

進化的メンテナンス:医療用医薬品文書検査ルール of データ駆動型 自動保守フレームワーク

Evolutionary Maintenance: A Data-Driven Automated Maintenance Framework for Prescription Drug Document Inspection Rules

八角 潤一^{*1} 小泉 直子^{*1} 藪 紘明^{*1} 後藤 誠^{*1} 後藤田 達哉^{*2} 田中 和哉^{*2*3}
Junichi Yasumi, Naoko Koizumi, Hiroharu Yabu, Takashi Goto, Tatsuya Gotoda, Kazuya Tanaka

^{*1} 株式会社 博報堂メディカル ^{*2} scheme verge 株式会社 ^{*3} 政策研究大学院大学
Hakuhodo-Medical Inc. scheme verge, Inc. GRIPS

In our previous work [1], we established auditability for pharmaceutical document inspection through a deterministic rule-based architecture. However, maintaining approximately 360,000 cross-check combinations per document proved unmanageable by human effort alone, as rule enhancements to improve recall structurally increased false positives. To address this maintenance bottleneck, we propose an Evolutionary Maintenance framework comprising three components: (1) an automated diagnostic engine that deterministically classifies false-positive causes into CONFIG, CODE_BUG, and STRUCTURE categories without relying on LLMs; (2) a multi-rater ground-truth labeling system that distinguishes evaluated from unevaluated rules; and (3) a unified diagnostic UI integrating inspection runs with cause analysis. Empirical evaluation on an internal demonstration document with approximately 320 inspection rules showed the diagnostic engine classified 99.1% of false positives as system-correctable, while recall improved from 50.0% to 85.4% over the development period. The labeling system further refined apparent precision from 27.7% to 93.2% by excluding unevaluated rules.

1. はじめに

1.1 前報の到達点と本研究の目的

同時投稿の前報 [1] では、医療用医薬品のプロモーション資料の検証において、構造抽出と論理検証を機能的に分離した Twin-Track Strategy ハイブリッドアーキテクチャを提案した。本手法は 320 件の検証ルールを決定論的なロジックツリーとして実装し、ハルシネーションを原理的に排除した。実証実験では専門家の検証時間を 95–99%削減し、必須修正項目を実用レベルの再現率(検出すべき項目のうち実際に検出された割合)で検出した [1]。

しかし、この決定論的アプローチは、過検出を許容する設計を採用しており、適合率(検出結果のうち正しい検出の割合)は初期値で 27.7%にとどまった。すなわち本手法で検出した評価結果の約 7 割が過検出であり、この過検出の原因特定と修正が実運用上の最大のボトルネックとなっていた。

本研究では、この保守問題を自動化する「進化的メンテナンス」アーキテクチャを提案し、その有効性を実証する。

1.2 開発過程で直面した問題

ルール保守において、以下の 3 つの問題が深刻であった。

問題 1: 組み合わせ爆発。

前報 [1]で示した通り、本システムの各ルールは「キーワードリスト L 、文書 D 、対象セクション S 、評価関数 f 」から構成される条件ブロック C の論理結合として定義される。本システムでは約 320 件のアクティブルールが約 500 個の条件ブロックを持ち、数十種のキーワードリスト(延べ約 14 万語)と約 100 種の文書セクションを照合する。全組み合わせの理論上界 $|L| \times |D| \times |S| \times R$ は数十億回に達するが、各条件ブロック C が参照するセクション集合

は限定的であり、実測による実効照合回数は 1 文書あたり約 36 万回となる。この 36 万回の照合空間から過検出の原因を人手で特定することは事実上不可能である。

問題 2: 再現率改善に伴う過検出の構造的増加。

医療情報は人の生命・健康に直接影響を及ぼしうるクリティカルな領域であるため、高い正確性と説明可能性が要求され、誤情報やハルシネーション(幻覚)の排除は不可欠である。そのため、本システムは見落とし(FN: False Negative)を最小化するために過検出(FP: False Positive)を許容する Safety First 設計を採用しており、各ルールの評価関数は意図的に高感度に設定されている。[2] の一般式 $C_i = f_i(L_i, D_i(S_i))$ において、再現率を向上させるにはルール数 R の追加、キーワードリスト $|L_i|$ の拡張、または対象セクション $|S_i|$ の拡大が必要となる。さらに各ルールは独立に判定を行い、あるルールの過検出を別のルールが抑制する機構はない。したがって、高感度設計の上でルールを拡張すると、ルール追加は新たな過検出の源となり、照合空間の拡大は誤合致の機会を増やすため、過検出数は構造的に単調増加する。実測でも再現率を 50.0%から 85.4%に改善する過程で過検出が 67 件から 107 件に増加し(表 2)、この適合率・再現率のトレードオフが設計・実態の両面で確認された。

問題 3: 修正のリグレーション。

1 件のルール修正が他のルールの判定結果に波及するリグレーションが頻発した。例えば、キーワード照合型ルールの過検出を 5 件削減する修正が、セクション存在確認型ルールの過検出を 19 件増加させる事例が発生した。主要な修正 4 件中 3 件で意図しない副作用が生じ、人手による個別修正では収束が困難であった。

これらの問題に対し、本研究では 3 つの対処を行った。

連絡先: 八角 潤一, 株式会社 博報堂メディカル, 東京都港区赤坂

6-2-4 S-GATE 赤坂 6 階, <https://www.hakuhodo-medical.co.jp/contact/>

2. 提案手法

2.1 進化的メンテナンス・アーキテクチャ

提案アーキテクチャは、検査器 (Inspector)、診断器 (Doctor)、修正器 (Mechanic) の 3 モジュールと、人間 (Human) から構成されるフィードバックループである (図 1)。人間は個々の過検出の原因調査から解放され、診断エンジンが提示する修正候補の承認に集中する。診断器の分類精度は、人間が提供する正解ラベルと原因確定により段階的に向上する。

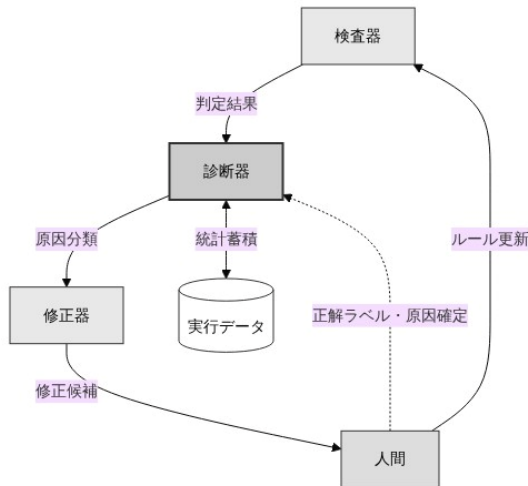


図 1: 進化的メンテナンス・アーキテクチャ

本手法は LLM を一切使用せず、実行データの統計的蓄積と決定論的パターンマッチングのみで動作する。これにより、診断結果自体にハルシネーションが混入するリスクを排除し、前報 [1] が確立した監査可能性をメンテナンスプロセスにも拡張する。

以下、1.2 節の 3 つの問題に対する 3 つの対処を順に述べる。

2.2 対処 1: 過検出・見落とし原因の自動診断

(1) 原因推定エンジン

過検出および見落としの原因を設定ミス、コードバグ、文書構造問題、OCR 未対応、分類不能の 5 カテゴリに自動分類する。分類は以下の 2 段階の決定論的推定器による。

Noise Probability Estimator (NPE) : ルール ID ごとに $P_{noise}(r) = FP(r) / (FP(r) + TP(r))$ を算出し、ルール固有の誤発火傾向を推定する。複数 Run (検査実行) の累積データに Laplace smoothing (データが少ない初期段階で極端な推定値を抑制する平滑化手法) を適用し、Run 数の増加に伴い推定精度が向上する設計である。なお本文中の TP と FP は、それぞれ正検出 (True Positive)、過検出 (False Positive) を意味する。

Code Bug Probability Estimator (CBPE) : NPE がシステムエラーと判定した過検出について、複数のパターン検出器 (キーワード存在検証・セクション整合性・ファミリー相関分析) で原因カテゴリを絞り込む。

見落としについても同様に、原因カテゴリ (コードバグ、文書構造問題、設定ミス、OCR 未対応、未分類) を自動推定する。複数の条件ブロックを持つルールでは診断エンジンが全条件を評価し、優先度順 (コードバグ > 文書構造 > 設定ミスの順) で最も深刻な原因を自動選択する。さらに、ルールに設定されたアラート重大度に基づき「要修正」「要注意」「任意」の 3 段階で修正優先度を提示する。

2.3 対処 2: 正解データのラベル付けシステム

検査ルールの正解データを、複数評価者の視点で管理するシステムを構築した。複数評価者方式とは、同一の検査結果に対して異なる立場の評価者がそれぞれ正解ラベルを付与し、視点ごとの精度指標を算出する手法である。

(1) 6 値分類

従来の正検出・過検出・正棄却・見落としの 4 値分類では、期待結果が定義されていないルールが暗黙に過検出または正棄却として扱われ、精度指標が実態を反映しないという問題があった。

本研究では、4 値に加えて「ラベル未定義・発火」と「ラベル未定義・非発火」の 2 値を追加した 6 値分類を導入した (表 1)。これにより、未評価のルールを明示的に分離し、評価済みルールのみで正確な精度評価が可能となった。

	ラベルあり (期待結果定義済)	ラベルなし
発火	正検出 (TP) 過検出 (FP)	ラベル未定義・発火
非発火	正棄却 (TN) 見落とし (FN)	ラベル未定義・非発火

表 1: 6 値分類

最終評価における 320 ルールの内訳は、ラベル済み 56 件 (正検出 41 件、過検出 3 件、正棄却 5 件、見落とし 7 件)、ラベル未定義・発火 108 件、ラベル未定義・非発火 156 件であった。従来方式 (全ルール評価) では適合率 27.7% であったのに対し、6 値分類によりラベル済みルール限定で適合率 93.2% となり、実態に即した精度評価が実現された。

2.4 対処 3: UI 統合による Run と診断の一体化

上記の自動診断と正解ラベル管理を、検査実行単位である Run と一体化した Web ベースの統合 UI を構築した。Run 一覧・診断結果詳細・複数評価者比較・PDF レポート出力を一面で提供し、Run 実行から修正優先度の判断までを完結させる。

3. 実証結果

3.1 実験設定

前報 [1] と同一の社内デモ用プロモーション資材 (総合概要、320 ルール) を対象に、ドメイン専門家による手動ラベリングを評価基準として実験を実施した。前報時点のベースラインから最終評価まで、数百回の Run を実行した。

3.2 過検出原因の自動分類結果

全量評価における過検出 107 件の原因内訳は、設定ミス 88 件 (82.2%)、コードバグ 18 件 (16.8%)、OCR 未対応 1 件 (0.9%) であった。このうち設定ミスとコードバグの計 106 件 (99.1%) はシステム修正可能である。

見落とし 7 件についても原因分類が付与され、設定ミス 3 件、コードバグ 3 件、OCR 未対応 1 件であった。

3.3 メトリクス推移

指標	ベースライン	全量評価	変化
正検出	26	41	+15
過検出	67	107	+40
正棄却	193	161	-32
見落とし	26	7	-19
再現率	50.0%	85.4%	+35.4pt
適合率	28.0%	27.7%	-0.3pt

表 2: ベースラインからの改善(対象 316 件, 4 値分類)

表 2 はデフォルト発火ルール 4 件を除く 316 件(ベースライン時は 312 件)を対象とした 4 値分類である。3.5 節の 6 値分類ではこの 4 件を含む全 320 件を対象とする。また、開発過程でドメイン専門家が期待結果を精査し、「非発火が正解」と判明した 4 件を正棄却(TN)に再分類したため、再現率の分母(TP+FN)はベースラインの 52 件から最終評価の 48 件に変動している。

コードバグの修正と設定ミスの適正化により、再現率はベースラインから 35.4 ポイント改善し、見落としは 26 件から 7 件に 73% 削減された。一方、適合率は再現率改善に伴う過検出の増加により横ばいとなった(表 2)。この過検出増加の内訳を 3.2 節で示した通り、99.1%がシステム修正可能であることが自動診断により特定されている。

3.4 診断駆動型修正サイクルの検証

診断エンジンがコードバグと分類した見落とし 5 件に対しコード修正を実施したところ、5 件全てが正検出に遷移し、既存の正検出からの劣化は 0 件であった。診断エンジン導入前は主要修正 4 件中 3 件でリグレーションが発生しており(リグレーション率 75%)、改善は顕著である。また、見落としの原因分類は Run データの蓄積に伴い精度が向上し、確定分類 12 件中 12 件が正確に分類された(初期の正答率 0%→100%)。これにより修正対象の優先順位付けが実用的に機能した。

3.5 6 値分類導入の効果

同一の検査結果に対し、従来方式(全 320 ルール評価)では正検出 41 件、過検出 107 件、適合率 27.7%であったのに対し、6 値分類(ラベル済み 56 件限定)では正検出 41 件、過検出 3 件、適合率 93.2%となった。

6 値分類により、ラベルが定義されていない 264 件のルールが「ラベル未定義」(発火 108 件 + 非発火 156 件)として明示的に分離された。これにより、従来暗黙的に正棄却・過検出としてカウントされていた未評価ルールが除外され、適合率が実態を正確に反映するようになった。再現率は両方で 85.4%と同一であり、正検出と見落としはラベル済みルールに集中しているため、6 値分類の導入が再現率評価に影響しないことが確認された。

4. 考察

4.1 自律的保守システムとしての位置づけ

提案アーキテクチャの検査器→診断器→修正器→人間のフィードバックループは、Kephart & Chess [2] が提唱した自律コンピューティングの MAPE-K (Monitor-Analyze-Plan-Execute over Knowledge) ループの構造と類似する。しかし、本手法は医療用医薬品領域の規制の観点から意図的に以下の 2 点で異なる実装戦略をとった。第一に、分析・計画フェーズにおいて確率的

推論を排し、決定論的アプローチを採用した点である。第二に、修正実行の直前に人間の承認プロセスを必須要素として組み込み(Human-in-the-loop)、変更管理の透明性と責任の所在を明確化した点である。これにより、診断結果自体の監査可能性が担保される。実証結果(3.2 節)では過検出の 99.1%がシステム修正可能と自動分類され、修正サイクル検証(3.4 節)ではリグレーションを完全に抑制した。これは、人間の役割を「原因調査」から「修正承認」へ移行させ、36 万回の照合空間における保守作業を実用的に成立させたことを意味する。

4.2 データ蓄積による診断精度の漸進的向上

NPE の Laplace smoothing は、Run データの蓄積に伴い推定値が真の値へ収束する設計である。実証では、確定分類の正答率が初期 0%から最終 100%(12 件中 12 件)へ向上した。この漸進的改善は LLM ベースの診断アプローチとの重要な差異である。LLM による原因推定はプロンプトやモデル更新に依存し、精度の継続的改善が保証されない。一方、本手法は Run 数の増加が統計的推定の信頼性を着実に向上させるため、運用を継続するほど診断精度が改善される特性を持つ。この特性は、医薬品規制のように長期運用と継続的改善が求められる領域において、実用上の大きな利点である。

4.3 制約と今後の課題

本研究にはいくつかの制約がある。第一に、修正器(Mechanic)による修正パッチの自動生成は未実証であり、現時点では人間が修正を実施している。第二に、実証実験は 1 文書・320 ルールに限定されており、複数文書への汎化性検証が必要である。第三に、ラベル済みルールは 320 件中 56 件(17.5%)にとどまり、ラベルカバレッジの拡大が全体精度の把握に不可欠である。

5. おわりに

本研究では、決定論的ルールベース検査における組み合わせ爆発下の保守問題に対し、「進化的メンテナンス」フレームワークを提案した。本フレームワークは LLM を使用せず決定論的に動作するため、前報 [1] が確立した監査可能性を保守プロセスにも拡張するものである。

参考文献

- [1] 小泉直子ほか, 生成 AI 時代の医療プロモーション資材における構造・論理分離アプローチ: 監査可能性の確立と検証プロセスの「質と量」の両立. 人工知能学会全国大会(第 40 回), 2026. (同時投稿)
- [2] Kephart, J.O. and Chess, D.M. The vision of autonomic computing. *Computer*, 36(1), 41-50, 2003.