DEIM2021 B21-5

制約脆弱データに対するデータクリーニングのための不整合候補検出

大森　弘樹† 　清水　敏之† 　吉川　正俊†

† 京都大学大学院情報学研究科　〒 606–8501 京都府京都市左京区吉田本町
E-mail: †hiroki@db.soc.i.kyoto-u.ac.jp, ††{tshimizu,yoshikawa}@i.kyoto-u.ac.jp

あらまし　近年，ビッグデータ解析や機械学習利用の需要が高まっている．これらの手法は分析用に整備されたデータの入力を前提としていることが多く，データ整備のためにデータクリーニング手法の重要性も高まっている．しかし，データ生成時の制約が乏しく，データクリーニングに有用な一貫性制約が存在しない場合は，制約を利用した既存手法の適用が困難である．また，統計的な推測が難しい不整合な値を含んでいる場合には学習に基づく既存手法の適用も困難である．本研究では制約脆弱なデータに対して一貫性制約や教師データを必要としない，エンティティ解決手法を応用した再現率重視の不整合候補検出手法を適用することを考えた．制約脆弱なデータの例として科学メタデータを挙げ，既存のデータクリーニング手法と比較を行う．また，提案手法の計算コストを低減する手法を検討し，その手法を適用した場合の影響を観察した．

キーワード　関係データ，データクリーニング，エンティティ解決，データ品質

1 はじめに

現在，ビッグデータの利用は官民共に盛んである．そうしたデータの利用分野の一つとして機械学習があり，機械学習の利活用もまた盛んである．機械学習やデータの分析を行うためには，空値の補填や値に一貫性をもたせるなどのデータ整備が重要となる一方で，データ整備は多大な労力を必要とする [1,2]．

データ整備の際に，関数従属性や辞書などの外部情報を表現した，データクリーニングに有用な一貫性制約を利用した既存のデータクリーニング手法が適用される場合が多い．これらの手法は，データクリーニングに有用な一貫性制約に違反している値をエラーの候補として発見し，修正を行う手法である．しかし，そのようなデータクリーニングに有用な一貫性制約が存在しないデータには，制約を利用する既存手法を適用することが困難であると考えた．我々はこのようなデータクリーニングに有用な一貫性制約が存在しないデータを制約脆弱データと呼ぶことにした．本研究では，制約脆弱データに対して，一貫性制約や教師データを必要としない，エンティティ解決手法を応用した再現率重視の不整合候補検出手法 [3] を適用することを考えた．以下，この手法を IDER (Inconsistency Detection based on Entity Resolution) 手法 [3] と呼ぶこととする．

本研究が適用する IDER 手法は，単純な誤字脱字などの誤りだけではなく，同義語や語順が異なる語などの単純な文字列のパターンで捉えることの難しい不整合な値も検出可能である．不整合な値は，例えば入力時に値の候補を提示したり，望ましい形式になっていない不正な値を入力できないようにするといったようなデータ生成時の制約が乏しく，かつ，制約脆弱であるデータに多く含まれていると本研究では考えている．そのような不整合な値は，単純には辞書などの外部知識を利用することでクリーニングが可能である．しかし，辞書を作成するには，人間の多大な労力が必要であり，利用可能な外部知識を作成するために必要とされるコストが高いことが多い．IDER 手法はそのようなコストを軽減するために，データの値の中から不整合な値である可能性があるものを候補として発見し，人間，特にデータ管理者をユーザとして想定し，ユーザに不整合な値の候補の提示を行うことで，効率よく不整合な値の修正を行うことを想定している．そのため，IDER 手法は再現率を重視している．

	カテゴリ	提供者	作成機関	収録年
t_1:	ocean	Omori Hiroki	KU	2011
t_2:	forest	Hiroki OMORI	Kyoto univ.	2016
t_3:	soils	Taro YAMADA	Kobe univ.	2017
t_4:	ocean	Hiroki OMORI	Kobe univ.	2018

図 1: 不整合なメタデータの例

修正に人間を必要とするような不整合を含むデータの例を図 1 に示す．この例では 1 つの組が 1 つのデータに対応しており，それぞれの組に t_1 から t_4 の ID を割り当てている．カテゴリの列はそのデータの属するカテゴリを，提供者の列はそのデータの提供者の名前を，作成機関の列はデータを作成した機関の名前を，収録年はそのデータが収録された年を表している．図 1 の例では，提供者の欄について，t_1 とそれ以外の組とで姓名の順番が異なっており，不整合が起きている．こうした姓名の順番の判断などは，他の値がどのような姓名の順番になっているかなどの情報を考慮する必要があり，単純に辞書を用いるなどといった機械的な判断が難しい．また，機関の略称として “KU” が用いられているが，その略称に対応する機関名は “Kyoto univ.” と “Kobe univ.” の 2 種類が表の中に存在しており，これもまた機械的な判断が難しい．本研究では，IDER 手法が実際の制約脆弱データにおいても，こうした機械的に判断することが難しい不整合な値の候補を検出することができているかについて実験を行った．

データクリーニング分野における既存研究は二種類に大別でき，一貫性制約などのデータクリーニングに有用な制約を利用する手法と，人間がラベル付けをした教師データを利用する機械学習を用いた手法が存在している．本研究は，制約脆弱データに対して，制約を利用する手法は適用しづらく，機械学習を用いた手法においても，制約脆弱データにおける不整合な値に適用する場合は，不整合な値の種類を網羅するような教師データを作成することが難しいと考えた．制約を利用する手法とは，属性間の従属性などの，データクリーニングに有用な制約を利用する手法のことである [4]．こうした制約を利用する手法には再現率が低くなるという欠点が存在している．その欠点を解決するため，機械学習を用いた手法が提案され，高い適合率と再現率でエラーを検出することが可能となっている [5]．機械学習を用いた手法はラベル付けのコストがあり，クリーニング対象のデータの全体ではなく一部にラベル付けを行い，教師データとして用いる．しかし，教師データ外に未知のパターンの不整合が存在しているような状況も考えられ，そうした不整合は機械学習を用いた手法による検出が難しくなる．誤字脱字といった単純なエラーは，少ない教師データでもデータ全体のエラーの傾向を掴むことが可能だが，不整合な値は，教師データに含まれていない場合，機械学習を用いた手法による対処が困難であると我々は考えた．

そのため、こうした教師データ外に未知のパターンの不整合が存在してしまう状況では，制約を利用する手法の方が適しているが，制約脆弱データを対象にしている場合，明示的な制約が存在せず，制約を利用する手法を適用することが難しいと考えた．本論文では，IDER 手法は辞書などのデータクリーニングに有用な情報を人手で作成する際の補助に使用可能であると考えており，その場合は，不整合な値の候補を取りこぼさずより多く検出できることが重要であると考えた．また，実際に不整合な値かどうかを判定することは難しい問題であり，人手の確認を必要とするため，より多くの不整合な値の候補を人間に提示することが重要であると考えた．そのため，IDER 手法は，単語の分散表現と機械学習を用いることで，通常の制約よりも更に柔軟で緩和された，再現率が高くなるかわりに適合率が低くなるような制約を用いて不整合候補を検出することを目的としている [3].

我々は，既存の制約を用いるデータクリーニング手法は，制約脆弱データにおける不整合候補を検出することは困難であると考えた．制約脆弱データにおける IDER 手法の有効性を確認するため，既存の制約を用いるデータクリーニング手法と IDER 手法とを再現率の観点から比較することとした．既存の制約を用いるデータクリーニング手法 [4] を，IDER 手法を適用したデータと同じ制約脆弱データに適用し，結果を比較した．また，IDER 手法は，入力として与えられる，二値分類に用いる閾値によって，結果が変化する．その閾値の変化に応じてどのように IDER 手法の結果が変化するのかについても観察を行った．

IDER 手法は不整合の検出を主眼とした手法であるが，誤字脱字などの単純なエラーも不整合の一種であると考えると，エラー検出手法としても利用可能だと思われる．既存の制約を利用したデータクリーニング手法 [4] と IDER 手法を，誤字脱字などの単純なエラーの検出能力を評価するための，制約脆弱でないデータに適用し，その結果を比較した．そして IDER 手法は，エラー検出手法として制約脆弱でないデータに適用された場合，再現率を重視した手法となっており，既存の制約を利用したデータクリーニング手法 [4] では検出できなかった不整合も検出できることを確認した．更に，IDER 手法の利用にあたり，タプルブロッキングと呼ばれる手法を用いて，教師データの数を絞ることで，機械学習の訓練のための計算コストを下げることを検討した [6]．タプルブロッキングを適用した場合の IDER 手法の結果と適用しなかった場合の IDER 手法の結果の比較を行った．

表 1: 既存手法 [6] と IDER 手法との差異

手法名	目的	エンティティの単位	検出したい対象
既存手法 [6]	タプルの重複解消	タプル全体	エンティティが同じタプルのペア
IDER 手法	値の不整合解消	一つのセル	エンティティが同じだが表記が異なる値のペア

本論文の構成を以下に示す．第 2 節では本論文が適用する手法である IDER 手法がどのようなものかを説明し，計算コストを下げるためにタプルブロッキングを IDER 手法に適用する方法について説明を行う．第 3 節では実際の制約脆弱データに対して IDER 手法と既存の制約を利用したデータクリーニング手法を適用し，その結果の比較を行う．第 4 節では IDER 手法の再現率に関して，制約脆弱でないデータについて行った実験に基づいて考察を行う．第 5 節では実際のデータに対してタプルブロッキングを適用した結果について観察を行う．第 6 節では本論文のまとめを述べ，今後の課題について議論する．

2 エンティティ解決手法を応用した不整合候補検出手法

この節では，本研究が適用する，エンティティ解決手法を応用した不整合候補検出手法である IDER 手法 [3] に関する説明を行う．

2.1 応用するエンティティ解決手法

IDER 手法は Muhammad らのエンティティ解決手法 [6] を応用して不整合検出を行う．この既存手法 [6] と IDER 手法との主だった差異について表 1 に示す．まず，既存手法 [6] と IDER 手法では目的が異なっている．既存手法 [6] は同じエンティティのタプルが誤って重複して入力されている状況を解消するために，タプルの重複を検出することを目的としているのに対して，IDER 手法は表記ゆれがあるなどの一つのエンティティに対して複数の表記が存在している状況を解消するために，不整合な値の候補の検出を目的としている．そのため，エンティティの捉え方が異なっており，既存手法 [6] はエンティティは各タプルに対応して存在していると考えており，IDER 手法はエンティティはタプルより細かく各セルに対応して存在

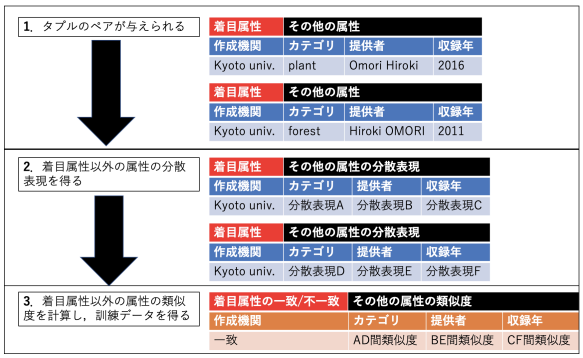


図 2: ニューラルネットを教師データを作成するまでの流れ

していると考えている．以上のことから，両手法の検出したい対象も異なっており，既存手法 [6] がエンティティが同じと推測されるタプルのペアを検出するのに対して，IDER 手法はエンティティが同じだと推測されるが実際の表記は異なる値のペアを検出するという点で異なっている．

2.2 入　　力

IDER 手法は，不整合候補検出の対象である関係データと，そのデータに入っている単語を含む学習済みの単語ベクトル集合を入力として受け取る．この単語ベクトル集合は，対象とする関係データの全単語を含んでいる必要はなく，関係データの単語の一部を含んでいるだけで良い．入力とする学習済み単語ベクトル集合としては，IDER 手法は手法 [6] と同じく GloVe [7] という単語をベクトルとして分散表現する手法により，一般的な Web ページなどのコーパスから学習された学習済みの単語ベクトル集合を利用することにした．

2.3 エンティティ解決

IDER 手法は，属性の一つを着目属性とし，その属性の値をなんらかのエンティティに対応するラベルだと捉えてエンティティ解決を行う．つまり，全てのタプルのペアに対して，そのタプルのペアの着目属性の値に対応するエンティティが一致するかどうかの予測を行う．本研究は，着目属性以外の属性の値は，着目属性の値に対応するエンティティの属性と捉えており，IDER 手法はそれらの属性の値から，タプルの各ペアの着目属性の値が一致しているかどうか判定できるようにニューラルネットを訓練することで，エンティティ解決を行う．ニューラルネットの教師データを作成するまでの流れの概要を，図 1 と同じスキーマを持つデータをイメージした例を用いて図 2 に示し，以下にその概要を説明する．

最初に，図 2 の 1 番目のステップのように，全てのタプルのペアが与えられる．図 2 は例として，そのうち一つのタプルのペアに関して説明している．続いて，各タプルのペアの全ての単語について，与えられた学習済みの単語ベクトル集合を用いて，単語間の共起関係を考慮しつつ単語ベクトルが作成される．それらの単語ベクトルを用いて，図 2 の 2 番目のステップで，各属性で分散表現 A，B，C，D，E，F が得られているように，各属性の値一つにつき一つの分散表現を得る．そして最後に，図 2 の 3 番目のステップで，カテゴリの属性では分散表現 A と D の，提供者の属性では分散表現 B と E の，収録年の属性では分散表現 C と F の類似度を計算する．その結果，カテゴリの属性，提供者の属性，収録年の属性にそれぞれ対応した類似度である，AD 間類似度，BE 間類似度，CF 間類似度の値を持つ類似度ベクトルが一つ得られる．また，着目属性である作成機関の属性は Kyoto univ. という同じ値を持っているため，値が一致していることを意味するラベルを類似度ベクトルに対応づける．こうして得られた類似度ベクトルとラベルを教師データとしてニューラルネットに与え，ニューラルネットは着目属性の値が一致しているかどうかについての訓練を行う．

2.4 出　　力

続いて IDER 手法は，全てのタプルのペアに対して，学習済みのニューラルネットを用いて，着目属性の値が一致しているかどうかの予測を行う．予測の結果，着目属性の値が一致していると判断されたペアをポジティブ，そうでないペアをネガティブとして分類する．IDER 手法のニューラルネットの予測結果は，0 から 1 の値を取り，1 に近いほどポジティブ，反対に 0 に近いほどネガティブと判定している．ユーザが入力として与えた閾値を上回るとポジティブと判定し，下回るとネガティブと判定するようにした．

IDER 手法はこの結果の中で，誤分類に着目しており，中でも，ポジティブと誤分類された値について着目している．IDER 手法は，ポジティブとして誤分類されたタプルのペアの着目属性の値は，着目属性以外の属性が類似していることから対応しているエンティティが同一であることが疑われるが，エンティティに対応するラベルは異なっている状態と見なしている．そのため IDER 手法は，ポジティブに誤分類されたタプルのペアは，着目属性の値は同じエンティティを指しているのに，実際には誤って異なる表記の値が入力されている可能性があると考え，不整合な値の候補としてそのペアの着目属性の値を出力する．

2.5 タプルブロッキングによる訓練コストの低減

上記の IDER 手法では，ニューラルネットが，全てのタプルのペアに対して訓練を行う必要性があった．そのため，データのサイズを大きくすると指数的に計算コストが増大するという問題がある．そこで，既存研究 [6] で導入されていたタプルブロッキングを IDER 手法にも適用することで，なるべく分散表現が類似しているタプルでのみペアを作り，計算コストを低減させることを検討した．全てのタプルを，ニューラルネットの訓練に用いるのではなく，タプルを複数のブロックのいずれかに割り当て，その割り当てられたブロック内でのみタプルのペアを作成することで，その後のニューラルネットの訓練時の計算コストを低減することが可能である．タプルブロッキングでは，タプルをブロックに割り当てる際に，ハッシュ関数を用いる．本研究では，コサイン類似度が高いタプルを同じブロックに割り当てる性質を持つハッシュ関数を，ランダム超平面手法 [6] により作成し，利用した．この手法は，原点を通る超平面をランダムに生成し，タプルの分散表現の指す点が超平面の

どちら側にあるかによって，1 か-1 かどちらかの値を超平面ごとに一つ取得する．タプルの分散表現を t，超平面の単位ベクトルを v とすると，

- v と t の内積が 0 以上なら $h(t) = 1$
- v と t の内積が 0 未満なら $h(t) = -1$

となるようにハッシュ関数を作ることで，上記の超平面に対応する値をタプルごとに得ることができる．各 t に対して，K 回超平面をランダムに生成し，値を K 個得ることにより，タプルごとに K 桁のハッシュコードを得ることができる．それぞれのタプルの持つ K 桁のハッシュコードが一致している場合，同じブロックに割り当てられたとみなす．そして，同じブロックに割り当てられたタプル同士を組み合わせてペアを作る．以上のような，各タプルに対するランダム超平面を利用したハッシュ関数の K 回の適用，各タプルに対応した K 桁のハッシュコード取得，同じハッシュコードを持つタプル同士でのペアの作成の流れを L 回実施し，作成されたペアの和集合を得る．こうして得られた，全てのタプルを組み合わせてできたタプルのペアより数が減少したタプルのペアを，ニューラルネットの訓練に用いることで，訓練時の計算コストを低減することができる．

3 IDER 手法の評価

本研究は，実際のデータ生成時の制約が乏しい制約脆弱データとして，データ統合・解析システム DIAS (Data Integration and Analysis System)[1] に提供されたデータに付属しているメタデータの一部を対象に IDER 手法を実施した．また，その結果と比較するために，既存の制約を利用するデータクリーニング手法である HoloClean [4] を同じメタデータを対象に実施した．DIAS のメタデータは本来 XML 形式であるが，それを簡易的に 12 属性の関係データへと変換して各手法に適用した．一つの行が一つのデータセットに対応しており，今回は 426 データセットのメタデータに対して実験を行った．

このメタデータは，カテゴリ，制作された日時，提供されたデータセットの作成者，その所属機関，メタデータの著者やその所属機関などを属性として持つ．DIAS は観測によって得られた地球各地での多様な観測データを収集しており，多様な入力者から多くのデータが自由記述で記入されている．そのため，データ生成時の制約が乏しい上に，データクリーニングに有用な一貫性制約が成立しづらく，組織名や人名に対する辞書の作成コストも高いため辞書が存在していないことから，不整合な値を含んだ制約脆弱データを想定した実験用データとして有用である．

上記データの属性のうち，そのデータを識別するための ID を表すデータ ID の属性，そのデータのタイトルを表すタイトルの属性，そのデータについて問い合わせを行う連絡先の名前を表す連絡先の属性，そのデータについて問い合わせを行う機関名や連絡先の所属機関名を表す連絡機関の属性，そのデータのドキュメントを書いた著者名を表す著者名の属性，その著者の所属機関名を表す著者所属機関の属性，そのデータセット自体を作成した人の名前を表す作成者名の属性，その作成者の所属機関名を表す作成機関の属性に着目して各手法を実施した．

表 2: IDER 手法のニューラルネット予測結果

属性名	真陽性	真陰性	偽陽性	偽陰性	精度
データ ID	0	90525	0	0	100%
タイトル	0	90523	0	2	99.9%
連絡先	5558	84356	557	54	99.3%
連絡機関	8036	81166	625	698	98.5%
著者名	5122	84915	406	82	99.5%
著者所属機関	7445	81695	922	463	98.5%
作成者名	5539	84578	372	36	99.5%
作成機関	8041	80920	738	826	98.3%

入力として与える学習済みの単語ベクトル集合として，GloVe を用いて Common Crawl のウェブアーカイブから学習された学習済みの単語ベクトル集合 [2] を使用した．また，HoloClean は一貫性制約を与えられる必要があるため，既存の一貫性制約発見手法により発見したサポート 2 以上の Conditional Functional Dependencies を与えた [8,9]．また，HoloClean はデータの全てを小文字化して扱うため，それにならい，与える関係データは全て小文字とし，値の前後の不要な空白の除去を行った．

バリデーションデータの比率を 20 %に，学習率を 0.01 に設定し，セル数が 500 で最終的に二値分類を行う 2 層のニューラルネットを用いて 20 エポックの学習を行った．タプルの全てのペアに対するポジティブとネガティブの二値分類を学習し，各属性の値を対象にして，値が一致する場合はポジティブ，そうでなければネガティブに分類するように学習を行う．各値のペアごとに出力される 0 以上 1 以下の値が閾値より大きいか小さいかでポジティブかネガティブかの判断が行われる．なお，本実験では訓練データの少なさから予測結果にばらつきが存在しており，それを抑えるためにニューラルネットの出力値の 2 0 回の平均値をとり，それをニューラルネットの出力値として扱うこととした．

各属性ごとに 90525 件存在する全ての値のペアを対象にポジティブかネガティブの予測を行う．人間が結果を確認することを想定しているため，予備実験を行って閾値を検討し，今回は件数が大きくなりすぎないように二値分類の閾値を 0.4 と設定した．そのため，ニューラルネットが 0.4 以上を出力した値のペアはポジティブに分類され，ニューラルネットが 0.4 未満を出力した値のペアはネガティブに分類されている．各属性に関する予測結果を表 2 に示す．表 2 の列はそれぞれ手法を適用した属性名を表す属性名，ニューラルネットにポジティブと判断され実際に値が一致したペアの件数を表す真陽性，ニューラルネットにネガティブと判断され実際に値が一致しなかったペアの件数を表す真陰性，ニューラルネットにポジティブと判断されたが実際には値が一致しなかったペアの件数を表す偽陽性，ニューラルネットにネガティブと判断されたが実際には値が一致していたペアの件数を表す偽陰性, 全ての値のペアに対するニューラルネットの判断と実際の値の一致不一致が同じだった

1 : https://www.diasjp.net

2 : https://github.com/stanfordnlp/GloVe

ペアの割合である精度を表している．表 2 に示したように全ての属性で高い精度で予測が行えていることが分かった．

この手法では，このうちの偽陽性に着目する．ニューラルネットにポジティブと判断されたが実際には値が一致しなかったペアを不整合候補として人間に提示する．偽陽性に分類された値のペアは 3620 件であるが，同じ値の組み合わせをしたペアが繰り返し登場しており，人間が目を通しづらいため，同じ値の組み合わせをしたペアは一つに纏めることとした．しかし，空値と値のペアは同じ値の組み合わせであっても，空値に入っているべき真の値が異なる可能性があるため，空値はそれぞれが別々の異なる値であるとして処理をした．そのため，閾値 0.4 の際には，778 通りの値のペアを人間に提示することになる．また，同じデータに対して既存の制約を利用したデータクリーニング手法である HoloClean [4] を適用した結果，30 箇所のセルをエラーとみなして修正が行われた．今回の実験では，修正される前の値と修正後の値のペアを不整合候補として扱うこととした．そのいずれもが空値を埋める形での修正であったため，上述の判断を適用した結果，30 通りの値のペアを人間に提示することになる．

これらの 2 つの手法で得られた値のペアをランダムに並べ替えてどの手法で検出されたか伏せた上で，その値がどの属性の値であるかについてや，手法が対象とした関係データ内にその値が出現した頻度，その値が属しているタプルに対応しているオリジナルのメタデータの URL，その値がその属性で出現している全てのタプルの ID，その値がその属性で出現しているタプルのうちランダムに選んだ代表のタプル一つの全ての値を参考情報として追加して，人間に提示した．それらの情報を参考に人間が，各値のペアに対してそのペアが不整合候補かそうでないかについて判断する実験を行った．二名の評価者による評価を行い，少なくともその一方が不整合候補であると判断した場合，その値のペアは不整合候補であるとして扱うこととした．その結果，IDER 手法が検出した 778 件の値のペアの中で 113 件の値のペアが不整合候補であると評価され，そのうち空値の穴埋めを提案している値のペアを除くと 72 件の値のペアが不整合候補であると評価された．HoloClean は検出した 30 件の値のペアのうち 29 件の値のペアが不整合候補であると評価され，その全てが空値の穴埋めを提案する値のペアだった．この結果から既存の制約を利用した手法は今回実験で用いたような制約脆弱データでは，単純な誤字脱字を対象にしたエラー検出手法としても上手く機能しないことに対して，IDER 手法は，空値の穴埋めを提案する以外にも不整合な値を検出できていることが分かった．

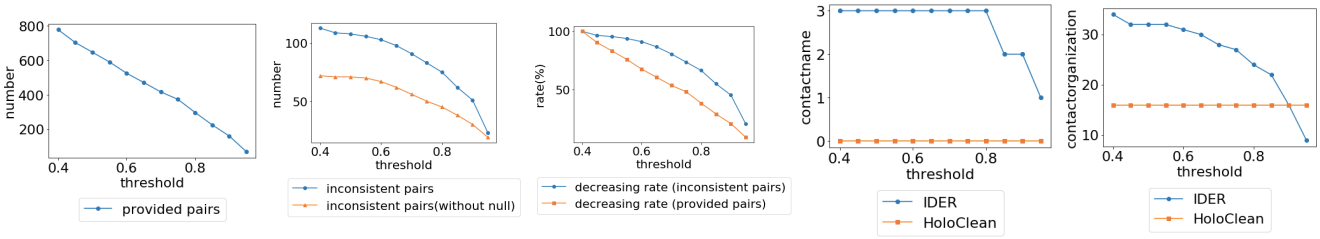
実際に IDER 手法によって発見できた不整合候補の例を以下に示す．

- “JAMSTEC” と “Japan Agency for Marine-Earth Science and Technology”
- “Dr. Shuhei Masuda” と “Masuda, Shuhei, Dr.”
- “Japan Aerospace Expolation Agency G-Portal support desk” と “G-Portal Support Desk”
- “Japan Aerospace Expolation Agency G-Portal support desk” と “Japan Aerospace Exploration Agency GCOM-W1 Data Providing Service Help Desk”
- “Dairaku Koji” と “Diraku Koji”

このように誤字脱字や語順の違いから，一見同じ組織名を指しているとは分からない略称や表記の違いについても発見することができている．

また，今回は IDER 手法のポジティブとネガティブの閾値を 0.4 に設定して結果を観察したが，閾値が低いほどポジティブと誤分類された偽陽性の値のペアも増加すると考えられる．そのため，本研究では，閾値が低いほど IDER 手法の再現率が高くなるような傾向が生じると思われる．そこで，閾値の上昇に伴って，IDER 手法が出力する値のペアの総数や，そのうちの人間が見つけた不整合候補の数がどのように変化するかについても調査を行った．IDER 手法がある値のペアに対して出力した，二値分類の判断に用いる 1 から 0 を取る値のうち，最大の値が閾値より下であった場合，その値のペアは手法の結果として出力されていないとみなすことによって，閾値の変化に伴う結果の変化を観察した．その結果を図 3a，3b，3c に示す．図 3a は，各閾値ごとに IDER 手法が人間に提示することになる値のペアの数を表している．横軸が閾値を表しており，縦軸が値のペアの数を表している．図 3b は，各閾値ごとに IDER 手法が人間に提示したペアの中で実際に不整合候補だと判断された値のペアの数を表している．横軸が閾値を表しており，縦軸が人間が不整合候補だと判断した値のペアの数を表している．青い線が不整合と判断された全ての値のペアの数を表しており，オレンジの線が，そのうち空値の穴埋めを提案する値のペアを除いた，それ以外の値のペアの数を表している．この結果から，事前の想定通り，閾値が低いほど，不整合候補として判断される値のペアの数も減少することが分かった．また，図 3c は，各閾値ごとに，IDER 手法が人間に提示する値のペアと，不整合と判断された値のペアについて，閾値の上昇ごとにどれくらい減少していくのかについてを図示したものである．横軸が閾値を表しており，縦軸が閾値が 0.4 の場合と比較して数が何%に減少しているかを表したものである．青い線が不整合と判断された値のペアについて，オレンジの線が IDER 手法が人間に提示する値のペアについて表している．図 3c によると，人間に提示された値のペアより，不整合候補と判断された値のペアの方が，閾値の上昇に伴う値の減少が小さくなることが分かった．そのため IDER 手法では，余計な値のペアになるべく目を通さずに不整合候補を発見したい場合には，閾値をより大きく設定することが望ましいことが分かる．

続いて，IDER 手法と HoloClean の結果の比較を，図 3d，3e に示す．結果の傾向は，連絡先，著者名，作成者名の人名を多く含む属性同士で類似しており，また，連絡機関，著者所属機関，作成機関の組織名を多く含む属性同士で類似している．そのため，人名を多く含む属性の代表として連絡先の属性を，組織名を多く含む属性の代表として連絡先機関の属性を対象に図示を行う．それぞれ，図 3d は連絡先，図 3e は連絡先機関について，着目属性として扱った際における IDER 手法と HoloClean の比較を行っている．データ ID とタイトルは，一



(a) IDER 手法が提示する値のペア数 **(b)** 不整合候補と評価された値のペア数 **(c)** 提示された値のペアと不整合候補の減少率 **(d)** 連絡先での IDER と HoloClean の比較 **(e)** 連絡先機関での IDER と HoloClean の比較

図 3: 閾値の上昇に伴う結果の変化

部の例外を除き一意な値を持つため，ニューラルネットは全てのペアをネガティブと判断しており，人間への値の提示は行われなかった．比較の対象は，各手法が人間に提示した値のペアの中で，人間が不整合候補だと評価した値のペアの数である．その値のペアの数で，各閾値ごとに IDER 手法と HoloClean の比較を行った．それぞれの図において，青い線が IDER 手法を示しており，オレンジの線が HoloClean を示している．横軸が閾値を表しており，縦軸が人間が不整合候補だと評価した値のペアの数を表している．

いずれの属性においても，IDER 手法は閾値が 0.85 以下の場合，IDER 手法が HoloClean により多くの不整合候補を検出できている．そのため，閾値を上昇させて人間に提示する値のペアの数を絞りたい場合でも，このデータを対象にするならば，閾値は 0.85 以下に設定するのが望ましい．また，連絡先，著者名，作成者名の人名を多く含む属性は，連絡機関，著者所属機関，作成機関の組織名を多く含む属性と比較して，閾値の上昇に伴う不整合候補と判断された値のペア数の減少するペースが抑えられている．本研究では，人名は不整合かどうかを判定しやすい簡単な値が多いことに対して，組織名は不整合かどうかを判定することが難しい値が多いため，組織名を多く含む属性では，人名を多く含む属性よりニューラルネットの確信度が下がり，出力する予測値が 0.5 に近づいていることがこのような傾向を持つ原因だと考えた．

4　エラー検出手法としての評価

IDER 手法は，通常の制約よりも更に柔軟で緩和された，適合率より再現率に比重をおいた制約を用いて不整合候補を検出することを目的としている．実際に，再現率が高くなるかわりに適合率が低くなるような手法になっているかを考察するための実験の一環として，IDER 手法を実際のデータに対してエラー検出手法として適用して，既存の制約を利用した手法である HoloClean [4] の結果と比較を行った．

本実験では，HoloClean の実装例に付属している Hospital データセットを使用した [3]．Hospital データセットは，データクリーニングに有用な一貫性制約を多く含むデータクリーニング用のテストデータであり，本実験で利用したデータは重複を除去した上で文字列のうち一つを他の文字に置き換えるような人工的なノイズを混入させたデータとなっている．このデータを空値でのみ構成される 2 属性を除いた 17 属性の 1000 行の関係データとして使用した．このデータにはノイズが混入する前のクリーンなデータも付属しており，そちらを正解のデータとした．また，HoloClean はデータ以外にデータクリーニングに有用な一貫性制約を必要とするため，HoloClean の実装例に付属している 15 のデータクリーニングに有用な一貫性制約を与えた．この実験ではエラー検出手法としての性能を評価するため，汚い値を含んでいるセルの検出数で性能を評価することとした．HoloClean は値が修正されたセルを検出できたセルとみなし，IDER 手法は，不整合候補として出力された値のペアの両方のセルを検出できたセルとみなした．

表 3: エラー検出手法としてのセルごとの評価

手法名	再現率	適合率
IDER 手法	63%	34%
HoloClean	49%	100%

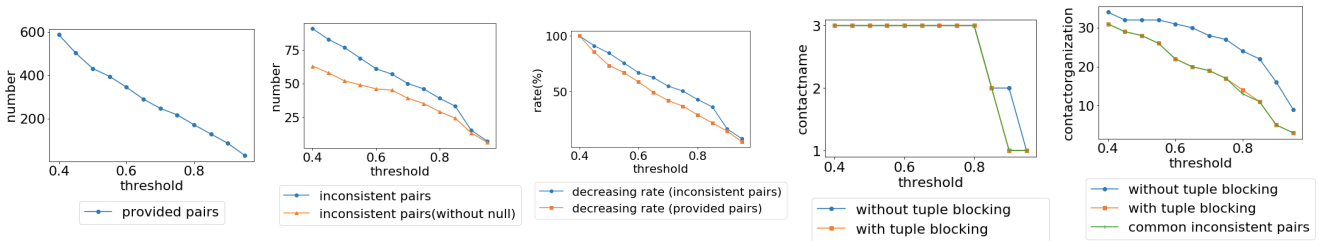
表 4: エラー検出手法としての値の種類ごとの評価

手法名	再現率	適合率
IDER 手法	74%	54%
HoloClean	50%	100%

入力として与えるノイズの混入したデータと正解のデータとで差異のある汚い値を含んでいるセルは 509 件存在している．HoloClean がノイズの候補として出力したセルは 248 件であり，IDER 手法がノイズの候補として出力したセルは 960 件だった．汚い値を含んでいるセル 509 件のうち，HoloClean が正しく検出できたセルの数は 248 件であり，IDER 手法が正しく検出できたセルは 323 件であったことから，それぞれの手法の適合率と再現率は表 3 のようになる．

また，エラーをセルごとに数えるのではなく，値の種類ごとに数えた場合，エラーの値の種類は全てで 394 件となる．そのうち，HoloClean がノイズの候補として出力したエラーの値の種類は 198 件となり，IDER 手法がノイズの候補として出力したエラーの値の種類は 539 件となる．全てのエラーの値の種類 394 件のうち，HoloClean が正しく検出できたエラーの値の種類は 198 件であり，IDER 手法が正しく検出できたエラーの値の種類は 341 件であったことから，それぞれの手法の適合率と再現率は表 4 のようになる．

3 : https://github.com/HoloClean/holoclean



(a) IDER 手法が提示する値のペア数 **(b)** 不整合候補と評価された値のペア数 **(c)** 提示された値のペアと不整合候補の減少率 **(d)** 連絡先での結果の比較 **(e)** 連絡先機関での結果の比較

図 4: タプルブロッキングを適用した場合の閾値の上昇に伴う結果の変化

表 5: タプルブロッキングを適用した場合の削減率とニューラルネット予測結果 (K=9, L=7)

属性名	削減率	真陽性	真陰性	偽陽性	偽陰性	精度
データ ID	42.0%	0	90525	0	0	100%
タイトル	19.9%	0	90523	0	2	99.9%
連絡先	23.7%	5557	84572	341	55	99.6%
連絡機関	23.8%	7917	81241	550	817	98.5%
著者名	30.7%	5123	84970	351	81	99.5%
著者所属機関	6.1%	7481	81340	1277	427	98.1%
作成者名	13.2%	5544	84674	276	31	99.7%
作成機関	19.3%	7915	81509	149	952	98.8%

以上のことから，IDER 手法はデータクリーニングに有用な一貫性制約を多く含むデータに対して，再現率を重視したエラー検出手法として適用することが可能であることが分かった．中でも特に，エラーをセルごとではなく値の種類ごとに検出する場合は，適合率と再現率が大きく向上し，より性能の良いエラー検出手法として適用することが可能であることが分かった．

5 タプルブロッキングの適用

第 3 節で利用した DIAS の科学メタデータと同じデータに対して，タプルブロッキングを利用し訓練時の計算コストを低減した状態で，提案手法である IDER 手法を実施し結果を観察した．タプルブロッキングではブロック数に対応しているハッシュコードの桁数 K と，そのハッシュコードの割り当てを行う回数 L を定める必要がある．予備実験を実施した結果，ニューラルネットの予測の精度を大きく下げるような極端な削減が起きず，安定した精度の予測結果が得られる $K = 9$， $L = 7$ でタプルブロッキングを適用した．また，第 3 節と同様に，20 回の予測結果に対して平均を取ることで値のばらつきを抑えることとした．

$K = 9$，$L = 7$ で各属性にタプルブロッキングを適用した結果のタプルのペアの削減率と，その場合のニューラルネットの予測結果を表 5 に示す．$K = 9$，$L = 7$ でタプルブロッキングを行った場合，94%や 95%等に精度が大きく落ちる属性はなく，98%以上となることが分かった．しかし，本研究が提案する IDER 手法は，対象とする属性における不整合候補の発見が目的であり，ニューラルネットの予測結果の精度以上に，予測結果の偽陽性の中身に大きく左右されるため，この予測結果だけを見て，タプルブロッキングが有効かどうかは判断できない．

そこで再び，第 3 節と同様の条件で，タプルブロッキングを適用した場合にどれくらい不整合候補が検出できているかを確認した．第 3 節と同様に，同じ値の組み合わせをしたペアは一つにまとめ，空値のみそれぞれが別々の異なる値であるとして処理をした．その結果，閾値 0.4 の際には，588 通りの値のペアを人間に提示することになる．第 3 節と同様の条件で，二名の評価者により評価を行った．

評価の結果，タプルブロッキングを適用した場合の IDER 手法が検出した 588 件の値のペアの中で，91 件の値のペアが不整合であると評価され，そのうち空値の穴埋めを提案している値のペアを除くと 63 件の値のペアが不整合候補であると評価された．第 3 節のタプルブロッキングを適用しなかった場合の IDER 手法の結果は，778 件のペアの中で 113 件の値のペアが不整合候補であった．そのため，閾値を 0.4 に設定した時，タプルブロッキングを使用した場合の不整合候補を検出する能力は，タプルブロッキングを使用しない場合と比較して，劣る結果となった．

続いて，第 3 節と同様に，タプルブロッキングを適用した場合においても，閾値の上昇に伴って，IDER 手法が出力する値のペアの総数や，そのうちの人間が見つけた不整合候補の数がどのように変化するかについて調査を行った．各閾値ごとの，IDER 手法が人間に提示した値のペアの数と，そのうち，いくつの値のペアが実際に不整合候補であったかについて，図 4a, 4b, 4c に示す．図 4a は，タプルブロッキングを適用した場合の，各閾値ごとに IDER 手法が人間に提示することになる値のペアの数を表している．図 4b は，タプルブロッキングを適用した場合の，各閾値ごとに不整合候補と評価された値のペアの数を表している．これらの図が第 3 節の結果と大きく異なる点として，閾値を 0.9 や 0.95 などの大きな値とした場合，不整合候補と評価される値のペアの数が大きく減少している点が挙げられる．このことから，タプルブロッキングを適用しない場合と比較して，タプルブロッキングを適用する場合では，低めの閾値を与える必要があることが分かった．また，図 4c は，各閾値ごとに，IDER 手法が人間に提示する値のペアと，不整合と評価された値のペアについて，閾値の上昇ごとにどれくらい減少していくのかについてを図示したものである．横軸が閾値を表しており，縦軸が閾値が 0.4 の場合と比較して数が何%に減少しているかを表したものである．青い線が不整合と評価さ

れた値のペアについて，オレンジの線が IDER 手法が人間に提示する値のペアについて表している．この図に関しても第 3 節と異なっており，不整合と評価された値のペアの減少率と人間に提示する値のペアの減少率の差が小さくなっている．以上のような差がでる理由としては，タプルブロッキングによって訓練に利用するタプルのペアを削減することにより，学習に影響が少ないペアだけでなく，学習に有益なタプルのペアも削減されてしまったため，ニューラルネットの予測の確信度が下がり，0.5 に近い値を出力するようになっていることが考えられる．

続いて，タプルブロッキングを適用しなかった場合と適用した場合の IDER 手法の結果の比較を，図 4d，4e に示す．第 3 節と同様に，人名を多く含む属性の代表として連絡先の属性を，組織名を多く含む属性の代表として連絡先機関の属性を図示する．それぞれ，図 4d は連絡先，図 4e は連絡先機関について，タプルブロッキングを適用しなかった場合の IDER 手法と，タプルブロッキングを適用した場合の IDER 手法の比較を行っている．比較の対象は，各手法が人間に提示した値のペアの中で，人間が不整合候補だと評価した値のペアの数である．その値のペアの数で，各閾値ごとにタプルブロッキングを適用しなかった場合の IDER 手法と，タプルブロッキングを適用した場合の IDER 手法との比較を行った．それぞれの図において，青い線がタプルブロッキングを適用しなかった場合の IDER 手法を，オレンジの線がタプルブロッキングを適用した場合の IDER 手法を示している．また，緑の線は両者に共通する値のペアの数を示している．これらの図から，基本的にタプルブロッキングを適用した場合では，タプルブロッキングを適用しなかった場合と比較して，不整合候補だと評価される値のペアの数が劣ることが分かる．

以上のことから，タプルブロッキングを適用した場合，ニューラルネットの予測結果の精度が大きく変化しなくても，IDER 手法が着目している偽陽性となる値のペアの数は敏感に変化することから，タプルブロッキング手法は IDER 手法に対して適していないと考えた．

また，タプルブロッキングの参考にした論文 [6] では，タプルの全ての属性の値から，重複したタプルを発見することを目的としており，重複したタプルの各値は，類似度が非常に高くなると推測される．それに対して，IDER 手法は対象属性において，エンティティが同一と思われる値のペアを発見することを目的としている．仮に対象属性において同一な値を持つタプルのペアであったとしても，他の属性の値は重複したタプルの値ほど類似していない場合が多いと推測される．そのような場合のポジティブにラベル付けされるタプルのペアの類似度の低さが，タプルブロッキングを適用する際に行われるランダム超平面手法のランダム性に大きく影響され，実行結果の精度やタプルのペアの削減率を不安定にしているのではないかと考えた．

6 おわりに

本研究は単純な誤字脱字のようなエラーではなく，同じエンティティを指しているが値の表記が異なるような，いわゆる表記ゆれなどの不整合な値に対して，辞書等の外部情報を含めたデータクリーニングに有用な一貫性制約がない制約脆弱データにおいては，既存のデータクリーニング手法の適用が困難であると考えた．そこで，既存の制約を利用するデータクリーニング手法と IDER 手法を実際の制約脆弱データに適用し，再現率の観点で IDER 手法が優れていることを確認した．また，制約脆弱でないデータに対して，IDER 手法をエラー検出手法として適用した場合でも，再現率の観点で，既存の制約を利用するデータクリーニング手法より IDER 手法が優れていることを確認した．さらに，タプルブロッキング手法を用いて IDER 手法の計算コストを低減することを検討し，タプルブロッキングを適用した場合の IDER 手法適用結果を，タプルブロッキングを適用していない場合の IDER 手法適用結果と比較した．

今後の課題としては，他の制約脆弱データでも同様の結果が得られるか確認することや，低い適合率を向上させるように手法を改善することに加え，検出された不整合を基に，実際に人間がチェックして値を修正できるようなデータクリーニング方式を実装することなどが挙げられる．

謝　辞

本研究の一部は JSPS 科研費 JP17H06099, JP18H04093, JP18K11315 の助成を受けたものです．

文　献

[1] Sean Kandel, Jeffrey Heer, Catherine Plaisant, Jessie Kennedy, Frank van Ham, Nathalie Henry Riche, Chris Weaver, Bongshin Lee, Dominique Brodbeck, and Paolo Buono. Research directions in data wrangling: Visualizations and transformations for usable and credible data. *Information Visualization*, 10(4):271–288, 2011.

[2] Tamraparni Dasu and Theodore Johnson. *Exploratory data mining and data cleaning*, volume 479. John Wiley & Sons, 2003.

[3] 大森 弘樹， 清水 敏之， 吉川 正俊. エンティティ解決手法を応用したデータクリーニングのための不整合検出. *DEIM2020*, 2020.

[4] Theodoros Rekatsinas, Xu Chu, Ihab F Ilyas, and Christopher Ré. Holoclean: Holistic data repairs with probabilistic inference. *Proceedings of the VLDB Endowment*, 10(11):1190–1201, 2017.

[5] Alireza Heidari, Joshua McGrath, Ihab F. Ilyas, and Theodoros Rekatsinas. Holodetect: Few-shot learning for error detection. In *Proceedings of the 2019 ACM SIGMOD*, page 829846, 2019.

[6] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment*, 11(11):1454–1467, 2018.

[7] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 EMNLP*, pages 1532–1543, 2014.

[8] Philip Bohannon, Wenfei Fan, Floris Geerts, Xibei Jia, and Anastasios Kementsietsidis. Conditional functional dependencies for data cleaning. In *Proceedings of the 23rd ICDE*, pages 746–755, 2007.

[9] Wenfei Fan, Floris Geerts, Jianzhong Li, and Ming Xiong. Discovering conditional functional dependencies. *IEEE TKDE*, 23(5):683–698, 2011.