日本データベース学会 データドリブンスタディーズ
Vol. 1, Article No. 4, 2023 年 3 月

データの量と性質を変化させた際のエンティティマッチングモデルの評価

山岡 大輝 [1]　林 勝悟 [2]　草野 元紀 [3]
岡留 剛 [4]

エンティティマッチングとは異なるデータ集合間で同一のエンティティを参照するレコードペアを識別するタスクのことである．一般的に機械学習によるアプローチでは Levenshtein 類似度などの類似度関数によりレコードペアの類似度ベクトルを用いて二値分類器を学習する手法や深層学習によるアプローチがある．特に BERT など事前学習済み言語モデルを用いた手法はそれ以前の手法を大きく凌駕する分類精度を達成しており盛んに研究が行われている．しかし，類似度関数を用いた手法より深層学習による手法がどのデータにおいても最も有効かは定かではなくデータの量や性質によってどの手法が適しているかは異なることが考えられる．本研究ではこれを複数の代表的なエンティティマッチングモデルをデータの量と性質から評価することで手法の利点と欠点を明らかにした．また，事前学習済み言語モデルを用いた手法がマッチングの予測を行うためにレコードペアのどの単語に着目しているか分析し改善点を考察する．

1　はじめに

あらゆる企業や組織は日々大量のデータを扱い，生み出している一方で，これらのデータは組織内でさえも統一された形式で管理されているとは限らず，さらに異なる組織間で管理されるデータは同じエンティティを指すものであっても表記上は全く異なる場合がある．この問題はデータ管理の非効率性やデータ分析の弊害に繋がり，このようなデータの統合を行うエンティティマッチングは重要なタスクである．例えば自社の商品データと外部の商品データが得られている場合にこれらのデータ集合間で同一の商品を特定することが可能となれば，その同一商品データを紐づけることで自社の持つデータをより情報量の多いデータへと拡張することができ，分析やアプリケーション開発がより行いやすくなる．しかし，一般的には異なるデータ集合間の同一のエンティティを参照するデータのペアは表記揺れにより商品名が異なっている場合があり，このようなペアを同一とみなすことが困難である．機械学習を用いたエンティティマッチングはレコードペアを入力としてこれらが同一のエンティティを参照するか否かを識別する二値分類問題を解くこととして定式化され，Levenshtein 類似度などの文字列の類似度関数を二値分類器の特徴量として学習を行う手法 [1, 5] や深層学習を用いた手法など様々なエンティティマッチング手法が提案されてきた [3, 6, 9]．特に最近では BERT など事前学習済み言語モデルを用いた手法が著しい成果をあげており注目されているが，学習データの量や性質によってどの手法が優れているのか自明ではなくこれまでに調査を行った研究はない．そこで本研究では代表的なエンティティマッチング手法をデータの量や性質など条件を変えて評価することで各手法の利点と欠点を考察する．また，検証用データの予測結果を分析することで特に顕著な成果を上げている事前学習済み言語モデルを用いた手法の改善点を考察する．

2　関連研究

エンティティマッチングとは，レコードの集合からなるデータセット $L = \{l_1, \ldots, l_{|L|}\}$ と $R = \{r_1, \ldots, r_{|R|}\}$ が与えられたとき，直積集合 $L \times R$ のレコードペアが同一のエンティティを参照するか否かで分類する問題である．

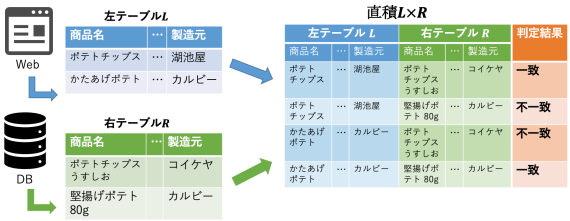


図 1: エンティティマッチングの概念図．直積集合 $L \times R$ の要素であるレコードペアのエンティティが一致するか否かを予測する．

教師あり二値分類問題としてのエンティティマッチングは，レコードペアの文字列の類似性に基づく特徴量を入力としランダムフォレストなどの二値分類器の学習を行うアプローチや，レコードペアを Glove [10] や fastText [2] による単語埋め込みや BERT [4] による文脈化単語埋め込みを利用して深層学習による二値分類を行うアプローチが一般的である．Blienko ら [1] は適応的な文字列の類似性に基づく手法を提案しており，学習可能な文字列の拡張編集距離による類似度と，サポートベクトルマシンによる学習可能な類似度ベクトルを利用した．Konda ら [5] は複数の文字列の類似度関数によって得られる特徴ベクトルを入力とする分類モデルを用いてエンティティマッチングを行う一連のシステムを提案している．文字列の表記に基づく類似度指標である Levenshtein 類似度，Jaro-Winkler 類似度，Jaccard 係数などはエンティティマッチングによく用いられ誤字脱字にも有用である一方で，略称や別称といった表記上に現れない類似性を測ることは困難である．Mudgal ら [9] は深層学習をエンティティマッチングに適用するためのフレームワークを提案し，Glove や fastText による単語埋め込みベクトルを用いたモデル，双方向 RNN，Attention を利用したモデル，これらを組み合わせた

[1] 学生会員　関西学院大学 大学院 理工学研究科
izu14587@kwansei.ac.jp
[2] 正会員　日本電気株式会社 データサイエンス研究所
shogo-hayashi@nec.com
[3] 正会員　日本電気株式会社 データサイエンス研究所
g-kusano@nec.com
[4] 正会員　関西学院大学 大学院 理工学研究科
tokadome@acm.org

Hybrid モデルを提案し比較検討を行ない，深層学習アプローチの利点を示した．Brunner ら [3] は Transformer ベースの事前学習済み言語モデルである DistilBERT [11]，BERT，XLNet [12]，RoBERTa [7] をエンティティマッチングタスクで評価を行い既存の深層学習ベースの手法を大幅に上回る結果を報告した．また，Li ら [6] は RoBERTa をベースとしてデータ拡張やドメイン知識の追加を入力データに行う DITTO を提案し既存手法を上回る分類精度を達成している．

3 比較手法

本研究では複数の代表的なエンティティマッチング手法をデータの量と性質を変化させた際の評価実験を行い各手法を比較する．本節では評価実験に用いる各手法の概略を説明する．表 1 に各手法の比較の表を示す．

3.1 Magellan

Magellan [5] は文字列の類似度関数を用いたエンティティマッチング手法であり，レコードペアの同じ属性の属性値同士を複数の類似度関数を用いて類似度ベクトルを計算し二値分類モデルの特徴量とする．使用されている類似度には Levenshtein 類似度や Jaro–Winkler 類似度があり，これらの類似度関数から得られた特徴量を用いて複数の二値分類器で学習を行い検証用データの評価値が最も高い分類器が予測に採用される．用意されている分類器は線形回帰，ロジスティック回帰，決定木，サポートベクトルマシン，ランダムフォレストがあり，本実験でもこれらを用いている．

3.2 DeepMatcher

DeepMatcher [9] は図 2 に示すようにエンティティマッチングに深層学習を適用するためフレームワークを提案した手法である．レコードペアの同一属性ごとに属性値のペアを fastText などの学習済みの単語埋め込みモデルで埋め込みベクトルに変換し (1.Attribute Embedding)，埋め込みベクトルのペアを属性ごとに用意される RNN などのディープニューラルネットワークによって得られる類似度ベクトル (2.Attribute Similarity Representation) を全結合層の入力とすることで二値分類を行う (3.Classification)．DeepMatcher では類似度ベクトルを獲得する機構を複数提案しているが本実験では論文内で最も分類精度が良いと報告されている Hybrid モデルを使用する．

3.3 DITTO

BERT が登場して以降，事前学習済み言語モデルは自然言語処理の様々なタスクで高い分類精度を達成することが報告されており，エンティティマッチングにおいても言語モデルを用いた手法がそれ以前の手法を上回る成果を報告されている．特に DITTO [6] はエンティティマッチングにおいて最も分類精度が高い手法の 1 つであり，言語モデルに RoBERTa [7] を用いている．言語モデル以前の手法ではレコードペアの同一属性間の比較を陽に行って類似度ベクトルを得る必要があったのに対し，言語モデルを用いた手法ではレコードペアを単純に繋げてシリアライズ処理を行なったシーケンスを入力とする特徴がある．DITTO では特に入力データに加工を施しており，レコードペアをシリアライズする際に属性名と属性値を区別するトークンを挿入，シーケンスの中でマッチングに有用であると思われる単語の前後にトークンを挿入するドメイン知識の追加，学習データにデータ拡張を適用する工夫をおこなっている．

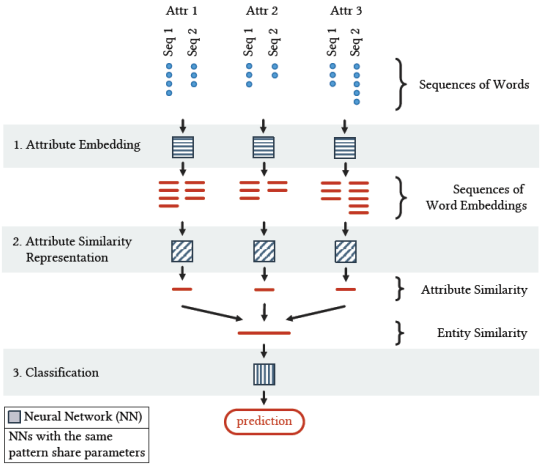


図 2: DeepMatcher で提案されたアーキテクチャ，Deep Learning for Entity Matching: A Design Space Exploration [9] より引用

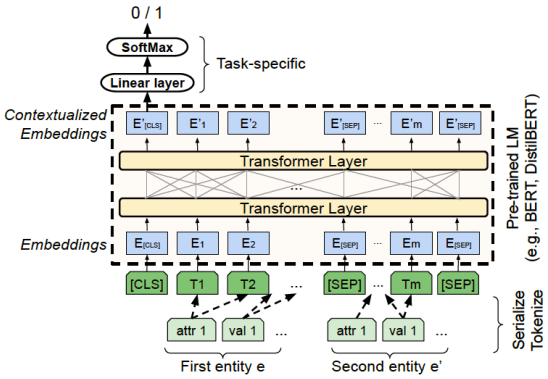


図 3: DITTO のモデルアーキテクチャ，Deep entity matching with pre-trained language models [6] より引用

3.4 RoBERTa

Brunner ら [3] は事前学習済み言語モデル（DistilBERT [11]，BERT，XLNet [12]，RoBERTa [7]）をエンティティマッチングに適用できるように，レコードペアの属性値をシリアライズする手法を提案している．本実験では RoBERTa を比較手法として採用し，以後この手法を RoBERTa と称する．

表 1: 手法の比較

	深層学習	古典的な機械学習
左右のテーブルの属性名集合が一致する必要性あり	DeepMatcher	Magellan
左右のテーブルの属性名集合が不一致でも可能	DITTO，RoBERTa	

4 実験

本実験では特に商品データにおいて実世界で起こりうるデータの性質について各手法の評価を行うため，データの量 (4.3 節)，欠損値の割合 (4.4 節)，レコード間の属性の不一致度 (4.5 節) を変化させた際の各エンティティマッチング手法の評価を行う．

4.1 評価指標

一般的なエンティティマッチングのデータは負例の数が正例の数より大幅に多い不均衡データであるため，評価指標には精度 (accuracy) ではなく適合率 (precision) と再現率 (recall) の両方を考慮する $F_1 \in [0, 1]$（式 (1)）を用い，以下の式で計算される．

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

4.2 使用するデータセット

本実験では表 2 に示す ER-Magellan データセット[*1]の商品に関するデータを使用する．これらのデータセットは各 EC サイト間の商品レコードをペアとしてそのペアが同一のエンティティを参照するか否かのラベルが与えられている．

表 2: ER-Magellan データセット

データセット	レコードペア数	正例の数	属性
Walmart-Amazon	10242	962	title,category, brand,modelno, price
Amazon-Google	11460	1167	title,price, manufacturer
Abt-Buy	9575	1028	name,price, description

4.3 学習データ数に関する評価

4.3.1 実験設定

学習データ，検証用データ，テストデータのそれぞれの割合は 3:1:1 となっており，Walmart-Amazon，Amazon-Google，Abt-Buy の各データセットは学習データ数が 6144，6874，5743 である．データ数に関する実験では学習データ数を 100，500，1000，3000，全ての 5 パターンで各手法の学習を行い評価を行なう．

4.3.2 実験結果

図 4 は横軸を学習データ数，縦軸をテストデータの F_1 とする各手法の実験結果である．各データセットにおいて 500 から 3000 以上の数の学習データを使用して学習を行った場合，DITTO と RoBERTa が Magellan と DeepMatcher を上回る F_1 を達成しており，商品の説明文が属性値となる属性 description が存在する Abt-Buy データセットではその差が顕著であることが分かる．しかし，Amazon-Google と Abt-Buy データセットでは 100 と学習データ数がごく少数である場合，Walmart-Amazon データセットでは 100 から 1000 程度の学習データ数において Magellan が他の手法を上回る F_1 を達成している．また，図 4 から DeepMatcher が Magellan より高い F_1 を達成するには 3000 程度の学習データ数が必要である．したがって，DeepMatcher と同じ深層学習ベースの手法であり，事前学習済み言語モデルを利用する DITTO と RoBERTa はデータ数が少ない場合においても DeepMatcher より優れた分類性能を持つことが分かった．

[*1] https://github.com/anhaidgroup/deepmatcher/blob/master/Datasets.md

4.4 欠損値の割合に関する評価

4.4.1 実験設定

実際に得られる商品データテーブルは記入漏れなど様々な理由で属性値が欠損している場合がある．本実験では Walmart-Amazon データセットに対して属性値を一定の確率で欠損させたデータを作成し，各手法の学習及び評価を行う．ただし，title 属性は商品名が格納される属性であり欠損するとは考えにくいため，title 属性を除く属性値を欠損の対象とする．欠損する確率は 0，0.1，0.3，0.5 で変化させ評価を行う．

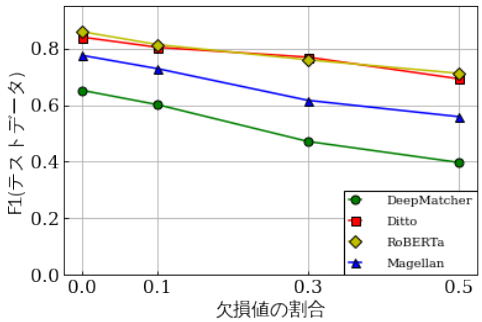


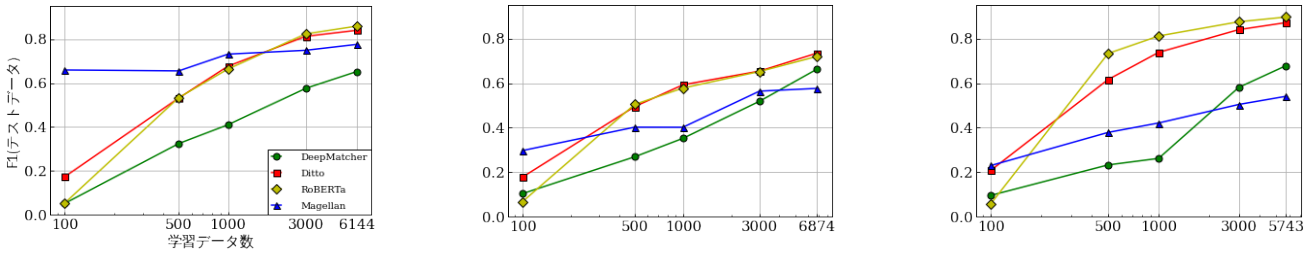
図 5: Walmart-Amazon データセットの欠損値の割合を変化させた際の各手法の F_1（random seed を 10 回変えて学習を行なった平均値）

欠損確率	RoBERTa	DITTO	DeepMatcher	Magellan
0	0.859	0.840	0.652	0.775
0.1	0.838 (-0.021)	0.804 (-0.036)	0.601 (-0.051)	0.728 (-0.047)
0.3	0.779 (-0.080)	0.769 (-0.071)	0.471 (-0.181)	0.616 (-0.159)
0.5	0.723 (-0.136)	0.693 (-0.147)	0.397 (-0.255)	0.559 (-0.216)

表 3: 欠損の割合に対する各手法の F_1 の値．カッコ内は欠損確率が 0 である時の F_1 の値からの減少量である．

4.4.2 実験結果

結果を図 5 に示す．欠損値の割合が増えるにつれて各手法の F_1 が減少しており，また，表 3 から DeepMatcher と Magellan が DITTO や RoBERTa と比べて F_1 の減少値の絶対値が大きいことが分かる．この違いは欠損値の扱いが異なることが一つの要因と考えられる．Magellan と DeepMatcher は左右のテーブルの同一属性間の属性値から算出される類似度からマッチングの学習を行うため，属性値が欠損している場合は欠損値の補間を行う必要がある．本実験での欠損値補間に関して，Magellan では欠損している属性値間の類似度を平均値で補間しており，DeepMatcher では属性値のベクトル化に fastText を使用しているため欠損している属性値を unknown トークンのベクトルに置換することにより欠損値の補完を行っている．これにより，これら 2 つの手法は欠損値を特定の値で補間しているため推定されたパラメータにはバ



(a) Walmart-Amazon (b) Amazon-Google (c) Abt-Buy

図 4: 学習データ数を変化させた際の各手法の F_1.（random seed を 10 回変えて学習を行なった平均値）

イアスが生じる．しかし，DITTO と RoBERTa は入力となるレコードペアをシリアライズしていることにより，属性値が欠損していたとしても補間の必要なく学習可能であり，補間によるバイアスは生じない．以上の理由から，レコードペアのシリアライズにより欠損値の補間が必要ない DITTO と RoBERTa がより欠損値の増加に対して頑健であると考えられる．

4.5 属性の一致数に関する評価

4.5.1 実験設定

Walmart-Amazon データセットなど一般的に用いられるデータセットは左右のテーブルで属性名が一致しており，対応する属性間の比較が容易である．しかし，現実的には左右のテーブルで異なる属性が存在する場合がほとんどである．そこで本実験では，Walmart-Amazon データセットのテーブル間の属性の一致数を変化させたデータを作成し各手法の学習と評価を行う．Walmart-Amazon は Walmart と Amazon の両方に title, category, brand, modelno, price が存在し一致する属性数は 5 であるため表 4 に示すように 3 つのパターンで評価を行う．Magellan と DeepMatcher は原則としてレコードペア間で対応する属性が必要となるため，本実験では全ての属性を一つの属性に結合することでテーブル間の不一致をなくす方法と，一致する属性のみを学習に使用する方法の 2 通りに対して評価を行なった．

属性を一つの属性に結合することを表 4 の「一致する属性数」が 1 の場合を用いて説明する．左テーブルでは title，brand, model no，右テーブルには title，category，price が存在し，それぞれのテーブルは 3 つの属性を持つが title 属性以外は対応しない属性である．そこで左右のテーブルにおいてこれら 3 つの属性を結合し，1 つの新しい属性 new-title を作成することで対応する属性として扱うことが可能となる．

表 4: 属性の一致数を変化させる

左テーブル	右テーブル	一致する属性数
title,category,brand, modelno,price	title,category,brand, modelno,price	5
title,brand, modelno,price	title,category, brand,price	3
title,brand, modelno	title,category, price	1

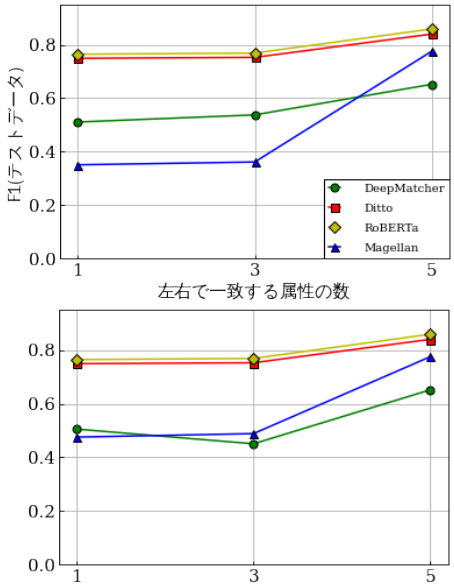


図 6: Walmart-Amazon データセットの左右の属性の一致数を変化させた際の各手法の F_1（random seed を 10 回変えて学習を行なった平均値）．上の図は DeepMatcher と Magellan に関して一つの属性に結合したデータで学習した結果，下の図は一致する属性のみ使用したデータで学習した結果を示している．

4.5.2 実験結果

結果は図 6 で示す通りであり，上の図は DeepMatcher と Magellan に関して一つの属性に結合したデータで学習した結果，下の図は一致する属性のみ使用したデータで学習した結果を示している．この結果から DITTO と RoBERTa は Magellan や DeepMatcher と比較して左右のテーブルで属性の一致する数が減少しても分類精度の減少が低いことが分かる．Magellan と DeepMatcher はレコードペア間で属性の対応が陽に与えられており同一属性の属性値間の比較が容易であるが，DITTO と RoBERTa は属性間の対応が与えられておらず学習時にデータから対応付けを学習する必要があると思われる．このため DITTO と RoBERTa はレコードペア間で対応しない属性の数が増えるほど学習が困難となり分類精度の低下が大きいと思われたが，本実験結果から学習による対応付けの獲得は困難ではないということが考えられる．

5 予測結果の分析

DITTO や RoBERTa など事前学習済み言語モデルをベースとしたエンティティマッチング手法はレコードペアをシーケンスにするためテーブル間で属性が対応付けられなくても学習にそのまま利用でき，データの量や質の変化に対しても他のモデル以上の分類精度を達成している．DITTO のモデル構造自体は RoBERTa と同一であるため，本節では RoBERTa の予測結果の分析を行う．ブラックボックスモデルの予測の説明を行う手法である SHAP [8] を利用して Walmart-Amazon データセットの検証用データに対する予測結果を分析することでレコードペアのどの単語が RoBERTa によるエンティティマッチング判定の予測に重要であるのか，また，予測結果が正解ラベルと異なっているレコードペアを分析することで事前学習済み言語モデルベースの手法の改善点を述べる．SHAP [8] は予測結果に対して特徴量がどの程度寄与しているかを貢献度とよばれる値を各特徴量について計算し予測結果の判断根拠を可視化できる手法である．RoBERTa における特徴量は入力であるテキストデータの各トークンのことであり，テキストの内どのトークンが予測結果に貢献しているかを算出し可視化する．

5.1 可視化と考察

検証用データ 2049 個の内，RoBERTa の予測が正解ラベルと異なったのは 48 個であり，その内訳は False Negative であるレコードペアが 31 個，False Positive が 17 個であった．

False Negative のレコードペアを分析したところ，False Negative の半数以上の 19 個がレコードペアのどちらかの製品番号である model no 属性の値が欠損していることが分かった．商品データにおいて製品番号など主キーにあたる属性値はエンティティマッチングにおいて特に重要であり，Walmart-Amazon データセットにおいて model no 属性の値がレコードペア間で完全一致している全てのレコードペアは一致を表すラベルが割り当てられている．そのため，RoBERTa でも特に製品番号にあたる model no 属性を予測のために重要視していると思われる．しかし，重要な model no 属性の値が欠損しているとレコードペア間で model no の比較ができないため，本来一致しているレコードペアの一致度が減る．これにより他の属性値からレコードペアが一致すると判定できなかった場合に False Negative になってしまうと考えられる．

また，False Positive の約半数の 9 個のレコードペアには表 5 に示すような model no 属性間の Levenshtein 距離が 3 以下と極端に近いという特徴が見られた．図 7 は検証用データに対する予測のうち False Positive であり model no 属性間の Levenshtein 距離が 1 であるレコードペアを SHAP により可視化したものである．赤くハイライトされているトークンはレコードペアがマッチしているという予測に対し貢献しているものを表しており，青は負の方向に寄与していることを表す．また，ハイライトの濃淡により各トークンの貢献度の大きさを表している．先述の通り，RoBERTa は学習により製品番号である model no の値に着目して予測していることが図 7 からも考えられる．しかし，このサンプルは上の表の製品番号を示す model no 属性が僅かに異なっていることからも分かるようにレコードペアは本来マッチしない．このように属性値間で完全に一致しているか否かがマッチングの予測に重要である場合 RoBERTa はそれを考慮できていない．これは製品番号などの ID は数字とアルファベットを並べたものが多く，これらの僅かな違いは埋め込みベクトル空間上でもかなり近い距離に存在していることが考えられ，これらの僅かな違いを学習モデルが認識して予測することが困難であると思われる．したがって文字列の完全一致が重要である単語の完全一致性を考慮する機構を追加することでさらなる分類精度の向上が可能になると考えられる．

left model no	right model no	Levenshtein 距離
f3u138b06	f3u138-06	1
gbu421w6	gbu421	2
ide3510u2	uni3510u2	3

表 5: False Negative であるレコードペアの中で model no 属性値が僅かに異なる例

title	category	brand	model no	price
belkin pro series 6 usb 2.0 5-pin mini-b cable	electronics - general	belkin	f3u138b06	7.88
belkin f3u138-06 pro series usb 5-pin mini-b cable 6-feet	usb cables	belkin	f3u138-06	5.05

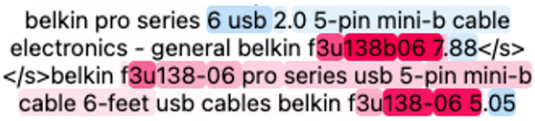


図 7: False Positive サンプルの SHAP による可視化．上の表は対応するレコードペアを属性別に表記したものであり一行目が左テーブル，二行目が右テーブルのデータに対応する．

6 結論

本研究ではデータ量と性質の観点から複数の代表的なエンティティマッチング手法の比較を行った．この評価実験の結果により，学習データ数の観点からは 100 から 1000 程度と少ない場合を除き，事前学習済み言語モデルベースの手法が他の手法よりも良い分類精度を達成し，さらに欠損値に対しても頑健であることが分かった．また，予測結果の分析と可視化により RoBERTa は商品データにおいてマッチングに重要な製品番号に注目して予測を行なっているがその完全一致まで考慮できていないことが観測された．このため，事前学習済言語モデルが不得意と考えられる文字列の完全一致性を考慮する機構を追加することでエンティティマッチングの判定性能の向上が可能であると見込まれる．

参考文献

[1] Mikhail Bilenko and Raymond J Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 39–48, 2003.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[3] Ursin Brunner and Kurt Stockinger. Entity matching with transformer architectures - a step forward in data integration. In *EDBT*, 2020.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Pradap Konda, Sanjib Das, C. PaulSuganthanG., AnHai Doan, A. Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeffrey F. Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. Magellan: Toward building entity matching management systems. *Proc. VLDB Endow.*, 9:1197–1208, 2016.

[6] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang Chiew Tan. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14:50 – 60, 2020.

[7] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[8] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, pages 4768–4777, 2017.

[9] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. *Proceedings of the 2018 International Conference on Management of Data*, 2018.

[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[11] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[12] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.