Locker.ai: Vision-Language モデルとスマートロッカーを用いた完全無人での遺失物管理サービス

塩畑晴人† 佐々木啓太† 柴沼厳† 杉原大河† 篠原真統†
石井一成† 郡司耕輔† 白田連大† 米倉観† 阪春樹†
弥生宗男† 蓬莱尚幸† 周而晶† 丸山智章†

概要：今日世の中で利用されている遺失物センターでは，主に職員による手作業で遺失物を管理していることがほとんどである．しかし，このアプローチは職員にとって多大な労力を要するだけでなく，主観的な判断による遺失者の認証プロセスは，安全性に潜在的なリスクをもたらすことがある．本研究では，これらの問題について，Vision-Language モデル（GPT-4V）と独自のニューラルネットワークを用いたシステムで代替することで無人化を実現し，より安全で堅牢な遺失物管理サービスを開発した．さらに，このサービスの性能を評価するために，Vision-Language モデルによる特徴文章の生成の正確さを評価した．結果，手動のものとの有意差が認められ劣る例もあったものの，概ね差は小さく，正確さの散らばりは手動よりも抑えられていたことから，十分な安定性があると評価できた．今後は劣る例の改善のための追加研究が求められる．

1. はじめに

1.1 研究の背景

我々は遺失物を発見した場合，あるいは紛失した場合，最寄りの遺失物センターに遺失物を届けたり，探しに行ったりする必要がある．警視庁の調査によれば，遺失物の受理件数は近年で増加傾向にあり，令和 4 年には 3,503,825 点の遺失物が届けられている［1］．うち，遺失物が遺失者に返還された割合は 16.7%となっており，それ以外は拾得者に引き渡されるか，都に帰属するか，廃棄されている．このことから，多くの遺失物が遺失者のもとに還ってきていないことがうかがえる．こうした遺失物を管理する遺失物センターでは，職員が手作業で遺失物を管理したり，遺失者の申告の信憑性を主観的に判断したりすることが多い．このようなアナログ的なシステムはおよそ一般的ではあるものの，いくつかの課題がある：(1) 遺失物がどのセンターで保管されているのかを事前に確認できない；(2) 保管する遺失物が増えるごとに職員の負担が増加する；(3) 職員の主観が遺失者の認証プロセスに影響を与え，悪意のある人物による成りすましに対処できない場合がある［2］．このような問題に対して包括的に対処するために，この領域における Digital Transformation（DX）の実施が望まれる．

1.2 関連研究

このような問題に対して，先行してソリューションを提供しているサービスがある［3-5］．それらのサービスでは，遺失物の登録・検索の処理を自動化することで職員の負担を軽減している．しかし，遺失物の受け渡しや遺失者の認証の際には人の手が必要なこと，また，遺失物を格納するロッカーの自動化がされていないことから，全てのプロセスにおいて完全な DX が実施されているとはいえない．

1.3 研究の目的

本研究では，前述の問題点を解決するための遺失物管理サービス「Locker.ai」を提案する．このサービスは，遺失物管理の完全な無人化を実現し，より安全性と信頼性の高い認証体験を提供することで，正しく効率的に遺失物を遺失者に返還することを目的とする．具体的には，遺失物とそれが保管されているロッカーとの関係をデータ化して保持することで（1）の問題を解決し，またロッカーの施錠・解錠機構を自動化した上で認証プロセスに統合することで(2)の問題を解決する．さらに，先の認証プロセスを Vision-Language モデル（gpt-4-vision-preview）と独自に開発したニューラルネットワークで代替することで（3）の問題を解決する．

2. システム概要

2.1 システム構成

Locker.ai は，主にユーザーが遺失物を登録・検索するためのインターフェースであるウェブサイト，遺失物を保管するためのロッカー，およびユースケースを適切に処理しデータを保持するためのサーバーで構成されている（図 1 参照）．ウェブサイトは主に Next.js を用いて実装され，サーバーは主に NestJS を用いて実装されている．さらに，認証，ストレージ，データベースの管理には Supabase を用いている．ロッカーの施錠・解錠機構の制御には Raspberry Pi と Mbed を用い，ロッカー上でユーザーを生体認証するために指紋認証モジュールとの統合を行っている．なお，ウェブサイトとサーバー間の通信には型安全性と実行効率性のために GraphQL を用い，Raspberry Pi とサーバー間の通

† 茨城工業高等専門学校 (National Institute of Technology, Ibaraki College)

信には実装容易性のために REST API を用いている．

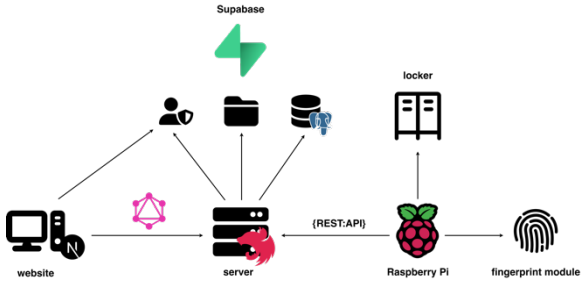

図 1 システム構成

2.2 動作フロー

Locker.ai の動作フローを図 2 に示す．遺失物を発見した拾得者は，スマートフォン等を用いて遺失物を撮影し，その画像をウェブサイトにアップロードする．サーバーは画像から遺失物の特徴を自然言語文として生成し，日付や画像 URL などの情報とともにデータベースに保存する．その後，拾得者は遺失物を持ってロッカーに移動し，指紋によって認証したのちに，指定の引き出しへ遺失物を格納する．紛失に気がついた遺失者は，まずウェブサイトに遺失物の特徴を自然言語文で入力する．サーバーは指定された特徴に類似する遺失物を検索し，特徴が一致した遺失物について，遺失者に対して自身のものであるかを質問する．提示された遺失物が自分のものであった場合，遺失者は質問に肯定し，指定のロッカーへ移動する．その後指紋によって認証したのちに，指定の引き出しから遺失物を回収する．

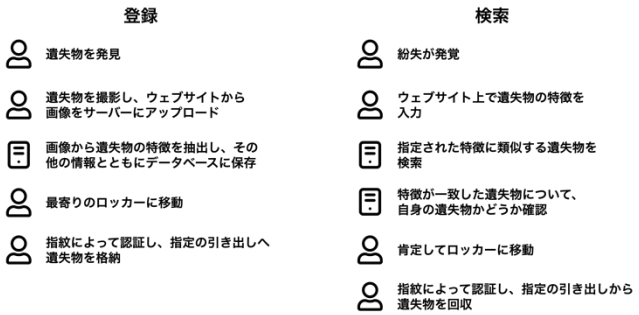

図 2 動作フロー

3. システム実装

3.1 遺失物の登録

Locker.ai のコア機能の一つに，サーバーによる遺失物の登録処理があげられる．この処理は次のように大別できる：（1）画像から遺失物の特徴を文章として生成する；（2）文章をベクトル化する；（3）ベクトルをデータベースに保存する．このとき，（1）と（2）の処理ではそれぞれ機械学習タスクを実行する必要があるため，それに応じたモデルを選定する必要がある．

3.1.1 画像から遺失物の特徴文章を生成するモデルの選定

モデルの選定について議論する前に，まず，この処理がどのようなタスクであるのかを明確にする．この処理は，単純に解釈すれば画像からのラベル検出と考えることができる．しかし，類似する遺失物を同時に管理する可能性があることから，画像中のオブジェクトを単語として検出するようなラベル検出では，例えばメーカーの違う複数の腕時計を同時に管理していた場合に，これらを一意に特定することができない．よって，今回のような処理では，ラベル検出ではなく，image to text（image captioning）の方がより適切であると考える．

次に，image to text を行うモデルの選定について述べる．今回は次の 3 つの手法を候補とした：（i）Salesforce/blip-image-captioning-large による image to text を行う手法；（ii）facebook/detr-resnet-101-dc5 によって画像の下処理を行った後で Salesforce/blip-image-captioning-large による image to text を行う手法；（iii）gpt-4-vision-preview による image to text を行う手法．これらの手法の性能を比較するために，それぞれ次の手順で評価した；（i）Salesforce/blip-image-captioning-large［6］は Hugging Face 上で提供されている BLIP［7］をベースとしたモデルであり，同サービスの Inference Endpoints に従って機能を提供している．このため，このモデルは REST API によるリクエストを行うことで使用することができる．（ii）単に image to text を行うモデルを用いた場合の懸念として，遺失物以外の特徴が出力に含まれてしまうことが挙げられる．これを防ぐために，画像の下処理として，facebook/detr-resnet-101-dc5［8, 9］の物体検出によって遺失物の領域の切り取りを行う．なお，画像内において複数の物体が検出された場合は，どの物体が遺失物であるかを特定する必要がある．これを行うために，さらに別のモデルを用いることもできるが，今回は予備実験であることから，簡略化のために画像の中心に最も近い物体が遺失物であるという仮定の下で対応することとした．（iii）gpt-4-vision-preview［10］は OpenAI によって提供されている大規模視覚言語モデルである．これは画像認識タスクを処理する機構が統合されている点で，従来の LLM とは大きく異なる．このモデルは，OpenAI の API として提供されているため，命令文章と画像 URL をプロンプトとしてリクエストすることで使用できる．今回は簡易的なプロンプトとして，“Describe the most prominent object in the provided image URL as a lost item. Ignore surroundings.” を命令文章とし，入力画像を Base64 エンコードしてデータ URL としたものを画像 URL とした．

以上の手順により，それぞれのモデルを評価する．図 3 の（a）に示す画像を入力したところ，それぞれ次のような結果が得られた：（i）“someone is holding a watch in their hand on a table” という出力が得られた；（ii）下処理段階では図

3 の（b）および（c）に示すような結果が得られ，最終段階では“a close up of a person holding a watch on a table”という出力が得られた；（iii）“A white digital-analog wristwatch with the brand 'Casio G-SHOCK' visible on the dial.”という出力が得られた．

以上の結果を考察する．まず，（i）の手法では，得られた出力は画像全体の説明であり，遺失物のみにフォーカスできていないことがわかる．また，（ii）の手法においても，下処理後の画像に写り込んでしまった手や机が認識され，出力に含まれてしまっている．一方，（iii）の手法では十分に周囲の情報を除外できており，遺失物の特徴のみを言語化できているといえる．これは，gpt-4-vision-preview 自身が遺失物の特徴文章を生成するというコンテキストを理解できているためであると考えられる．また，命令文章を工夫すれば遺失物の損傷具合など，その遺失物に固有の情報を抽出することも可能である．これらのことから，Locker.ai では，gpt-4-vision-preview を用いて，画像からの遺失物の特徴文章の生成を行うこととした．

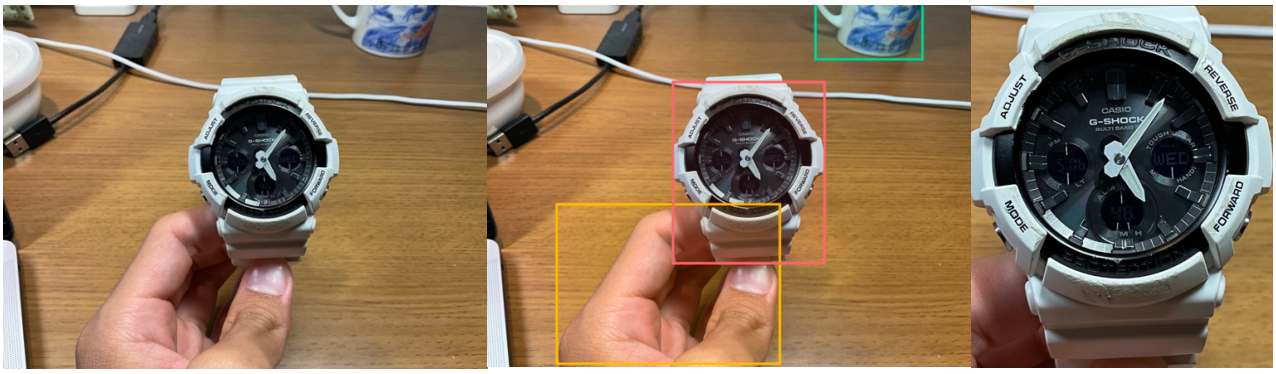

図 3 遺失物の写真とその加工の例（左から（a），（b），（c）とする）

3.1.2 テキストをベクトル化するモデルの選定

図 2 で示したように，遺失者を認証するためには，Vision-Language モデルが生成した遺失物の特徴文章と，遺失者が入力した遺失物の特徴文章とを比較し，これらの一致性を検証する必要がある．この処理ではコサイン類似度をベースとした検証を行っているため，両文章をベクトル化する必要がある．今回は，ベクトル化する言語モデルとして次の 3 つを候補とした：（i）BERT；（ii）sentence-transformers/all-MiniLM-L12-v2；（iii）OpenAI Embedding API. これらのモデルを候補として挙げた理由を次に示す：（i）BERT は高速なアーキテクチャである Transformer をベースとした言語モデルである［11］．このため，高速なベクトル化ができることに期待し，候補とした；（ii）sentence-transformers/all-MiniLM-L12-v2 が表現できるベクトルの次元数は比較的少ないものの，コサイン類似度を用いて交差エントロピー誤差を計算していることから，高い精度を期待できるため候補とした［12］；（iii）OpenAI Embedding API によるベクトル化は，文章を 1356 次元のベクトルに変換する［13］．これは今までに挙げたどのモデルよりも高次元なベクトルで文章を表現できることを意味する．このため，高い変換精度を期待して候補とした．

これらのモデルを比較するために，今回はそれぞれに対していくつかのケースを実施し，各ケースではベースとなる文章と 3 つの比較文章との類似度を算出した．

以上の手法による評価で得られたデータの一部を，図 4 に示す．この結果から，BERT と OpenAI Embedding API は，ベース文章の名詞に形容詞を加えた比較文章と全く異なる文章との類似度に，顕著な差は見られない一方で，sentence-transformers/all-MiniLM-L12-v2 は，およそ文脈が一致する文章と見当違いな文章との類似度の差が大きいことがわかる．このことは，検証のための閾値をより精度高く設定できることを意味する．以上のことから，テキストをベクトル化するモデルには，sentence-transformers/all-MiniLM-L12-v2 を使用することとした．

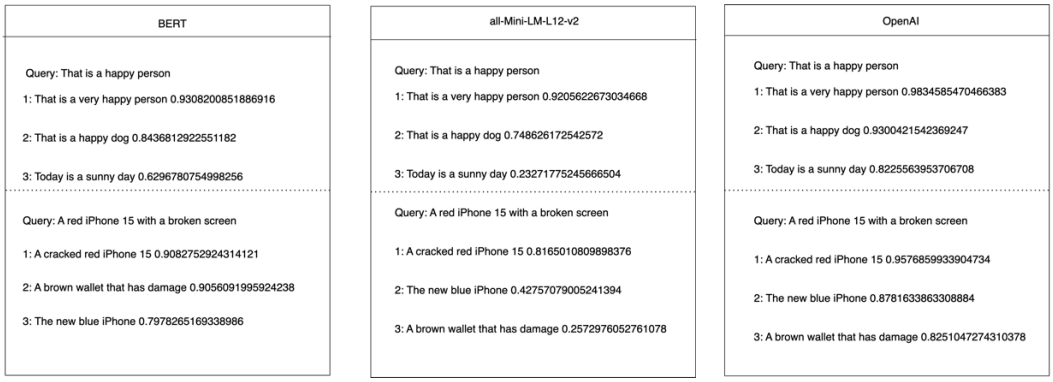

図 4 言語モデルの評価結果の一部

3.2 遺失物の検索

3.1 節で挙げた遺失物の登録処理に加え，遺失物の検索処理もまた，Locker.ai にとって重要な機能の一つである．この処理は，次のように大別できる：(1) 遺失者が検索した遺失物の特徴文章をベクトル化する；(2) ベクトル化した遺失者による特徴文章をもとに，データベース (PostgreSQL) に保存されている，3.1.2 節で示した手法によって自動生成された特徴文章のベクトルに対してベクトル検索を行い，top-K の遺失物データを取得する；(3) 取得したデータに対応するコサイン類似度および，拾得者が遺失物を報告した日時と，遺失者が思い当る紛失した日時との差を入力とし，一致および不一致の確率を出力とするニューラルネットワークを用いて，top-K の遺失物データから最も一致確率の高い遺失物を算出する．ここで，コサイン類似度のみで文章の一致性を検証した場合，遺失者が説明した文章で述べられている遺失物の特徴が，自動生成された文章で述べられている特徴よりも少なかったために，コサイン類似度が低く算出されてしまい，意図せず不一致と判定されてしまう場合がある．ただし，単に自動生成された文章に対して抽象的すぎる文章を一致するように実装した場合，成りすましに対処できない懸念がある．これらの問題に対応するために，今回は先に述べた日時の差を考慮することとした．これは，紛失した日時が遺失者本人にしか知り得ない情報であるためである．これにより，検索プロセスにおける，意図しない不一致を防ぎつつ，悪意のある人物による成りすましを防ぐ．

以上で説明したニューラルネットワークの実装について述べる．実装には主に TensorFlow.keras を用いた．訓練データに使用するために作成したデータセットの分布を図 5 に示す．なお，ミリ秒で表される日時の差は，正規化を行うことで範囲を縮小しつつ，外れ値については閾値を設けることでフィルタリングをしている．

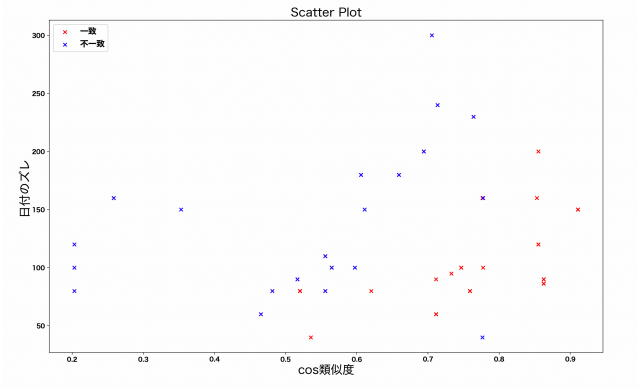

図 5 訓練データの分布

3.3 ユーザー認証

Locker.ai の認証処理は，ウェブサイトとロッカーの二つの場面で行われる．

3.3.1 ウェブサイトでの認証

Locker.ai のウェブサイトは，Supabase に統合された Google 認証システムによって認証処理を行っている．ユーザーは Locker.ai のコア機能を初めて利用する際，任意の Google アカウントにログインするだけで Locker.ai にサインアップすることができる．

3.3.2 ロッカーでの認証

遺失物の保管を担うロッカーでは，遺失物を拾得または回収したユーザーを認証するために，R307 Fingerprint Module [14] を用いた指紋認証機構を設けている．このモジュールに対してユーザーがアクセスすると，モジュールはユーザーの指紋を一意の ID として Raspberry Pi に提供する．その後，Raspberry Pi は，この指紋 ID を持つユーザーについてサーバーに問い合わせることで認証を行う．なお，サービス全体として，ユーザーが初めて認証を行うのは 3.3.1 節で示したようにウェブサイト上である．つまり，初期状態ではユーザーのアカウントには指紋が紐付けられていない．このため，サーバーが指紋 ID を用いてユーザーを特定できなかった場合，Raspberry Pi は指紋紐付け処理に移行する．ここでは，主に Supabase の機能の一つで

あるマルチファクター認証に従って処理を行うことで，より安全に指紋の紐付けを行っている．

3.4 ロッカーの設計

ロッカーの詳細な設計について，図 6 に示す．本研究では，管理する遺失物として，スマートフォン，長財布といった比較的小型なものを対象とした．これらを無理なく格納できるように，一つの引き出しの内寸は各辺およそ 200mm としている．また，扉の施錠・解錠機構にはソレノイド式電気錠スリムロック［15］を使用している．これは，扉を閉めると施錠され，ソレノイドに通電すると解錠される．また，厚さが 10mm であることから，引き出し内部を最大限に使用することができる．

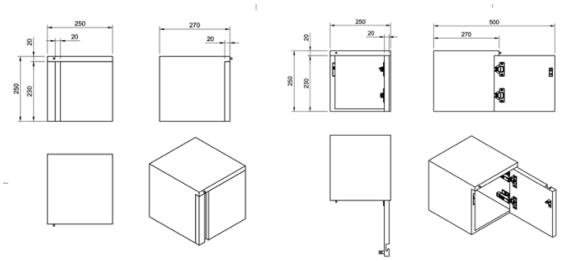

図 6 ロッカーの詳細な設計図

3.5 ロッカーの制御システム

ロッカーの制御部分は，主に Raspberry Pi と Mbed によって構成される．ここで，Raspberry Pi での実装では ROS（Robot Operating System）を用いることで，バックエンドサーバーとの通信と，ソレノイドロックおよび LED の制御を，副作用的に行なっている．さらに，制御回路までの経路で Mbed を経由することで，制御する引き出しの増加に伴う GPIO ピンが不足する問題に対処している．

4. 性能評価

大規模視覚言語モデル（LVLM）を用いて生成された遺失物の特徴文章の正確さを示すために，次の手法を用いて性能評価を行った．まず，警視庁による主な遺失物の分類［1］を参考に，典型的な遺失物 6 つとその複数枚画像をサンプルとして収集した．そして，その遺失物の特徴を網羅する正解テキストをそれぞれ 5 つずつ設定した．また，ほぼ共通のプロンプトと遺失物の画像を与えて LVLM と人間に説明文（以下出力テキスト）を書かせた．

4.1 UniEval を用いた定性的評価

UniEval［16］が提供する，事実の一貫性の指標 Factual Consistency Score を算出するように事前訓練されたモデル MingZhong/unieval-fact［17］を用いて，正解テキストと LVLM が生成した出力テキストの事実の一貫性の指標と，正解テキストと人間が書いた出力テキストの事実の一貫性の指標とに，有意差が認められるかどうかを調査した．具体的には，正解テキストと LVLM が生成した出力テキストの全ての組み合わせおよび，正解テキストと人間が書いた出力テキストの全ての組み合わせを用いて指標を求めた（図 7 参照）．また，有意差の有無を調べるために，ウェルチの t 検定を行った．なお，有意水準は 95%とする．

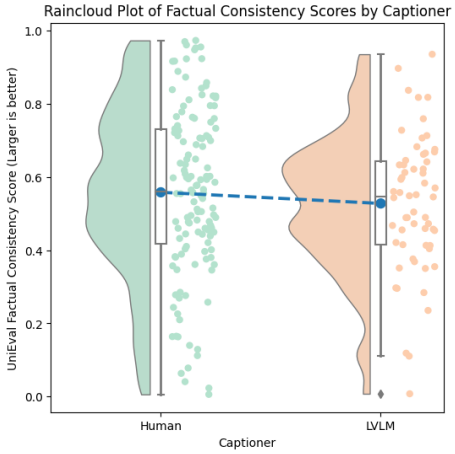

図 7 (a) Factual Consistency Score の人間と LVLM の分布

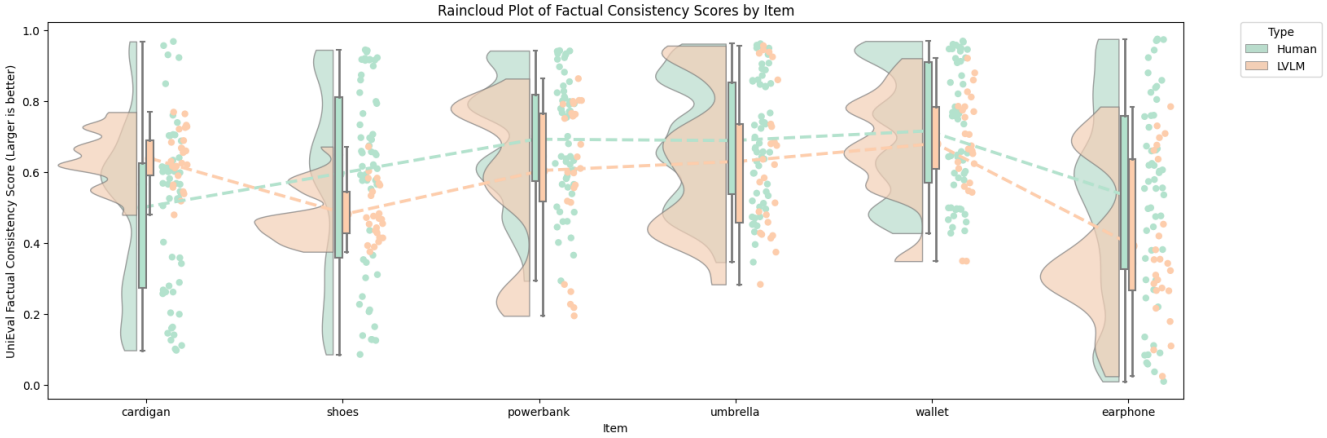

図 7 (b) Factual Consistency Score の遺失物サンプル別の分布

4.2 評価の結果

評価の結果を表 1 に示す．衣類では，人間を上回る正確さで特徴文章を生成した．また，傘類，財布類，モバイル充電器では，人間と同等の正確さで特徴文章を生成した．しかし，無線イヤホンおよび履物類では，人間を下回っていることがわかる．全体的には，正確さに有意差が出たものの，平均値の差は十分に小さく，分散は LVLM の方が小さかった．

表 1 ウェルチの t 検定による
評価結果（有意水準：95%）

遺失物の名前	T 統計量	P 値	有意差
衣類	-3.86434	0.00023	有
履物類	2.90867	0.00484	有
モバイル充電器	1.81819	0.07672	無
傘類	1.23414	0.22426	無
財布類	0.93585	0.35328	無
無線イヤホン	2.43843	0.01775	有
すべて	2.43843	0.01775	有

4.3 評価の結果の考察

正確さが人間を下回ってしまった無線イヤホンおよび履物類の正解テキストには，有名なため人間は認識できるものの，画像中にテキストとして写っていない固有名詞（AirPods Pro や Converse Runstar Hike など）が含まれていた．つまり，出力テキストに固有名詞が含まれることが想定される遺失物では，LVLM の正確さが低くなる傾向があったと言える．さらなる性能向上には，製品名の推定能力の改善が必要となる．

5. おわりに

本研究では，正しく効率的に遺失物を遺失者に返還することを目的に，遺失物管理の完全な無人化を実現する Locker.ai を提案した．性能評価より，Locker.ai は多くのケースにおいて手動の場合と同等か，それ以上の遺失物の説明能力を持っていることが言えた．またこのことから，多くの典型的な遺失物について，完全無人での管理が可能であることが言える．しかしながら，一部の遺失物では手動の場合よりも低い性能が示されたことから，システムのさらなる研究が求められる．

参考文献

[1] “警視庁：遺失物取扱状況(令和 4 年中)”. https://www.keishicho.metro.tokyo.lg.jp/about_mpd/jokyo_tokei/kakushu/kaikei.html, (参照 2023-12-13).
[2] “落とし物所有者のふりして詐欺罪で逮捕 埼玉県戸田市の刑事事件弁護士 | 埼玉で刑事事件・少年事件でお悩みなら無料法律相談対応の「あいち刑事事件総合法律事務所-さいたま支部」へ”. https://saitama-keijibengosi.com/otoshimono-shoyuusha-sagi-taiho-saitama-toda-keiji-jiken-bengoshi, (参照 2023-12-13).
[3] “遺失物管理システム、及びプログラム”, 特願 2023-013515, 株式会社ティファナドットコム, 佐野弘, 石井明夫, 藤井亮, 願 https://www.j-platpat.inpit.go.jp/c1800/PU/JP-7371843/97B841E5BF20BB12F325065FA206727022C510B5477D0AB816EB6FDFE0A22033/15/ja, (参照 2023-12-13).
[4] “find｜落し物クラウド”. https://service.finds.co.jp/, (参照 2023-12-15).
[5] “NEC 遺失物管理ソリューション |NEC ソリューションイノベータ”. https://www.nec-solutioninnovators.co.jp/sl/lams/index.html, (参照 2023-12-13).
[6] “Salesforce/blip-image-captioning-large · Hugging Face”. https://huggingface.co/Salesforce/blip-image-captioning-large, (参照 2023-12-21).
[7] Junnan Li, Dongxu Li, Caiming Xiong, Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. International Conference on Machine Learning. 2022, vol. 162, p 12888-12900.
[8] “facebook/detr-resnet-101-dc5 · Hugging Face”. https://huggingface.co/facebook/detr-resnet-101-dc5, (参照 2023-12-21).
[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko. End-to-End Object Detection with Transformers. Computer Vision – ECCV 2020. 2020, vol. 2020, p. 213-229.
[10] “GPT-4V(ision) system card”. https://openai.com/research/gpt-4v-system-card, (参照 2023-12-15).
[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin. Attention Is All You Need. Advances in Neural Information Processing Systems 30. 2017.
[12] “sentence-transformers/all-MiniLM-L12-v2 · Hugging Face”. https://huggingface.co/sentence-transformers/all-MiniLM-L12-v2, (参照 2023-12-21).
[13] “Embeddings - OpenAI API”. https://platform.openai.com/docs/guides/embeddings, (参照 2023-12-21).
[14] “R307 Fingerprint Module”. https://www.rajguruelectronics.com/Product/1276/R307%20Fingerprint%20Module.pdf, (参照 2023-12-15).
[15] “ソレノイド薄型電気鍵スリムロック - タカハ機工”. https://www.takaha.co.jp/co/slimlock/, (参照 2023-12-21).
[16] Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, Jiawei Han: Towards a Unified Multi-Dimensional Evaluator for Text Generation. https://arxiv.org/abs/2210.07197, (参照 2022-10-13).
[17] “MingZhong/unieval-fact · Hugging Face”. https://huggingface.co/MingZhong/unieval-fact, (参照 2023-12-22).