Convergence Lab. Whitepaper

現場 AI は SoTA だけでは決まらない

データと評価の実装戦略

画像・自然言語・音声の実装例から考える
データファースト AI の設計と運用

Convergence Lab. 株式会社
代表取締役 社長 木村 優志
kimura@convergence-lab.com
2026 年 8 月 19 日

1. エグゼクティブ・サマリー

AI 導入の相談では、「最も精度の高いモデルはどれか」という問いから議論が始まりやすい。画像認識なら最新の Vision Transformer、自然言語処理なら大型の基盤モデル、音声認識ならベンチマークで上位のモデルが候補に並ぶ。最新技術を追うことは重要である。しかし、研究用データセットで優れた数字を示すことと、個別の現場で継続して成果を出すことは同じではない。

論文や公開ベンチマークの精度は、定義されたデータ、正解ラベル、評価方法、計算条件の上で測定された数字である。その条件が実運用と異なれば、数字をそのまま移植することはできない。学習時と運用時のデータ分布が異なる「dataset shift」は、機械学習を実環境へ適用する際の基礎的な問題として整理されている 1。現場には、照明の交換、設備の摩耗、新製品、担当者ごとの判断差、文書改訂、方言、騒音、入力形式の乱れといった、ベンチマークに含まれない変化が存在する。

本稿の主張は、「データが常にモデルより重要である」という単純なものではない。新しいモデルは、少量データでの学習、推論速度、扱える入力の範囲を改善し、従来は難しかった課題を可能にする。SoTA（State of the Art：その時点での最高性能）を知ることは、試すべき選択肢を増やすために不可欠である。一方、導入の成否と再現性を強く制約するのは、現場を代表するデータ、合意された正解、業務上の失敗を含む評価セット、そして運用から失敗を戻す仕組みである。

本稿では、画像処理、自然言語処理、音声処理、外観検査、RAG（Retrieval-Augmented Generation）、Text-to-SQL を例に、モデルを比較する前に整えるべきデータ設計を解説する。結論は明確である。企業が蓄積すべき資産は、モデル名ではない。自社の業務で何を正解とし、どの条件で失敗し、どう改善するかを記録したデータと評価の循環である。

2. なぜ SoTA だけでは現場の問題を解けないのか

2.1. 論文の数字は、そのデータセット上の数字である

SoTA とは、特定のベンチマーク、指標、評価手順、計算条件で最もよいと報告された結果を指す。研究開発において有用な比較軸だが、データ収集、ラベル、分割、前処理、計算資源の前提から切り離して採用理由にはできない。

実運用においては、入力分布、出力と入力の関係、あるいはデータを取得する過程が変わりうる。季節で日射や需要が変わる、カメラを交換する、製品型番が追加される、業務規程が改定される、利用者が新しい言い回しを使うといった変化は、その例である。dataset shift に関する研究は、学習・テスト・運用の条件差を明示し、学習時の評価だけでは実環境での性能を十分に説明できないことを示している 1。

領域	ベンチマークで固定されやすい条件	現場で追加して確認すべき条件
画像処理	撮影距離、照明、背景、対象カテゴリ、ラベル定義	カメラ個体差、振動、汚れ、反射、型番追加、昼夜・季節、設置位置
自然言語処理	公開コーパス、正規化済みの文書、固定された正解	社内略語、表・スキャン PDF、規程改定、入力の欠落、権限による参照範囲
音声処理	収録環境、話者属性、語彙、マイク、発話スタイル	騒音、方言、専門用語、同時発話、通信品質、マイク配置
外観検査	撮影済みの良否画像、既知の欠陥種類	不良の希少性、水滴・油・傷、設備状態、許容差、検査員の判断基準

2.2. 「モデルの精度」と「業務の成果」は異なる

業務で必要なのは、平均精度が高いことだけではない。見逃してはいけない重大な不良を検出できるか、情報が不足した質問に断定せず回答不能と返せるか、誤った SQL で二重計上しないか、重要な問い合わせを誤分類しないかが問われる。平均値が改善していても、高リスクの一部条件で悪化していれば、導入判断としては不適切になりうる。

モデルの評価条件、用途、既知の制約を明示する Model Cards の提案は、単一の性能指標だけで判断せず、対象利用に応じた評価と限界の説明を求めている 2。同様に、NIST の AI Risk Management Framework（AI RMF）は、AI のリスクを組織の文脈に合わせて Govern、Map、Measure、Manage することを示す 3。現場の成果は、モデル単体の性能ではなく、データ、業務フロー、人の判断、監視、変更管理を含むシステム全体の性質で決まる。

3. データファースト実装の 4 条件と運用実践

3.1. 代表性：解きたい状況がデータに含まれているか

最初に確認すべきは件数ではなく、解きたい問題がデータに現れているかである。100 万件の正常画像があっても、重要な不良、照明変化、型番差、設置位置の差がなければ、それらへの性能を評価できない。数十件でも、失敗しやすい条件、境界条件、例外を意図的に含めていれば、PoC としての実現可能性を判断できる場合がある。

代表性は、統計的な抽出だけを意味しない。対象業務で発生する頻度、失敗した際の影響、将来変わる可能性を踏まえ、必要な条件を設計する作業である。安全・品質・金額・契約に関わる低頻度だが高影響の事象は、自然に集まるのを待つのではなく、評価ケースとして意図的に確保する。

3.2. 正解定義：ラベルは業務上の合意を写す

教師あり学習で使うラベルは、単なる前処理ではない。外観検査の「傷」、音声認識の専門語表記、問い合わせの緊急度、RAG の「現行文書」、売上の定義は、いずれも業務上の判断を機械可読な形にしたものである。

同じ画像を見た検査員が異なる判定をする、部門ごとに「売上」の計上時点が異なる、規程の草案と正式版が同じフォルダに置かれている、といった状態では、モデルが一貫した正解を学べない。判断が分かれたデータを先に集め、なぜ分かれたのかを現場の責任者と確認する。必要ならラベルを細分化し、「要確認」「対象外」「根拠不足」といった区分を設ける。この合意形成が、後のモデル精度の上限を決める。

データセットの目的、構成、収集方法、推奨用途、既知の制約を文書化する Datasheets for Datasets は、データを適切に使い、再現性と説明責任を高めるための実務的な枠組みを提示している 5。少なくとも、誰が、何の判断のために、どの期間・条件でデータを集め、どの変換を行ったかを残すべきである。

3.3. 評価分離：学習データと同じ問題で自分を採点しない

評価データは、学習に使わない。さらに、同じ設備、同じ利用者、同じ文書の改訂前後、同じ動画の連続フレームのように、互いに似すぎたデータが学習用と評価用に混ざらないようにする。混入があれば、未知の条件への強さではなく、類似データへの再現能力を測ってしまう。

評価には異なる目的のデータを混ぜない。本番比率を反映する評価と、重大な失敗を意図的に集める評価は、別の指標として報告する。数十件のケースは失敗の発見には使えるが、希少事象を見逃さないことの統計的な保証には通常十分ではない。

3.4. 来歴と版管理：データは変わるものとして扱う

運用中の AI では、学習データ、評価セット、ラベル基準、特徴量、検索インデックス、モデル、プロンプト、ツール定義が変化する。変更した結果、何が改善し何が悪化したかを追うために、データの版と実験条件を残す必要がある。

NIST AI RMF は、データの出所、変換、ラベル、依存関係、制約、メタデータを記録することを、透明性と説明責任の観点から挙げている 3。すべてを大規模な基盤で始める必要はない。小規模なプロジェクトであっても、データの取得日・対象範囲・抽出条件・ラベル基準・評価実行日・モデル版を表計算やリポジトリで残すだけで、調査可能性が大きく改善する。

3.5. 運用実践：品質ゲート、負荷、変化を管理する

運用では、スキーマ・欠損・値域・リークを検査し、異常時の隔離・確認・停止を決める 8。収集目的、保存期間、アクセス権、現場のラベル付け負荷も同時に設計する。件数目標を先に置くのではなく、「どの条件で品質が不足するか」という不確実性を減らすデータを優先する。

4. 評価セットを業務の合格基準にする

4.1. 最初に「許容できない失敗」を書き出す

評価セットを作る前に、次の三つを業務責任者と合意する。

- AI に何を判断・生成・提案させるか。
- 正しい結果とは何か。根拠、形式、対象範囲、回答不能時の挙動を含める。
- 許容できない失敗は何か。その失敗が起きた場合、誰が止め、誰が確認するか。

たとえば外観検査なら、「重大欠陥を見逃さない」「判定が不確かなものを人へ回す」が重要になりうる。RAG なら、「現行版以外を根拠にしない」「根拠がない場合は回答しない」が必要になる。Text-to-SQL なら、「定義されていない指標を推測しない」「複数テーブルの結合で二重計上しない」が合格条件になる。

評価セット	含めるべき例	用途と扱い
通常評価セット	本番の頻度に近い日常入力	通常時の品質を測る。母集団比率と分け方を記録する。
ストレスセット	重大、境界、異常、安全性ケース	失敗しやすい条件を検出する。通常評価の平均点へ混ぜない。
封印ホールドアウト	開発・調整に使わない代表データ	リリース候補を限定回数で評価する。失敗例を学習へ回す場合は補充する。
時系列・OOD セット	新型番、照明交換、規程改定、新用語	未来の変化への強さと、運用後の劣化を確認する。

4.2. 採点方法をタスクに合わせて選ぶ

分類、ID 抽出、必須項目、JSON 形式のように機械的に判定できるものは、完全一致、集合一致、構造検証、ルール検証を優先する。自由文、要約、業務上の妥当性のように単一の正解がないものは、明示したルーブリックに従って人がレビューする。LLM を採点補助に使う場合も、人による採点との一致・不一致を確認し、採点モデル、プロンプト、基準の版を記録する。

大切なのは平均点だけを見ないことである。高リスクケースを一件でも失敗した場合にリリースを止めるのか、人のレビューを必須にするのかを事前に決める。NIST AI RMF も、実際の配備条件に近い環境での測定、結果の文書化、継続的な監視を重視している 3。

4.3. RAG とエージェントは中間工程も評価する

RAG では、回答が自然でも、古い規程や無関係な文書を根拠にしていれば問題である。質問と期待回答だけでなく、参照すべき文書、避けるべき文書、必要な根拠箇所を評価ケースへ持たせる。回答品質と検索品質を分離することで、「LLM が悪い」のか「必要な根拠を渡していない」のかを切り分けられる。

外部ツールを呼び出すエージェントでは、最終文だけで十分ではない。ツール選択、引数、実行順序、権限確認、人の承認を求めるタイミングを別々に評価する。操作を伴う業務ほど、誤った実行を防ぐ評価と Human-in-the-Loop が必要になる。

4.4. 評価セットは、変更履歴を持つ運用資産である

失敗例は回帰テスト用のストレスセットへ追加する。一方、開発者が繰り返し最適化するセットは、独立した性能証明には使えない。封印ホールドアウトは利用を限定し、学習または調整へ使ったケースは新しいデータで補充する。

各ケースには、出所、想定リスク、期待結果、採点方法、レビュー担当、データ・指標・モデルの版を記録する。RAG なら検索対象文書の版、Text-to-SQL なら指標定義の版を含める。変更ごとに通常・ストレス・封印・時系列の結果を分けて比較する。

5. 実装例：データ設計が成果を左右する領域

5.1. 画像処理：撮影条件とアノテーションを仕様にする

画像処理の失敗は、モデルの層数より前に、撮影条件の違いから起きることが多い。学習画像では対象物が中央にあり、現場では振動でぶれる。評価時には一定の照明で、実運用では照明交換や外光の影響を受ける。背景、カメラ、レンズ、距離、角度、露出、圧縮率の変化も、入力分布を変える。

導入時には、画像そのものに加え、撮影日時、設備、カメラ、レンズ、照明、対象型番、設置位置、工程、オペレーターなどのメタデータを取る。誤検出と見逃しを、このメタデータで集計できるようにすると、どの条件のデータを追加すべきかが見える。アノテーションでは、対象範囲、隠れ・重なり、境界が曖昧な物体、除外対象をルールとして定義し、判断が割れた事例を残す。

実装パターン：エッジ **AI** によるキーポイント検出
匿名の過去案件で採用したパターンである。キーポイント検出では、どの位置を点として定義するかが教師データそのものになるため、モデル交換の前にアノテーション基準と現場画像の整備を行った。

> 実装上のポイント
>
> 対象：エッジ AI 環境での画像認識・キーポイント検出
> 課題：初期モデルの検出精度が実運用の要件に届かなかった
> 対応：現場画像に対するデータアノテーションを強化
> 検証：キーポイント定義、評価指標、設備・撮影条件ごとの誤差を分けて確認する

数値、評価データの分割、モデル固定条件を公開できない案件では、改善率を主張しない。代わりに、キーポイント誤差の定義、データの取得条件、評価方法を先に合意する。

5.2. 外観検査：希少な不良と「異常」を区別する

製造現場では、品質管理が良いほど不良画像が少ない。このため、不良品を大量に集める前提の教師あり学習は、開始時点で行き詰まることがある。良品のみを学習する異常検知は有力な選択肢だが、水滴、油、影、汚れ、反射、部材位置のずれなど、業務上は許容すべき変化まで異常として検出する場合がある。

重要なのは、「通常と違う」ことと「不良である」ことを同一視しないことである。重大な欠陥、軽微だが要確認の変化、許容する外乱、撮影不良を区別した評価セットを作る。対象物・欠陥・撮影条件を言葉で指定できる Vision-Language Model や、少量の現場データを用いた追加学習を検討する場合も、現場の許容基準をラベルと評価へ反映しなければ、導入後の誤検知を減らせない。予知保全・製造 AI の研究レビューでも、方式選定だけでなく、設備データ、故障定義、評価指標、実際の運用条件への適合が重要な論点とされている 6 7。

実装パターン：**CutPaste** による疑似異常データの活用
実不良が限られる案件で採用したパターンである。CutPaste は、正常画像の一部を切り取り、別の位置へ貼り付けることで局所的な不規則性を作り、正常画像と区別する自己教師ありの表現学習に利用する手法である 10。

> 実装上のポイント
>
> 対象：外観検査における異常検知
> 課題：実際の異常データが少なく、学習に必要な異常の多様さを確保しにくい
> 対応：正常画像から CutPaste で疑似異常データを生成
> 検証：未加工の実画像、可能であれば実不良を含むデータで性能を評価する

ここで重要なのは、CutPaste を実不良データの代替と扱わないことである。切り貼りで作る不規則性は、実際の傷、欠け、汚れ、反射などを完全に再現するものではない。そのため、疑似異常は異常を見分ける表現を学ぶための補助として使い、最終的な導入判断は、現場で得た実画像、可能であれば実不良を含む評価セットで行う必要がある。どの大きさ・形状・位置のパッチを使うかも、対象物の形状と実際に起きる異常を踏まえて決めるべきである。

5.3. 自然言語処理：文書の鮮度と業務用語を管理する

自然言語処理では、入力の文章だけでなく、文章がどの業務上の意味を持つかがデータ品質を左右する。たとえば「売上」は受注日、出荷日、請求日、入金日、返品控除後の金額など複数の定義を持ちうる。社内文書には、正式版、改定前のコピー、会議資料、個人のメモが同居する。文章が読めても、どれが正本で、いつ有効で、誰が参照できるかが不明なら、業務で正しい回答は作れない。

したがって、文書を AI へ渡す前に、文書種別、主管部署、有効開始日・終了日、版、正本かどうか、アクセス権、更新元を管理する。社内略語、製品名、部署名、表記ゆれは、辞書や例示として収集する。分類や抽出では、正解ラベルに加え、判断不能、複数カテゴリ、要確認といった出力を定義する。モデルを大きくしても、曖昧な業務定義や古い正本を自動的に正せるわけではない。

実装パターン：自然言語分類におけるデータバランスの点検
カテゴリごとのデータ数に偏りがある案件で採用したパターンである。全体の平均精度だけでは少数カテゴリの品質が見えにくいため、データ量、入力の多様さ、誤分類の影響をカテゴリ別に点検する。

> 実装上のポイント
>
> 対象：自然言語のカテゴリ分類
> 課題：カテゴリ間のデータ数に偏りがあり、一部カテゴリの精度が課題となった
> 対応：カテゴリごとのデータバランスを見直した
> 検証：カテゴリ別の再現率・適合率、混同行列、実運用の閾値で確認する

単に件数を均等にするのではなく、業務上の重要度と本番の出現率に基づき、再収集、重み付け、閾値調整などを比較する。不均衡学習では、手法選定と評価指標を分けて設計する必要がある 12。

5.4. RAG：回答品質の前に検索結果を確認する

RAG は、社内文書全体を毎回読む仕組みではない。質問に関連すると判断した一部のチャンクを検索し、その範囲を LLM へ渡して回答を生成する。必要な根拠が検索されなければ、高性能な LLM でも正しい回答は難しい。

RAG の評価では、質問に対して、正しい文書が上位に検索されたか、現行版が優先されたか、無関係な文書を根拠にしていないかを先に見る。原本管理、PDF・表・画像からの抽出、分割単位、メタデータ、索引更新、検索方式は、モデル選定と同程度に重要である。更新後の文書が索引に反映されたか、失効文書が検索対象から除外されたかを運用監視に組み込む。

実装パターン：**JSON** カテゴリと **ICL** を用いた質問整理
質問が多様な RAG 案件で採用したパターンである。業務上の質問カテゴリ、説明、代表例を JSON で整備し、LLM へ文脈として渡す In-Context Learning（ICL）で質問を分類する。

> 実装上のポイント
>
> 対象：社内文書を対象とした RAG
> 課題：質問の意図が多様で、一律の検索では参照情報を絞り込みにくい
> 対応：質問カテゴリ・説明・分類例を JSON で整備し、ICL で質問を分類
> 検証：分類精度、カテゴリ外・複数カテゴリの扱い、検索の再現率、根拠に対する忠実性を分けて確認する

JSON という形式より、カテゴリの粒度、業務用語、境界質問、分類例の品質が重要である。単一カテゴリへ強制せず、複数カテゴリ、分類不能、広域検索へのフォールバックを設ける。RAG の評価は、検索の再現率・適合率と、回答の忠実性・関連性を分けて行う 11。

5.5. Text-to-SQL：SQL の前に指標の意味を読む

自然言語で「先月の売上を商品別に出して」と質問しても、LLM は「売上」の定義を自動で知っているわけではない。どのテーブルを基準にするか、注文と注文明細をどう結合するか、取消・返品をどう扱うか、日付列の意味は何かを理解しなければ、構文上正しい SQL でも業務上誤った数字になる。

Text-to-SQL では、スキーマに加え、指標定義、計算式、許容される結合、粒度、更新頻度を管理する。評価は SQL 文字列の一致ではなく、版管理したデータベース上での実行結果、読取専用権限、クエリ費用上限、破壊的操作の遮断を含めて設計する。SQL の意味的評価には、複数のデータベース状態で実行するテストスイートの考え方がある 13。

5.6. 音声処理：音響条件と語彙を継続的に集める

音声認識では、話者、方言、専門用語、マイク、距離、騒音、同時発話、通信品質が精度を変える。公開ベンチマークで優れたモデルでも、工場の機械音、コールセンターの回線、現場固有の略語にはそのまま適合しない場合がある。

運用データを扱う際は、同意・利用目的・保存期間・アクセス権を定めた上で、誤認識が発生した音声条件と正しい書き起こしを収集する。単語誤り率の平均に加え、固有名詞、設備名、安全に関わる指示、数字、否定表現など、業務上重要な語の誤りを独立して測定する。音声対話では、割り込み、応答遅延、聞き返し、意図の取り違えも評価に含める。

5.7. 実装パターン：異常音分析におけるデータ拡張

異常音が少ない、または録音条件が限られる案件で採用したパターンである。機器の個体差、周囲の騒音、運転状態を考慮して、拡張の対象と範囲を定める。

> 実装上のポイント
>
> 対象：設備・機器の異常音分析
> 課題：収録できる音データの量と条件に限りがあり、判定精度が課題となった
> 対応：データ拡張を適用して学習データの多様さを補った
> 検証：未拡張の実録音を、機器・時期・環境で分けて評価する

拡張は、データ量を機械的に増やす手段ではない。実設備にない音響条件を作ると、不要な特徴へ反応するおそれがある。音声・音響の拡張手法と、そのパラメータ範囲は、ラベルを保つ根拠とともに記録する 14。

5.8. 横断的な失敗分析：モデルの誤りを「一つの精度」に潰さない

画像、言語、音声のいずれでも、誤りを一つの数字だけで扱うと、次の改善が難しくなる。誤りは、少なくとも「入力の取得」「データの意味・正解」「検索・特徴量・前処理」「モデル出力」「人への提示と業務フロー」に分けて記録する。

たとえば、RAG が古い規程を答えた場合、原因は LLM の知識不足ではなく、古い文書を削除していない、現行版のメタデータがない、索引が更新されていない、検索順位が不適切、回答時に有効日を確認していない、といった複数の箇所にありうる。外観検査の誤検知も、モデルの閾値だけでなく、照明の劣化、カメラの汚れ、対象位置のずれ、許容基準の未定義、レビュー画面の見づらさが原因になりうる。

失敗チケットには、発生日時、対象業務、入力条件、期待した結果、実際の結果、影響度、暫定対応、推定原因、恒久対応、評価セットへの反映有無を残す。これにより、単発の問い合わせが、次のモデル更新を検証する回帰テストへ変わる。失敗を隠すのではなく、再現可能なデータとして扱うことが、実装を強くする。

6. 失敗データを資産に変える改善ループ

改善は、モデルを次々と交換することではなく、失敗を観測し、原因ごとにデータと評価へ戻す循環として設計する。簡潔なベースラインを作り、入力欠損、ラベル不一致、条件の偏り、検索漏れ、業務定義の欠落を切り分けてから、データ・ルール・モデルを変更する。

担当者、正本管理者、評価承認者、停止権限を決める。新しいモデルは、重要な失敗、運用コスト、再評価可能性、人による確認・復旧を現場データで比較して採用する。

7. 結論：現場で再評価できる状態をつくる

AI の技術進歩は速い。今日の SoTA が、数か月後に最有力の選択肢であるとは限らない。企業が特定のモデル名だけへ投資すれば、その価値は技術更新とともに薄れやすい。一方で、現場を代表するデータ、合意された正解、失敗を含む評価セット、データの来歴、運用からのフィードバックは、新しいモデルへ移行した後も残る。

7.1. Convergence Lab.の実装前チェックリスト

実装を始める前に、少なくとも次の問いへ答える。

- 解きたい業務判断と、AI に任せない判断は何か。
- 失敗すると最も困る条件は何か。その条件はデータと評価セットに含まれているか。
- 正解・正本・指標の定義に、現場の責任者は合意しているか。
- 本番で起きた失敗、訂正、環境変化を誰がどこへ記録し、いつ再評価するか。

- 新しいモデルを採用する場合、現場データでどの重要な失敗が改善したかを説明できるか。

PoC で動くことと、現場で再評価・改善できることは異なる。必要なのは最初から完璧なデータではなく、データの不足と曖昧さを可視化し、評価可能な小さな範囲から失敗を戻す仕組みである。

7.2. データ・評価診断のご相談

Convergence Lab.では、AI 導入前のデータと評価の整理を支援しています。現場データ、正解定義、評価セット、運用後の改善ループを確認し、PoC で検証すべき範囲とリリース条件を整理します。ご相談は kimura@convergence-lab.com までお問い合わせください。

引用文献

1. Dataset Shift in Machine Learning. - Quiñonero-Candela, Sugiyama, Schwaighofer, Lawrence（編）, MIT Press, 2008、 https://doi.org/10.7551/mitpress/9780262170055.001.0001
2. Model Cards for Model Reporting. - Mitchell et al., ACM FAT*, 2019、 https://doi.org/10.1145/3287560.3287596
3. Artificial Intelligence Risk Management Framework (AI RMF 1.0). - NIST AI 100-1, 2023、 https://doi.org/10.6028/NIST.AI.100-1
4. Hidden Technical Debt in Machine Learning Systems. - Sculley et al., NeurIPS, 2015、 https://proceedings.neurips.cc/paper_files/paper/2015/file/86df7dcfd896fcaf2674f757a2463eba-Paper.pdf
5. Datasheets for Datasets. - Gebru et al., Communications of the ACM, 2021、 https://doi.org/10.1145/3458723
6. Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. - Zhang, Yang, Wang, IEEE Systems Journal, 2019、 https://doi.org/10.1109/JSYST.2019.2905565
7. A systematic literature review of machine learning methods applied to predictive maintenance. - Carvalho et al., Computers & Industrial Engineering, 2019、 https://doi.org/10.1016/j.cie.2019.106024
8. Data Validation for Machine Learning. - Polyzotis et al., SysML, 2019、 https://research.google/pubs/data-validation-for-machine-learning/
9. Guidelines for Human-AI Interaction. - Amershi et al., ACM CHI, 2019、 https://doi.org/10.1145/3290605.3300233
10. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. - Li, Sohn, Yoon, Pfister, CVPR, 2021、 https://doi.org/10.1109/CVPR46437.2021.00954
11. RAGAS: Automated Evaluation of Retrieval Augmented Generation. - Es et al., EACL System Demonstrations, 2024、 https://aclanthology.org/2024.eacl-demo.16/
12. A survey on imbalanced learning: latest research, applications and future directions. - Yang et al., Artificial Intelligence Review, 2024、 https://doi.org/10.1007/s10462-024-10759-6
13. Semantic Evaluation for Text-to-SQL with Distilled Test Suites. - Zhong, Yu, Klein, EMNLP, 2020、 https://aclanthology.org/2020.emnlp-main.29/
14. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. - Park et al., Interspeech, 2019、 https://doi.org/10.21437/Interspeech.2019-2680