公理系を用いたHuman-in-the-loopエンティティマッチング

小泉 崇裕† 伊藤 寛祥†† 吉本 龍司††† 福島 幸宏†††† 原田 隆史††††† 森嶋 厚行††

† 筑波大学 情報学群 情報メディア創成学類 〒305–0821 茨城県つくば市春日 1–2
†† 筑波大学 図書館情報メディア系 〒305–0821 茨城県つくば市春日 1–2
††† 株式会社カーリル 〒509–9232 岐阜県中津川市坂下 1645–15
†††† 慶應義塾大学 文学部 〒108–8345 東京都港区三田 2–15–45
††††† 同志社大学 免許資格課程センター 〒602–8580 京都府京都市上京区今出川通新町上ル
E-mail: †koizumi.takahiro.qg@alumni.tsukuba.ac.jp, ††{ito,mori}@slis.tsukuba.ac.jp, †††ryuuji@calil.jp,
††††fukusima-y@keio.jp, †††††ushi@slis.doshisha.ac.jp

あらまし エンティティマッチングは，データベースの中から同一実体を参照するレコードの統合を行う問題として広く知られる．この課題へのアプローチに，機械学習とクラウドソーシングを掛け合わせた Human-in-the-loop によるレコードペアの同定を行う手法がある．既存手法では，機械学習における判断の確信度が低い場合に人間の判断を仰ぐ手法が一般的であった．本研究では，機械学習における判断とエンティティマッチングにおける公理の矛盾を手がかりに人間に判断を求める手法を提案する．また，そのデータを教師データとして再学習させることで，同定精度をより高める事を試みる．全国 963 の公立図書館の書誌データの一部を用いた実験により，提案手法の効果を確認した．

キーワード エンティティマッチング，公理系，Human-in-the-loop，データ統合技術

1 はじめに

複数のデータベースを統合する際に，各データベースに格納されるレコード間で互いに識別可能な固有番号がないことがしばしばある．この状態で単純に結合してしまうと，重複したレコードが格納され，データの実情把握や情報検索の観点で不利になることが予想される．この問題は，同じ実体を参照するレコードを，項目内容の比較により発見し統合する「エンティティマッチング問題」として広く研究されている．

エンティティマッチングには，データ規模が大きくなると計算機の誤判定とクラウドソーシングコストが急増するという問題がある．一般にエンティティマッチングは組み合わせの問題であるから，n 個の同定対象データが k 倍になると比較対象となるペアは ${}_{kn}\mathrm{C}_2$ となり k^2 オーダーで増加する．

エンティティマッチングの問題は長年の研究にも関わらず，依然として困難な問題として指摘 [1] されており，当初は計算機のみで解決を図る手法 [2] [3] が研究されてきたが，その困難さから，Human-in-the-loop を用いた手法の研究が近年盛んに行われている．Human-in-the-loop とは，各々得意とするタスクを計算機と人間に割り当てて相互補完しながら動作するシステムのことを指し，機械学習とクラウドソーシングの組み合わせで注目を集めている．例えば，固有番号のないペアのみクラウドソーシングを行う手法 [4] が提案されており，また，機械学習での判定をベースに，確信度が低いデータペアを人間が修正することで，コストを抑えつつ高精度なエンティティマッチングを実現する手法 [5] [6] が提案されている．

本研究では，既存研究と異なり，機械学習における判断とエンティティマッチングにおける公理系との矛盾を手がかりに人間の判断を求める手法を提案する．また，そのデータを教師データとして再学習させることで，学習精度をより高める事を試みる．公理系に矛盾したグループは，推論器が誤判定したペアが確実に含まれていることから，再学習に対し価値の高いデータと考えられる．このグループをクラウドソーシングで問い合わせることで，信頼性の高いデータを獲得することができ，このデータを教師データとして推論器にフィードバックすることで，既存モデルとの差異が減少し，より精度の高い推論を行えることが期待できる．

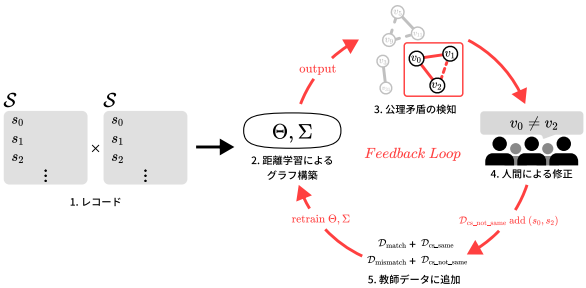


図 1 **提案フレームワークの概要**: 公理系によって AI の推論の矛盾を発見し，人間が修正したものを再度 AI の学習に用いる Human-in-the-Loop のフレームワークを提案する

図 1 に，本研究の新規性にあたるフィードバックループを示す．従来手法で実装された「学習フェーズ」と「同定フェーズ」に加えて，公理による推論機の誤判定を自己検出する「公理矛盾検出フェーズ」と，そのデータを推論器にフィードバックする「再学習フェーズ」を追加した図 2 の手法を提案する．

本研究は，データの規模を大きくした場合に再学習が与える

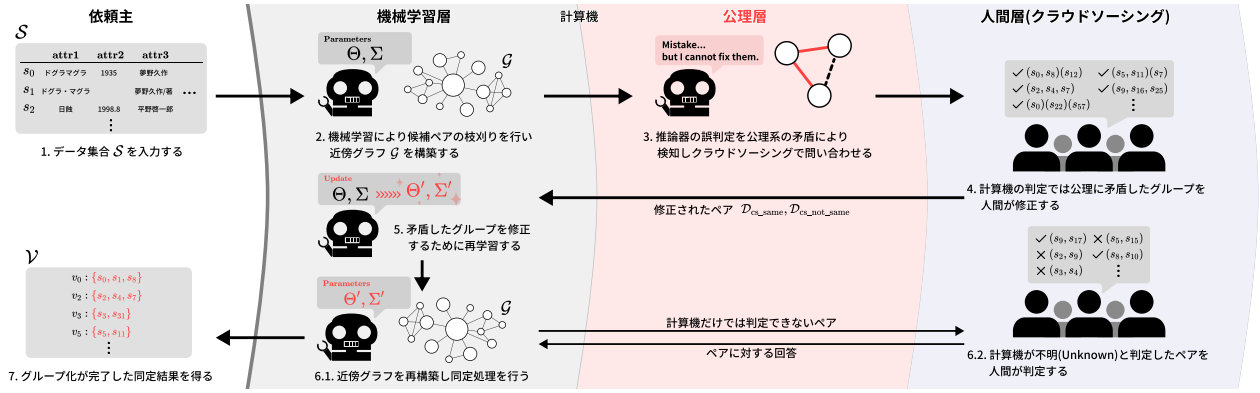


図 2　提案手法概略

推論器精度の影響の検証を目的とした実験を行った．

本研究の貢献

- **公理系矛盾検出と再学習による同定手法の提案**: エンティティマッチングのドメインの公理系との矛盾による学習データの獲得を試みる手法を提案している．
- **実データによる実験**: 実際の公立図書館の書誌データ統合における実験結果により，提案手法が有望であることを示しており，クラウドソーシングコストと精度の関係についても調査した結果を報告する．

この成果は，比較的大きなデータ群に対し高精度かつ低コストにエンティティマッチングを行えることを示唆したものである．

2　関 連 研 究

本章では，本研究に関連の深い研究を紹介し，その手法の問題点や提案手法との比較について述べる．

2.1　計算機のみのエンティティマッチング

エンティティマッチングの基礎として，属性値に着目したブロッキング手法 [2] が提案された．ある 1 つの属性と一致しない候補を大幅に削除する手法であるが，入力ミスや欠損，表記揺れ等により，対象の属性を決めるのが難しい．

上田ら [3] は SVM(support-vector machine) を使った固有識別情報を持たない書誌同定手法を提案しており，計算機による同定および検索において精度向上を図った．この手法はデータにより向き不向きがあり，15% から 81% と F 値に揺れが見られる．本研究は，旧 Facebook 社が提供する Fasttext[1]から成る特徴ベクトルを利用し，SVM ではなく距離学習を利用する．

2.2　クラウドソーシングを利用した書誌同定

クラウドソーシングは，不特定多数の人にタスクを割り振るシステムのことを指す．原田ら [4] は，誤入力やデータの欠損などにより計算機だけでの自動同定処理は限界があると提起し，計算機が判定困難な書誌データペアを，人の介入が可能なクラウドソーシングを活用した手法を提案した．計算機が誤判定すると考えられる書誌ペアの特定を試み，クラウドソーシングによる該当ペアの修正と計算機による自動化手法の改善を行った．また，類似度計算には Okapi BM25 [7] を利用した．実験の結果，8 人以上の人間の介入が行われると信頼できる結果を得られたとしている．この研究はクラウドソーシングの効果について主眼を置いているため，コストの考慮や，計算機が同定したペアに対する検証は行われていない．

2.3　Human-in-the-loop によるエンティティマッチング

Li [5] は，エンティティマッチングの重要な問題として,「2 つのレコードが同じ実体かを評価することは容易ではない」こと,「レコードペアを全て確認することはコストが大きい」ことを挙げている．このことを踏まえ，まず大量の非一致ペアを計算機が刈り込み，誤同定を防ぐためにクラウドソーシングを利用する．さらに，コスト減少のために公理系による推論を取り入れており，一致と考えられるペアから問い合わせることで，3 つのタスクを 2 つに減少できる可能性があることに言及している．

大沢ら [6] は，距離学習による候補ペアの枝刈りとベイズ推論，クラウドソーシングによる高精度な同定フレームワークを提案している．できる限り計算機が判定する手法で，コストを最小化する工夫も見られる．一方で，複数の評価指標を束ねる推論方法は，評価指標が拮抗した場合に正しく判定できない問題がある．本研究では，指標が拮抗した際に値が大きくなる「Unknown(不明) 領域」を考慮した SUN 推論を提案する．

2.4　エンティティアラインメント半教師あり学習モデル

Zhu ら [8] は，異なる知識グラフから同じ意味を示す実体を探す為に，半教師あり学習モデルを提案する．GCN(Graph Convolutional Networks) を利用し，エンティティとリレーションのグラフ構築に対する再学習を行うことで精度向上を図った．本研究とは学習方法が異なるが，エンティティアラインメント分野では半教師あり学習モデルの効果が示された．

3　準　　備

本章では記号定義を行い，さらに提案手法の軸となる「距離学習器」と「SUN 推論器」,「近傍グラフ」について説明する．

1 : https://fasttext.cc/

3.1 記号定義

本論文において使用する記号を表 1 に示す．一部の記号は $\mathcal{D}_{\text{match}}$ のように添字が付されている場合があるが，添字付き記号については適宜説明する．

表 1　記号の定義

$\mathcal{S} = \{s_i\}$	同定対象のターゲットデータベース
s_i	各データ (レコード)
$\mathcal{D} = \{(s_i, s_j)\}$	データペア
$\mathcal{T} = \{(s_i, s_j, s_k)\}$	データトリオ
$M_\Theta : s_i \to \mathbb{R}^n$	距離学習による埋め込み
$m \in \mathbb{R}$	距離学習時の不一致ペアのためのマージン
$\Sigma \in \mathbb{R}^n$	SUN 推論器を構築するパラメータ集合
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	ノードとエッジで構成される近傍グラフ
$\mathcal{V} = \{v_i\}$	近傍グラフにおけるノード集合
$v_i = (s_i, group : \{s_i\})$	各ノード (s_i: データ，$group$: 同定済データ集合)
$\mathcal{E} = \{(v_i, v_j)\}$	近傍グラフにおけるエッジ集合
τ	閾値

データとノードのインデックスは共有するものとし，$s_i \equiv v_j (i = j)$, $s_i \neq v_j (i \neq j)$ が成り立つ．

3.2 距離学習器

距離学習器は，主に比較対象のペアの枝刈りを目的としており，後述する SUN 推論器の指標の 1 つとしても利用する．

3.2.1 学　　習

まずデータ s_i が持つ各属性値について，旧 Facebook 社が提供する Fasttext によって，各々を 300 次元のベクトルに変換する．日付など数値しか持たない属性値は，One-hot ベクトルを利用し，300 次元になるよう不足分は 0 埋めした．これらのベクトルを連結し距離学習を行う．ここで，一致ペアを含む教師データを $\mathcal{D}_{\text{match}}$，不一致ペアを含む教師データを $\mathcal{D}_{\text{mismatch}}$ とする．

距離学習は *Siamese Network* [9] を利用し，Θ を最小化する問題として，式 (1) より定義される．

$$\Theta = \arg\min_{\Theta} \sum_{(s_i,s_j)\in\mathcal{D}_{\text{match}}} Dist_1(M_\Theta(s_i), M_\Theta(s_j)) - \sum_{(s_i,s_j)\in\mathcal{D}_{\text{mismatch}}} Dist_2(M_\Theta(s_i), M_\Theta(s_j)) \quad (1)$$

損失関数には *Contrasive Loss* [10] を利用する．

$$Loss = \frac{1}{2}\left(\sum_{(s_i,s_j)\in\mathcal{D}_{\text{match}}} Dist_{euc}(s_i, s_j)^2 + \sum_{(s_i,s_j)\in\mathcal{D}_{\text{mismatch}}} \max(m - Dist_{euc}(s_i, s_j), 0)^2 \right) \quad (2)$$

なお $Dist_{euc}(s_i, s_j)$ は式 (3) として定義される．

$$Dist_{euc}(s_i, s_j) = \|M_\Theta(s_i) - M_\Theta(s_j)\|_2 \quad (3)$$

距離学習器の実装にあたっては Keras[2] を利用し，3 つの全結合層と 3 つのドロップアウト層により構成される．

3.2.2 推　　論

推論時に得られる 2 つのデータ間の距離は，式 (3) からユークリッド距離で算出できる．

3.3 SUN 推論器

SUN 推論器は，各類似度の確率密度関数の値を束ねて，そのデータペアがどの程度一致しているかを推論するものである．

3.3.1 学　　習

SUN 推論器の構築にあたっては，Bubble [6] の手法を参考に，事前に用意された一致・不一致ペアの教師データ $\mathcal{D}_{\text{match}}, \mathcal{D}_{\text{mismatch}}$ を利用し，各類似度指標から出力される値を基に確率密度関数を構成するパラメータを取得する．本研究では，表 2 の類似度指標を使用する．

表 2　類似度指標

指標	分布	
Fasttext vector	γ	距離学習によるデータ間の距離
Jaro-winkler [11]	β	文字列間の置換の必要度合い
Levenshtein [12]	γ	置換や削除などの編集距離
Gestalt Pattern Matching[3]	β	文字列長と一致文字数から算出

教師データを入力とし，これらの各類似度指標から得られる値を一致ペアと不一致ペアに分け，β 分布または γ 分布を基とする確率密度関数へのフィッティングを図 3 のように行う．確率密度関数へのフィッティングは scipy[4] を利用した．

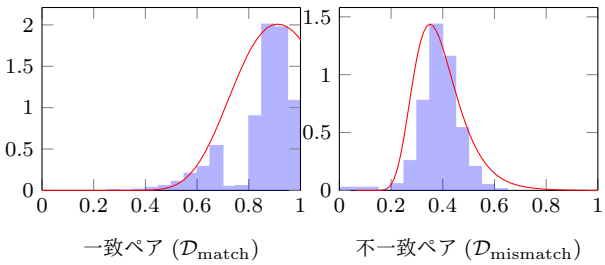


図 3　確率密度関数へのフィッティング

これら各々の確率密度関数を表現するためのパラメータ全てを総称して Σ とする．

3.3.2 推　　論

推論時は，複数の類似度指標に基づく確率密度関数の値を束ねることにより，*Same* 領域，*Unknown* 領域，*Not − same* 領域の各々の値を算出する．例えば類似度 A から得られる値を x_A とする時，一致ペアの確率密度関数の値と不一致ペアの確率密度関数の値を，それぞれ $a(x_A)$，$\bar{a}(x_A)$ とする (図 4)．

x_A の値を取った時，類似度 a においてそのペアが一致する

2 : https://keras.io/
3 : https://docs.python.org/3/library/difflib.html
4 : https://scipy.org

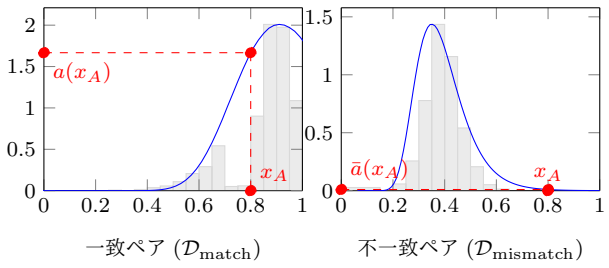


図 4　類似度指標 A において $x_A = 0.8$ の確率密度関数の値

確率 $P_a(x_A)$ を式 (4) のように定義する．

$$P_a(x_A) = \frac{a(x_A)}{a(x_A) + \bar{a}(x_A)} \tag{4}$$

不一致確率も同様に定義できる．

$$P_{\bar{a}}(x_A) = \frac{\bar{a}(x_A)}{a(x_A) + \bar{a}(x_A)} \tag{5}$$

なお，$P_a(x_A) + P_{\bar{a}}(x_A) = 1$ であることに留意する．

本研究では類似度指標を 4 つ使うため，各指標の一致確率については $P_a(x_A), P_b(x_B), P_c(x_C), P_d(x_D)$，不一致確率は $P_{\bar{a}}(x_A), P_{\bar{b}}(x_B), P_{\bar{c}}(x_C), P_{\bar{d}}(x_D)$ と表現できる．各指標の一致確率の積，すなわち式 (6) の値が大きくなる時，全指標において一致確率が高い時であり，これを $Same$ 領域と定義する．

$$P_{\text{same}} = P_a(x_A)P_b(x_B)P_c(x_C)P_d(x_D) \tag{6}$$

反対に不一致確率の積，式 (7) の値が大きくなる場合は，すべての指標において不一致確率が高い時であるから，これを $Not - same$ 領域と定義する．

$$P_{\text{not\_same}} = P_{\bar{a}}(x_A)P_{\bar{b}}(x_B)P_{\bar{c}}(x_C)P_{\bar{d}}(x_D) \tag{7}$$

$P_a(x_A)P_{\bar{b}}(x_B)P_c(x_C)P_d(x_D)$ のような値が大きくなる場合，A, C, D 指標の一致確率が高く，B 指標の不一致確率が高い時である．このように指標が拮抗している積の和を $Unknown$ 領域と定義し，式 (8) が成り立つ．

$$P_{\text{unknown}} = 1 - P_{\text{same}} - P_{\text{not\_same}} \tag{8}$$

$Unknown$ 領域が大きい場合は，計算機のみでの判定が困難なペアであり，人間による介入が必要と考えられる．

3.4　近傍グラフ

近傍グラフは，データを表すノードと候補ペアを表すエッジで構成され，そのグラフを操作することで同定処理を進める．一致と考えられるペアを 1 つのノードに集約し，不一致と考えられるペアのエッジを切断する．エッジがなくなった段階で同定処理は終了となり，残ったノード群が同定結果となる．

3.4.1　構　　築

近傍グラフの構築時は，書誌データを全てノードとして配置し，距離学習器で出力される距離が一定の距離よりも小さいノードペアに対しエッジを張る．

3.4.2　スコアの計算

スコアは，2 つのノードについて共有している接続ノードの個数と定義する．$\mathcal{N}(v_i) \subseteq \mathcal{V}$ がノード v_i と接続されているノード集合とするとき，式 (9) で計算される．

$$\boldsymbol{Score}(v_i, v_j) = |\mathcal{N}(v_i) \cap \mathcal{N}(v_j)| \tag{9}$$

3.4.3　縮 約 操 作

SUN 推論器およびクラウドソーシングによって，一致と考えれられるペアに対して縮約する．例えば (v_i, v_j) が一致と考えられるペアで v_i に集約する場合，まず，$v_i.group$ に $v_j.group$ が属することを記録する．次に v_j に張られているエッジを v_i に張り直し，ノード v_j を削除する．

3.4.4　切 断 操 作

SUN 推論器およびクラウドソーシングによって，不一致と考えれられるペアに対しては切断処理を行う．例えば (v_i, v_j) が一致と考えられるペアの場合，この 2 つのノードに張られたエッジを削除する．

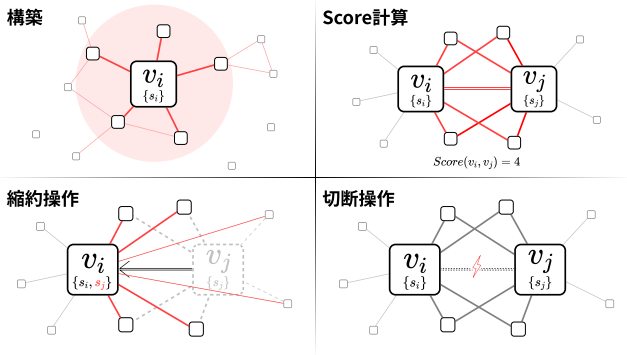


図 5　近傍グラフの操作

4　提 案 手 法

提案手法は「学習フェーズ」「公理矛盾検出フェーズ」「再学習フェーズ」「同定フェーズ」の 4 段階のワークフローによって構成される．これら一連のワークフロー，Algorithm，データ入出力の関係を図 6 にまとめた．図中の (∗) は，既存手法のワークフローの再現であり，再学習しない手法として比較検証時に利用する．

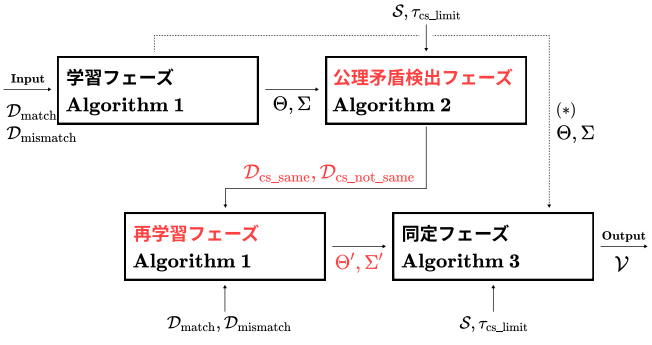


図 6　提案ワークフロー概要

4.1 学習フェーズ

このフェーズでは **Algorithm 1** を基に，教師データ $\mathcal{D}_{\text{match}}, \mathcal{D}_{\text{mismatch}}$ を用いて，距離学習器と SUN 推論器のパラメータ Θ, Σ を学習する．

Algorithm 1 Training

Input: Same pairs (teaching data) $\mathcal{D}_{\text{match}}$ and
Not-same pairs (teaching data) $\mathcal{D}_{\text{mismatch}}$
Output: Metric learning parameters Θ and
SUN reasoner parameters Σ

$\Theta \leftarrow \text{MetricLearning}(\mathcal{D}_{\text{match}}, \mathcal{D}_{\text{mismatch}})$
$\Sigma \leftarrow \text{FittingToPDF}(\mathcal{D}_{\text{match}}, \mathcal{D}_{\text{mismatch}})$
return Θ, Σ

4.2 公理矛盾検出フェーズ

このフェーズでは **Algorithm 2** を基に，近傍グラフを構築し，その近傍グラフ内のエッジで接続された 3 つのノードについて，公理に矛盾した推論器の判定パターンを検出する．矛盾した組み合わせについて，修正の必要順にランク付けをし，クラウドソーシングの上限 $\tau_{\text{cs\_limit}}$ に到達するまで問い合わせる．公理に矛盾するパターンとは，図 7 のことであり，例えば 3 つのデータの関係について $s_i = s_j, s_j = s_k$ but $s_k \neq s_i$ のような現実世界に存在しない推論パターンを指す．

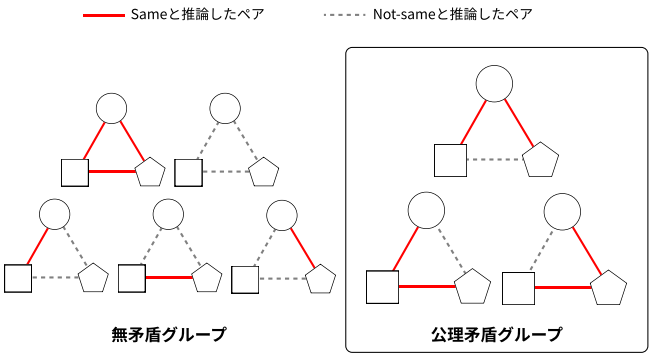


図 7　公理に矛盾しているグループ

本研究では 3 つのノードによって構成される 3 つのエッジのみに着目し，それ以外の多角形については考慮しない．また，グラフ上ではノード 3 つに対して 2 つしかエッジが存在しない場合も，エッジがあるものとみなして検出の対象とする．

推論器の修正において，情報量が多いと考えられる以下の戦略によりクラウドソーシングに問い合わせて修正を行うペアを決定する．

I. Same 領域の大きい不一致ペアや，Not-same 領域の大きい一致ペアを検出し修正する

II. 絶対数が少ないと考えられる一致ペアは情報量が多いことを考慮し，一致ペアが多く含まれていると考えられるペアから検出する

よって，公理矛盾の可能性がある組み合わせの優先順は，SUN 推論器が判定した $Same, Unknown, Not-same$ ペアをそれぞれ S, U, N と表す時，式 (10) となる．

$$S-S-N \succcurlyeq S-S-U \succcurlyeq S-N-U \succcurlyeq S-U-U \succcurlyeq N-U-U \succcurlyeq U-U-U \quad (10)$$

クラウドソーシングの問い合わせにより得られた一致ペアを $\mathcal{D}_{\text{cs\_same}}$，不一致ペアを $\mathcal{D}_{\text{cs\_not\_same}}$ と表す．

Algorithm 2 Discover and correct axiomatic inconsistency

Input: Target database $\mathcal{S}$, Metric learning parameters Θ,
SUN reasoner parameters Σ, and
Crowdsourcing limit $\tau_{\text{cs\_limit}}$
Output: Same pairs obtained by crowdsourcing $\mathcal{D}_{\text{cs\_same}}$ and
Not same pairs obtained by crowdsourcing $\mathcal{D}_{\text{cs\_not\_same}}$

$\mathcal{G} \leftarrow \text{NN-GraphConstructor}(\mathcal{S}, \Theta)$
$\mathcal{T}_{\text{error}} \leftarrow \text{AxiomaticError}(\mathcal{G}, \Theta, \Sigma)$
Sort $\mathcal{T}_{\text{error}}$ based on priority rules using Eq.(10);
cs_counter $\leftarrow 0$
for (s_i, s_j, s_k) in $\mathcal{T}_{\text{error}}$ **do**
  if cs_counter $\geqq \tau_{\text{cs\_limit}}$ **then** break
  for (a, b) **in** Combination(i, j, k) **do**
    if Crowdsourcing$(s_a, s_b) =$ **True then**
      $\mathcal{D}_{\text{cs\_same}} \leftarrow \mathcal{D}_{\text{cs\_same}} \cup \{(s_a, s_b)\}$
    else
      $\mathcal{D}_{\text{cs\_not\_same}} \leftarrow \mathcal{D}_{\text{cs\_not\_same}} \cup \{(s_a, s_b)\}$
    end if
  end for
  cs_counter $\leftarrow$ cs_counter $+ 1$
end for
return $\mathcal{D}_{\text{cs\_same}}$, $\mathcal{D}_{\text{cs\_not\_same}}$

function COMBINATION(i, j, k)
  return $\{(i, j), (i, k), (j, k)\}$
end function

4.3 再学習フェーズ

このフェーズでは公理矛盾検出フェーズでクラウドソーシングにより得られた一致ペア $\mathcal{D}_{\text{cs\_same}}$ と不一致ペア $\mathcal{D}_{\text{cs\_not\_same}}$ について，人間の介入によって得られた信頼度の高いデータとして教師データに追加し，距離学習器と SUN 推論器のパラメータ Θ, Σ の更新を行う．**Algorithm 1** の一致ペアの入力を $\mathcal{D}_{\text{match}} + \mathcal{D}_{\text{cs\_same}}$，不一致ペアの入力を $\mathcal{D}_{\text{mismatch}} + \mathcal{D}_{\text{cs\_not\_same}}$ にし，パラメータ Θ', Σ' として更新する．

4.4 同定フェーズ

このフェーズでは **Algorithm 3** を基にターゲットデータベース $\mathcal{S}$ のエンティティマッチングを行う．距離学習器および SUN 推論器には，再学習フェーズで更新したパラメータ Θ', Σ' を使用し，更新した Θ' で，近傍グラフ $\mathcal{G}$ を再構築する．

クラウドソーシング数の上限に達するまで，計算機は $P_{\text{same}} > \tau_{\text{same}}$ を満たす場合に縮約操作し，$P_{\text{not\_same}} > \tau_{\text{not\_same}}$ を満たす場合に切断操作をする．この条件をどちらも満たさないペアは，クラウドソーシングにより人間に判断を仰ぐ．τ_{same} と $\tau_{\text{not\_same}}$ は，要求する精度に応じて決定する．

クラウドソーシング数の上限に達した後は，$P_{\text{same}} > P_{\text{not\_same}}$ を満たす場合に縮約し，満たさない場合を切断する．

近傍グラフ $\mathcal{G}$ の全てのエッジが $0(|\mathcal{E}| = 0)$ になると終了となり，グラフに残ったノード群 $\mathcal{V}$ が成果物となる．

Algorithm 3 Identification

Input: Target database $\mathcal{S}$, Metric learning parameters Θ,
SUN reasoner parameters Σ,
Crowdsourcing limit $\tau_{\text{cs\_limit}}$,
Same pairs obtained by crowdsourcing $\mathcal{D}_{\text{cs\_same}}$, and
Not same pairs obtained by crowdsourcing $\mathcal{D}_{\text{cs\_not\_same}}$
Output: A set of nodes in the NN-Graph $\mathcal{V}$

$\mathcal{G} \leftarrow$ NN-GraphConstructor($\mathcal{S}$, Θ)

for (s_i, s_j) **in** $\mathcal{D}_{\text{cs\_same}}$ **do**
  Contraction(v_i, v_j)
end for
for (s_i, s_j) **in** $\mathcal{D}_{\text{cs\_not\_same}}$ **do**
  Remove(v_i, v_j)
end for

cs_counter $\leftarrow 0$
while $\mathcal{E} \neq \varnothing$ **do**
  $(v_i, v_j) \leftarrow \underset{(v_i,v_j)\in\mathcal{E}}{\arg\max}$ Score (v_i, v_j)
  P_{same}, $P_{\text{not\_same}} \leftarrow$ SunResoner((s_i, s_j), Θ, Σ)
  if cs_counter $< \tau_{\text{cs\_limit}}$ **then**
    if $P_{\text{same}} > \tau_{\text{same}}$ **then**
      Contraction(v_i, v_j)
    else if $P_{\text{not\_same}} > \tau_{\text{not\_same}}$ **then**
      Remove(v_i, v_j)
    else
      if Crowdsourcing(s_a, s_b) = **True then**
        Contraction(v_i, v_j)
      else
        Remove(v_i, v_j)
      end if
      cs_counter $\leftarrow$ cs_counter + 1
    end if
  else
    if $P_{\text{same}} > P_{\text{not\_same}}$ **then**
      Contraction(v_i, v_j)
    else
      Remove(v_i, v_j)
    end if
  end if
end while
return $\mathcal{V}$

function SCORE(v_i, v_j)
  return $|\mathcal{N}(v_i) \cap \mathcal{N}(v_j)|$
end function

function CONTRACTION(v_i, v_j)
  v_i.group $\leftarrow$ v_i.group $\cup$ v_j.group
  for v **in** $\mathcal{N}(v_j)$ **do**
    $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_i, v)\}$
  end for
  $\mathcal{V} \leftarrow \mathcal{V} \setminus \{v_j\}$
end function

function REMOVE(v_i, v_j)
  $\mathcal{E} \leftarrow \mathcal{E} \setminus \{(v_i, v_j)\}$
end function

5 実　　験

提案手法に基づき，ターゲットデータベース $\mathcal{S}$ の規模とクラウドソーシング数 $\tau_{\text{cs\_limit}}$ を変更しながら同定処理を行うことで，公理系による矛盾検出および再学習の効果を示す．5.1〜5.4 は，利用するデータや検証方法等を記述し，5.5 以降は，実験内容と結果を示す．

5.1 利用データ

本実験では，全国 963 の公立図書館が保有する書誌データ約 1211 万件が格納されたデータベースから抽出した一部を実験データとする．ワークフローで利用するデータの属性は，「書誌名」「著者名」「出版社名」「出版年」の 4 つのみを利用する．書誌の固有識別番号である「ISBN」は，

- $\mathcal{S}_n, \mathcal{D}_{\text{match}}, \mathcal{D}_{\text{mismatch}}$ の生成 (入力データの生成)
- 同定完了後の $\mathcal{V}$ の検証 (出力データの検証)
- クラウドソーシングの回答生成

のみ利用し，計算機が係わるワークフローの処理に利用しない．

教師データに用いる一致ペア $\mathcal{D}_{\text{match}}$ と不一致ペア $\mathcal{D}_{\text{mismatch}}$ を無作為に各 30000 ペア抽出する．ターゲットデータベースは，500 件の書誌データで構成されるデータベース 10 個 $(\mathcal{S}_i(0 \leqq i \leqq 9))$，2000 件の書誌データで構成されるデータベース 10 個 $(\mathcal{S}_j(10 \leqq j \leqq 19))$ をそれぞれ抽出する．なお，ターゲットデータベースに含まれる書誌データは，表 (3) の条件を満たすようにランダムに抽出する．カッコ外は $\mathcal{S}_0$〜$\mathcal{S}_9$，カッコ内は $\mathcal{S}_{10}$〜$\mathcal{S}_{19}$ の内訳である．

表 3　ターゲットデータベース $\mathcal{D}$ の内訳

	グループ数	一致ペア数	データ数
3 件一致グループ	75 (300)	225 (900)	225 (900)
2 件一致グループ	125 (500)	125 (500)	250 (1000)
1 件独立	25 (100)	0 (0)	25 (100)
計	225 (900)	350 (1400)	500 (2000)

5.2 SUN 推論器のパラメータ

SUN 推論器で必要なパラメータを $\tau_{\text{same}} = 0.7, \tau_{\text{not\_same}} = 0.5$ と設定し，計算機が一致ペアと判定する条件を厳しくした．

5.3 クラウドソーシング

本実験では実際のクラウドソーシングは行わず，問い合わせには必ず正解を返すシミュレーション実験として検証した．

5.4 結果の検証方法

検証には，F1 値と完全一致率の 2 つの指標を用いる．いずれの指標も 1 に近いほど精度が高いことを示す．

F1 値は「再現率」と「適合率」によって計算が可能．

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{11}$$

完全一致率は，グルーピングが正しく成功した程度を表す．

正解一致ペアを使って正しく縮約した際に得られるターゲットデータベースのノード集合 $\mathcal{V}_{\text{correct}}$ と，ワークフロー完了後に得られるノード集合 $\mathcal{V}$ を用いて，式 (12) により定義される．

$$\text{FullAgreementRate} = \frac{|\mathcal{V} \cap \mathcal{V}_{\text{correct}}|}{|\mathcal{V}_{\text{correct}}|} \tag{12}$$

5.5 再学習の有無による精度の検証 (実験 1)

再学習の有無により，縮約結果の違いについて混同行列，F1 値および完全一致率で示す．クラウドソーシングの上限について，ターゲット書誌データ $\mathcal{D}_0$ は，$\tau_{\text{cs\_limit}} = 600$，$\mathcal{D}_{10}$ は，$\tau_{\text{cs\_limit}} = 5000$ とする．

表 4　$\mathcal{S}_0$ の同定処理結果

		No-retraining		Retraining	
		推論		推論	
		一致ペア	不一致ペア	一致ペア	不一致ペア
正解	一致ペア	307	43	307	43
	不一致ペア	31	124369	35	124365
適合率		0.9083		0.8977	
再現率		0.8771		0.8771	
F1 値		0.8924		0.8873	
完全一致率		0.8133 (183 / 225)		0.8222 (185 / 225)	

表 4 より，データが 500 件の場合，精度について再学習が無くてもさほど悪く無く，ほとんど変わらない．

表 5　$\mathcal{S}_{10}$ の同定処理結果

		No-retraining		Retraining	
		推論		推論	
		一致ペア	不一致ペア	一致ペア	不一致ペア
正解	一致ペア	1102	298	1197	203
	不一致ペア	11096	1968504	20	1997580
適合率		0.09034		0.9836	
再現率		0.7871		0.8550	
F1 値		0.1621		0.9148	
完全一致率		0.3789 (341 / 900)		0.8444 (760 / 900)	

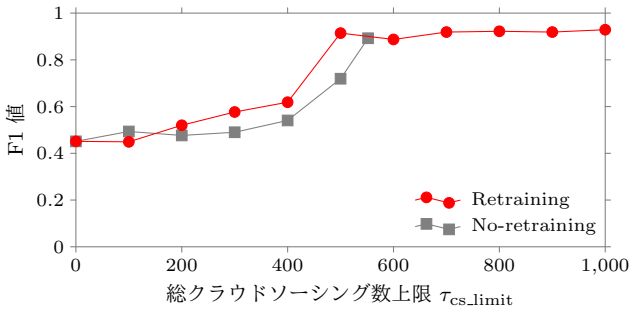
表 5 より，データが 2000 件の場合，再学習しない場合は著しく性能が低下するが，再学習を行った場合は低下しない．

5.6 総クラウドソーシング上限数変動と精度の検証 (実験 2)

再学習の有無により，クラウドソーシング数の上限 $\tau_{\text{cs\_limit}}$ を変更することによる F1 値と完全一致率の変動を検証する．ターゲット書誌データは $\mathcal{S}_0$ と $\mathcal{S}_{10}$ を利用し，$\mathcal{S}_0$ については 100 ずつ，$\mathcal{S}_{10}$ については 1000 ずつ $\tau_{\text{cs\_limit}}$ を変動させる．

図 8 より，データ数が 500 件の場合は，再学習の有無に関わらずクラウドソーシング上限数を 600 件に設定したあたりで性能が頭打ちしており，結果も有意に差があるとは言えない．

図 9 より，データ数が 2000 件の場合は，顕著に差が出ている．再学習しない場合は 6000 件強の問い合わせでアルゴリズムが終了しており，精度も大きくは上がらない．一方で，再学習ありの場合は 5000 件ほどで性能は頭打ちになるものの，精度の高い同定処理を行うことができる．



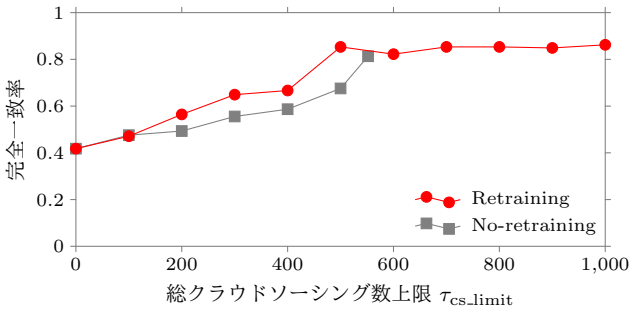
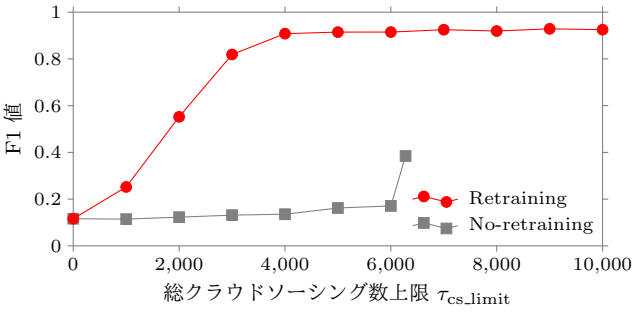


図 8　$\mathcal{S}_0$ の F1 値と完全一致率の推移



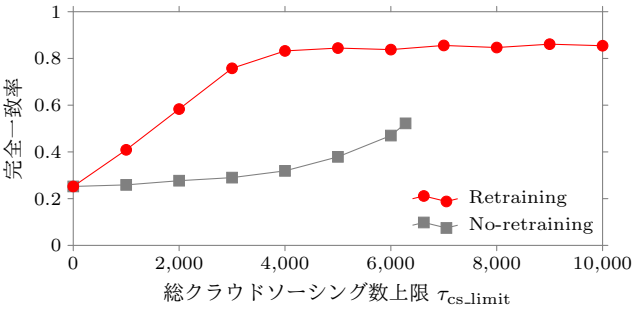


図 9　$\mathcal{S}_{10}$ の F1 値と完全一致率の推移

5.7 複数のデータベース比較による有意性の検証 (実験 3)

ターゲットデータベース $\mathcal{S}_0$〜$\mathcal{S}_{19}$ の 20 個について，各システムの精度がデータベースにどの程度依存しているかを検証する．クラウドソーシングの上限を，500 件のデータを持つ $\mathcal{S}_0$〜$\mathcal{S}_9$ は $\tau_{\text{cs\_limit}} = 600$，2000 件のデータを持つ $\mathcal{S}_{10}$〜$\mathcal{S}_{19}$ は $\tau_{\text{cs\_limit}} = 5000$ で固定する．

図 10 より，データ数が 500 件の場合は，精度の最高値は再学習の有無に起因しないが，再学習なしに比べて再学習ありの方が，様々なデータベースに対し安定的である．

図 11 より，データ数が 2000 件の場合は，より顕著であり，再学習ありの場合は，データベースに依らず安定的に高精度な同定を行うことができる．

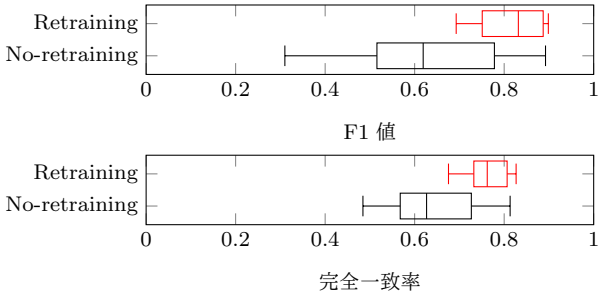


図 10　$\mathcal{S}_0$〜$\mathcal{S}_9$ の F1 値と完全一致率の統計

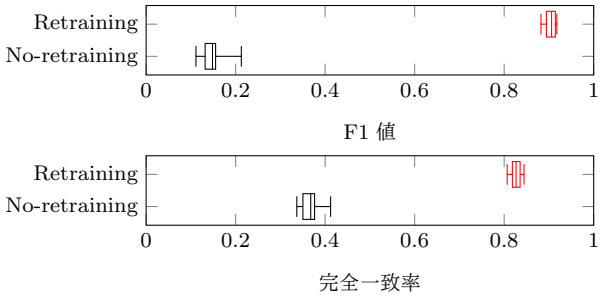


図 11　$\mathcal{S}_{10}$〜$\mathcal{S}_{19}$ の F1 値と完全一致率の統計

6　議　　論

6.1　公理系矛盾の問い合わせ戦略

本研究では，公理系矛盾の問い合わせを，式 (10) の戦略により行った．*Unknown* のほとんどが不一致ペアであることなどを踏まえて決定したが，これに捉われず，戦略の違いによる精度とクラウドソーシング数の差については検討の余地がある．

6.2　パラメータの設定

このワークフローは，動作のために $\tau_{\text{same}}, \tau_{\text{not\_same}}, \tau_{\text{cs\_limit}}$ など事前に決めなければならないパラメータが存在する．これらの適切な値はターゲット $\mathcal{S}$ に依存していると考えられ，現段階では実験を行うまで適切な値が得られない問題がある．データ特性の事前分析による，適切なパラメータを事前に得られるような評価指標の考案，あるいは，ワークフロー中に適切なパラメータに修正できるシステムの構築に取り組みたい．

7　ま　と　め

本論文では，Human-in-the-loop を用いた同定処理において，規模の増加に伴い組み合わせ爆発を引き起こし，精度低下およびコストの大幅に増加について問題を提起した．その解決にあたり，再学習を取り入れた Human-in-the-loop エンティティマッチング手法，および再学習に効果的なデータを発見するための公理系矛盾を利用した手法を提案した．

実験では，データセットの規模とクラウドソーシング上限数に着目し，再学習の有無による違いを検証した．提案手法は，データセットの規模を拡大しクラウドソーシング上限を課しても，安定的な精度を維持できることが示唆された．一方で，公理矛盾の問い合わせ戦略など改善の余地が見られる点もある．

今後の展望として，「公理矛盾検出フェーズ」と「再学習フェーズ」を複数回繰り返すことによる精度とクラウドソーシングコストの検証と，シミュレーションを脱したクラウドワーカの回答精度および手法全体の精度の検証を行いたい．前者は，再学習を複数回行うことで，更新の度に効果的な公理矛盾ペアを少ない問い合わせ回数で得ることができ，精度は維持しつつも，クラウドソーシング数の減少に繋がることが期待される．後者は，実際のクラウドソーシングの利用により，タスクの難易度や結果の信頼性を含めた提案手法全体の検証を行いたい．

謝　　辞

本研究の一部は JSPS 科研費 (JP22H00508) と JST CREST(Grant Number JPMJCR22M) の支援を受けたものである．ここに謝意を示す．

文　　献

[1] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.*, Vol. 53, No. 6, dec 2020.

[2] Matthew A. Jaro. Advances in record-linkage methodology as applied to matching the 1985 census of tampa, florida. *Journal of the American Statistical Association*, Vol. 84, No. 406, p. 414, 06 1989.

[3] 上田洋, 村上晴美, 辰巳昭治. 集中型横断検索システムのための自動書誌同定処理の検討. 第 57 回日本図書館情報学会研究大会発表要綱, pp. 29–32, 2009.

[4] Harada Takashi, Fukushima Yukihiro, Sato Sho, Tsuruta Misato, Yoshimoto Ryuji, and Morishima Atsuyuki. Advancement of bibliographic identification using a crowdsourcing system. *Proceedings of the 9th Asia-Pacific Conference on Library & Information Education and Practice(A-LIEP 2019)*, pp. 71–82, 11 2019.

[5] Guoliang Li. Human-in-the-loop data integration. *VLDB Endowment*, Vol. 10, No. 12, pp. 2006–2017, August 2017.

[6] Naofumi Osawa, Hiroyoshi Ito, Yukihiro Fukushima, Takashi Harada, and Atsuyuki Morishima. Bubble : A quality-aware human-in-the-loop entity matching framework. In *The The 5th IEEE Workshop on Human-in-the-loop Methods and Future of Work in BigData (IEEE HMData2021)*, pp. 3557–3565, December 2021.

[7] Stephen Robertson, S. Walker, S. Jones, M. M. Hancock-Beaulieu, and M. Gatford. Okapi at trec-3, January 1995.

[8] Yao Zhu, Hongzhi Liu, Zhonghai Wu, and Yingpeng Du. Relation-aware neighborhood matching model for entity alignment, 2020.

[9] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. In J. Cowan, G. Tesauro, and J. Alspector, editors, *Advances in Neural Information Processing Systems*, Vol. 6. Morgan-Kaufmann, 1993.

[10] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2, pp. 1735–1742, 2006.

[11] William Winkler. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. *Proceedings of the Section on Survey Research Methods*, pp. 354–359, 01 1990.

[12] Vladimir I Levenshtein, et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10, pp. 707–710. Soviet Union, 1966.