances in Neural Information Processing Systems* 35 (2022), S. 24824–24837.

参考文献

- [WHA23] WHATWG. *HTML Living Standard: Web messaging*. Web Hypertext Application Technology Working Group. 2023. URL: <https://html.spec.whatwg.org/multipage/web-messaging.html>.
- [Zap+25] Jasmine S. Zaphir, Karen A. Murphy, Alex J. MacQuarrie und Matthew J. Stainer. „Understanding the Role of Cognitive Load in Paramedical Contexts: A Systematic Review“. In: *Prehospital Emergency Care* 29.2 (2025). DOI: [10.1080/10903127.2024.2370491](https://doi.org/10.1080/10903127.2024.2370491).
- [Zec+25] J. R. Zech, D. Kim, A. Somoano, I. Landi, E. S. Krauss, J. Kim, J. Gichoya, S. Soffer und E. Klang. „Transformer-Based Open-Source Whisper Software Versus Leading Commercial Speech Recognition Software for Radiology Transcription: Comparison Study“. In: *American Journal of Roentgenology* 225.1 (2025), e2532903.

Persönliche Angaben / Personal details

サルマ・エル・ワルギ・エル・マイジ

Familienname, Vorname / Surnames, given names

28.12.2000

Geburtsdatum / Date of birth

情報工学

Studiengang / Course of study

03895439

Matrikelnummer / Student registration number

Eigenständigkeitserklärung**Declaration**

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und noch nicht anderweitig für Prüfungszwecke vorgelegt habe. Ich habe keine anderen als die angegebenen Quellen oder Hilfsmittel benutzt. Die Arbeit wurde weder in Gänze noch in Teilen von einer Künstlichen Intelligenz (KI) erstellt, es sei denn, die zur Erstellung genutzte KI wurde von der zuständigen Prüfungskommission oder der bzw. dem zuständigen Prüfenden ausdrücklich zugelassen. Wörtliche oder sinngemäße Zitate habe ich als solche gekennzeichnet.

Es ist mir bekannt, dass im Rahmen der Beurteilung meiner Arbeit Plagiatserkennungssoftware zum Einsatz kommen kann.

Es ist mir bewusst, dass Verstöße gegen Prüfungsvorschriften zur Bewertung meiner Arbeit mit „nicht ausreichend“ und in schweren Fällen auch zum Verlust sämtlicher Wiederholungsversuche führen können.

I hereby certify that I have written this thesis independently and have not submitted it elsewhere for examination purposes. I have not used any sources or aids other than those indicated. The work has not been created in whole or in part by an artificial intelligence (AI), unless the AI used to create the work has been expressly approved by the responsible examination board or examiner. I have marked verbatim quotations or quotations in the spirit of the text as such.

I am aware that plagiarism detection software may be used in the assessment of my work.

I am aware that violations of examination regulations can lead to my work being graded as "unsatisfactory" and, in serious cases, to the loss of all repeat attempts.

Unterschrift Studierende/Studierender / Signature student

日本・生駒市, 28.02.2026

Ort, Datum / Place, date