グラフニューラルネットワークを用いたエンドツーエンド表構造解析手法の提案

青柳　拓志† 　金澤　輝一†† 　高須　淳宏†† 　上野　史††† 　太田　学†††

† 岡山大学大学院自然科学研究科　〒 700–8530 岡山県岡山市北区津島中 3-1-1
†† 国立情報学研究所　〒 101–8430 東京都千代田区一ツ橋 2-1-2
††† 岡山大学学術研究院自然科学学域　〒 700–8530 岡山県岡山市北区津島中 3-1-1
E-mail: †ao2516@s.okayama-u.ac.jp, ††{tkana，takasu}@nii.ac.jp, †††{uwano, ohta}@okayama-u.ac.jp

あらまし　学術論文では実験結果を表にまとめることが多いが，実験結果を一目で把握するには視覚的に優れたグラフが適している．そのため，表からグラフを自動生成する研究が行われているが，表の形式は著者によって様々であるため，まず表の構造を解析する必要がある．そこで，本稿では表検出を含むグラフニューラルネットワーク（GNN）によるエンドツーエンド表構造解析手法を提案する．提案手法は，まず表の検出を行い，表中のトークンをグラフのノード，隣接関係をグラフのエッジとするグラフを生成し，GNN によって隣接トークンが同じセルにあるか否かを推定する．また，表構造解析精度の向上のため，日米の公的機関が発行した文書から 800 表を選び構造情報を付与し，これを ICPRAM 2022 で著者らが発表した表構造解析手法で用いた 209 表に加えて学習に用いる．実験では，ICDAR 2013 table dataset を用いて表構造解析精度を評価し，その精度を商用の ABBYY FineReader PDF 等と比較した．その結果，エンドツーエンドの表構造解析精度の一つであるセルの隣接関係の再現性の F 値は 0.984 となり，ABBYY FineReader PDF のそれより 2.4 ポイント高かった．また，表検出を人手で行った場合のセルの隣接関係の再現性の F 値は 0.986 となり，ICPRAM 2022 で著者らが発表した表構造解析手法を 1.4 ポイント上回った．

キーワード　表構造解析，文書解析，グラフニューラルネットワーク

1 はじめに

表は様々な文書で用いられる．例えば，学術論文では実験結果を表でまとめることが多い．しかしながら，そのような実験結果を一目で把握，比較するには表よりも視覚的に優れたグラフが適しているため，数値を含む表からグラフを自動生成する研究が行われている．また，表の書き方は著者によって様々であるため表の構造を解析する必要がある．近年は，arXiv[1] などのウェブサイトから学術論文のような文書を PDF で入手することが容易となったこともあり，PDF 文書を対象とした表の構造解析に関する研究が行われている [1], [2].

例えば著者らが発表した表構造解析手法 [2] では，表中の隣接トークンを結合する水平結合，垂直結合，セル生成と，実際には引かれていない罫線を推定する補助罫線推定の 4 つのニューラルネットワーク（NN）モジュールを用い，表の構造を解析する. [2] では，評価データセットとして ICDAR 2013 table dataset [3] を，評価指標としてセルの隣接関係の再現性に基づく評価指標 [4] を用いた．その結果, [2] の手法はセルの隣接関係の再現性の F 値 0.972 を達成し，ICDAR 2013 table competiton [3] における最高成績の参加者の F 値を 2.6 ポイント上回った.

本稿では，グラフニューラルネットワーク（GNN）を用いて [2] を改良したエンドツーエンドの表構造解析手法を提案する．提案手法は，水平結合，補助罫線推定，セル生成の 3 つの NN モジュールを用いて表の構造を解析する．水平結合およびセル生成は，まず表中のトークンをグラフのノード，隣接関係をエッジとするグラフを生成し，GNN で隣接トークンが同じセルにあるか推定し，そうである場合隣接トークンを結合する．なお，補助罫線推定は [2] と同様である．また，提案手法では CascadeTabNet [5] を利用して文書中の表を検出することでエンドツーエンドの表構造解析を行う [6]．また，各 NN モジュールの学習データセットに，日米の公的機関が発行した文書に含まれる表に著者らが構造情報を付与した 800 表に [2] で用いた 209 表を加えた計 1,009 表を用いる．評価実験では，評価データセットとして ICDAR 2013 table dataset を用い，表構造解析精度の評価指標としてセルの隣接関係の再現性に基づく評価指標および HTML で表した表の木構造として類似度を定義した Tree-Edit-Distance-based Similarity (TEDS) [7] を用いる．また，提案手法の表構造解析精度を，OCR で文書のデジタル化を行う商用ソフトウェアである ABBYY FineReader PDF[2] 等と比較する.

2 関連研究

Chi らは，GNN を用いた PDF ファイル中の表の構造解析手法である GraphTSR を提案した [8]．また，PDF の 15,000 表で構成された大規模データセットである SciTSR を作成し

1：https://arxiv.org

2：https://pdf.abbyy.com

た．GraphTSR では，まず PDF ファイルから矩形に対応したセルの内容を取得する．次に，取得したセルに対して無向グラフを構築する．このグラフのノードは，セルのサイズ，絶対位置，相対位置の 3 種類の特徴を持つ．グラフのエッジは，セル間のユークリッド距離，絶対位置による距離，相対位置による距離を特徴として持つ．加えて，セルのペアの x 座標，y 座標の重複もエッジの特徴とする．また，GraphTSR では Graph attention を GNN モデルに導入した．SciTSR の構築のために，arXiv から取得した TeX ファイルから表を抽出し，その表をコンパイルすることで PDF ファイルを生成した．実験では，SciTSR と ICDAR 2013 table dataset を用い，表構造解析精度を評価した．その結果，SciTSR のセルの隣接関係の再現性を示す F 値が 0.953，複雑な表を含む SciTSR-COMP の F 値が 0.955 となり，ベースラインの DeepDeSRT [9] と比較して，それぞれ 6.3 ポイント，10.9 ポイント高かった．

Raja らは，文書画像中の表の構造解析モデルである TabStruct-Net を提案した [10]．TabStruct-Net は，セル検出，検出したセル間の行，列関係の決定（構造解析），解析結果を XML 形式で出力する後処理の 3 段階で構成されている．セル検出には，Mask R-CNN [11] を採用し，表画像からセルを検出する．構造解析ではグラフを用いて表構造を解析する．Structure recognition network は，セル検出で検出されたセル間の行関係と列関係を推定する．後処理では，推定されたセルから Tesseract[3]によってセルの内容を抽出し，セルのスパン情報や内容とともに検出したセルの座標を出力する．評価には，SciTSR，SciTSR-COMP，ICDAR 2013 table dataset，ICDAR 2019 table dataset [12]，unlv [13]，Marmot [14]，TableBank [15] と PubTabNet [7] を用いた．実験の結果，特に ICDAR 2013 table dataset において，セルの隣接関係の再現性を示す F 値が 0.981 となり，ベースラインである TableNet [16] や GraphTSR をそれぞれ 7.1 ポイント，10.9 ポイント上回った．また，表構造解析の誤り分析の結果，多くの空白セルを持つ表に誤りが多いことがわかった．

Zheng らは，視覚ベースの表検出と表構造解析モデルである Global Table Extractor (GTE) を提案した [17]．GTE は，Table boundary network (GTE-Table) や Cell structure recognition network (GTE-Cell) などの視覚ベースの複数の NN で構成されており，Object detection network は他のネットワークからの出力を利用する．GTE-Table は，表は少なくとも複数のセルを持たなければならないといった制約を明示的に学習する Cell detection network を用いる．GTE-Cell は，GTE-Table から与えられる表の境界と Table-level style information を用いる．表検出モデルの学習には，TableBank と PubTabNet を用い，セル構造認識モデルの学習には PubTabNet を用いた．実験では，ICDAR 2013 table dataset と ICDAR 2019 table dataset に対して，表構造解析の精度を評価した．その結果，ICDAR 2013 table dataset では表構造解析精度のセルの隣接関係の再現性を示す F 値が 0.962 となり Tensmeyer らの手法 [18] と比較して 1.0 ポイント高かった．

3：https://github.com/tesseract-ocr/tesseract

3 提案するエンドツーエンド表構造解析手法

3.1 エンドツーエンド表構造解析手法の概要

本稿で提案するエンドツーエンドの表構造解析手法の概要を図 1 に示す．青でハイライトされた部分は NN モジュールである．提案手法では，まず入力として，PDF 文書を与える．次に，PDF 文書を画像に変換し CascadeTabNet を用いて，文書画像から表を検出する．また，pdfalto[4]を用いて，PDF 文書を XML ファイルに変換する．その後，検出された表領域と PDF 文書の XML ファイルを入力とし表構造解析を行う．表構造解析は，前処理，水平結合，補助罫線推定，セル生成，後処理で行う．最後に，解析した表構造を XML ファイルまたは HTML ファイルとして出力する．

3.2 CascadeTabNet による表検出

CascadeTabNet [5] は，Cascade R-CNN を用いて文書画像中の表検出と表中のセル検出を行うエンドツーエンド表構造解析手法である．[5] の評価実験では，ICDAR 2013 table dataset，ICDAR 2019 table dataset，TableBank のデータセットを用いた．その結果，表検出において ICDAR 2013 table dataset と TableBank で表検出の再現性を表す F 値がそれぞれ 1.0，0.943（いずれも第 1 位）となった．

本稿で提案する表構造解析手法では，CascadeTabNet は表検出のみに使用し，セルの検出には用いない．また，CascadeTabNet の実装は GitHub 上で公開されているもの [5]を利用する．

3.3 表の構成要素

図 2 に本稿で扱う表の例を示す．図中で，赤い矩形に囲まれたものをトークンと呼び，罫線または補助罫線で囲まれているものをセルと呼ぶ．補助罫線は，実際には引かれていないがセルを分割するために必要である罫線である．なお，図 2 中の実線が罫線，点線が補助罫線である．トークンは，表中の単語であることが多い．データの種類を区別する際などに使用される行をサブヘッダ行と呼び，図中で緑色でハイライトされた行である．

3.4 GNN を用いた表構造解析手法の概要

GNN を用いた表構造解析は，図 1 の前処理から後処理までの部分である．GNN を用いた表構造解析では，入力として，CascadeTabNet により検出された表の領域と，pdfalto により PDF 文書を XML ファイルに変換した XML を与える．さらに，文書画像中の表領域の座標を XML ファイルにおける座標に変換する．そして，変換された XML 中の表領域に基づいて，XML から表中のトークンを取得する．同時に，PDFMiner[6]と

4：https://github.com/kermitt2/pdfalto
5：https://github.com/DevashishPrasad/CascadeTabNet
6：https://github.com/pdfminer/pdfminer.six

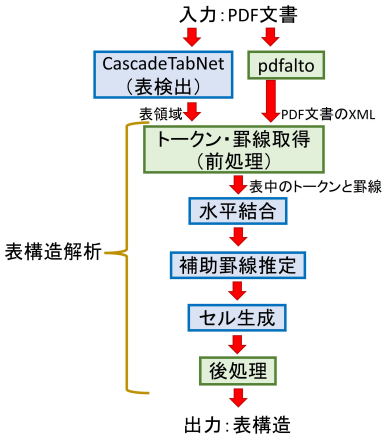

図 1　提案するエンドツーエンド表構造解析手法

	Method 1	Method 2	Method 3
Dataset A			
a1	0.91	0.87	0.85
a2	0.86	0.92	0.89
Dataset B			
b2	0.96	0.92	0.91

図 2　表と表の構成要素

OpenCV[7]を用いて表中の罫線を取得する．次に，表中の水平方向に隣接する 2 トークンを，トークンの特徴を用いて再帰的に結合する水平結合を行う．結合し終えたら，トークンの位置関係に基づいて補助罫線を推定する．そして，垂直方向と水平方向の隣接 2 トークンを交互に再帰的に結合してセルを生成する．なお，各 NN モジュールの詳細については 3.5 節以降で述べる．最後に，Ohta らの手法 [1] と同様に後処理で行や列の結合とセルの拡張を行い，最終的な表構造を確定する．

3.5　水平結合

3.5.1　水平結合の概要

水平結合は，水平方向に隣接する 2 トークンを繰り返し結合する処理で，結合する隣接トークンがなくなるまで行う．具体的にはまず，トークンをノード，トークンの隣接関係をエッジとする無向グラフを生成する．次に，GNN で隣接 2 トークンが同じセル中にあるか否か推定し，同じセル中にあると推定された場合，水平方向に隣接する 2 トークンを結合する．

図 3 に，水平結合の例を示す．なお，本稿の以降の例は ICDAR 2013 table dataset [3] の表である．まず図 3 の上の表のように，トークンの隣接関係から無向グラフを生成する．このとき，赤線で囲まれたトークンをノード，緑と青の線で示されるトークンの水平方向と垂直方向の隣接関係をエッジとする．次に，水平方向に隣接する 2 トークンが同じセル中にあるか GNN で推定して結合する．例えば，図 3 では，“Postnatal” と “Day” と “1” などの隣接トークンが結合されている．

3.5.2　水平結合の GNN モデル

水平結合の GNN モデルを図 4 に示す．なお，図中の n は

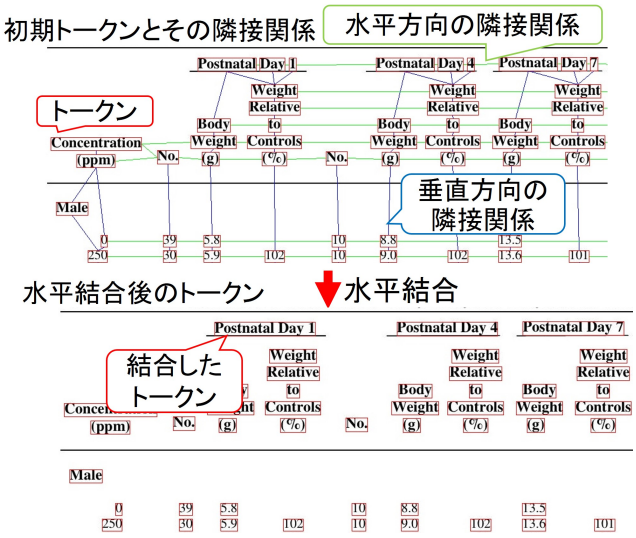

図 3　水平結合の例

全トークン数，e は全トークンペア数である．図 4 の GNN モデルは，Graph Attention Networks (GAT) [19] であり，モデルの実装には PyTorch Geometric[8]を利用する．このモデルの入力は，トークンのテキストの 100 次元の分散表現，トークンの 42 次元の特徴ベクトル，垂直方向と水平方向の隣接 2 トークンの 11 次元の隣接特徴のベクトル，同じセル中にあるか推定する隣接 2 トークンの 84 次元（42 次元 ×2）の特徴ベクトルおよびその 11 次元の隣接特徴のベクトルである．なお，結合を推定する隣接 2 トークンの特徴はそれぞれのトークンの特徴ベクトルを連結したものである．また，テキストの分散表現の獲得には Word2vec [20] の学習済みモデルとして glove-wiki-gigaword-100[9] を利用する．図 4 のモデルでは，まず，トークンのテキストの分散表現と隣接 2 トークンの隣接特徴のベクトルを左の GAT 層に入力し，トークンの分散表現の埋め込み表現を得る．同時に，同様にトークンの特徴ベクトルと隣接 2 トークンの隣接特徴のベクトルを右の GAT 層に入力し，トークンの埋め込み表現を得る．次に，得られたそれぞれの埋め込み表現から，同じセル中にあるか推定される隣接関係が結ぶ隣接 2 トークンに対応するベクトルをそれぞれ取り出し，連結する．例えば，図 5 において，隣接するトークンの番号が i と j である場合，i 番目のトークンに対応するベクトルと j 番目のトークンに対応するベクトルを取り出し連結する．そして，得られたトークンの分散表現の埋め込み表現，トークンの埋め込み表現と同じセル中にあるか推定される隣接 2 トークンの特徴およびその隣接特徴を連結し，276 次元のベクトルを全結合層に入力する．そして，出力層で水平方向の隣接 2 トークンが同じセル中にあるか否かの 2 次元のベクトルを出力する．このモデルの活性化関数は，出力層では Log softmax 関数，それ以外では ReLU である．また，損失関数に 2 値クロスエントロピーを用い，最適化関数に Adam を用いる．

7：https://opencv.org

8：https://github.com/pyg-team/pytorch_geometric

9：https://nlp.stanford.edu/projects/glove

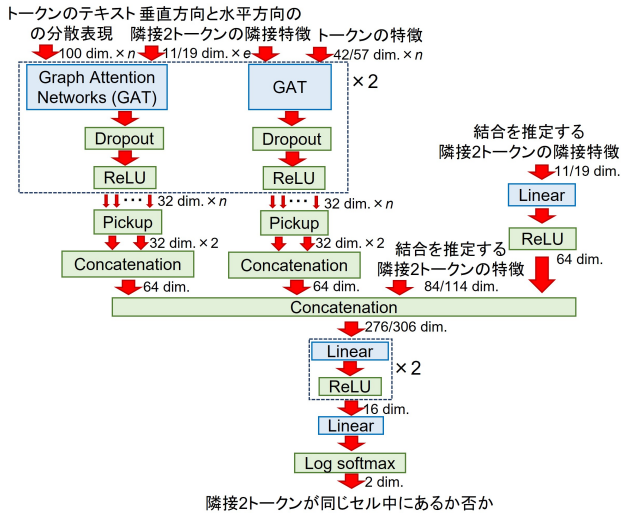

図 4　水平結合/セル生成の GNN モデル

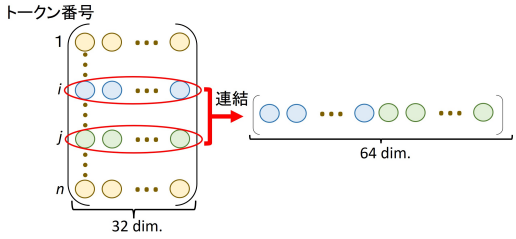

図 5　隣接 2 トークンの埋め込み表現の連結

3.5.3　水平結合の GNN モデルへの入力特徴

水平結合の GNN モデルへの入力特徴であるトークンの特徴を表 1，隣接 2 トークンの隣接特徴を表 2 に示す．また，トークンの特徴の一部である周辺の各トークンの特徴を表 3 に示す．まず，表 1 のトークンの特徴について述べる．トークンのサイズは，表の幅（高さ）に対するトークンの幅（高さ）の比率である．他トークンとの座標の一致は，注目するトークンの左上，重心，右下の x，y 座標が他のトークンのそれらと一致した回数である．トークンの品詞は，Natural Language Toolkit (NLTK)[10] を用いて取得した品詞に基づく 12 次元の one-hot ベクトルである．サブヘッダの可能性は，注目するトークンと水平方向に隣接するトークンが存在しない場合に 1 とする．また，周辺のトークンは図 6 に示すようにトークン自身と上下左右に隣接するトークンを合わせた 5 トークンである．図 6 では，青線で囲まれたトークンである “63.4” の周辺トークンは “63.4” 自身と赤線で囲まれた 4 トークンがある．また，周辺の各トークンは表 3 の 3 次元の特徴ベクトルを持つ．

次に，表 2 の隣接 2 トークンの隣接特徴について述べる．隣接トークン間の距離は，水平方向の隣接関係の場合は左側のトークンの右端の x 座標と右側のトークンの左端の x 座標との差の絶対値であり，垂直方向の隣接関係の場合は上側のトークンの下端の y 座標と下側のトークンの上端の y 座標との差の絶対値である．隣接トークン間のテキストの類似度は，Python のライブラリである difflib を用いて算出する．結合位置は，表の幅（高さ）に対する隣接トークン間の重心の中点の座標が占める比率とする．

10：https://www.nltk.org

表 1　水平結合で利用するトークンの特徴

特徴	次元数
トークンのサイズ	2
他トークンとの座標の一致	6
トークンのテキストが数値であるか	1
トークンのテキストの品詞	12
サブヘッダの可能性	1
トークンのテキストの先頭文字が大文字であるか	1
トークンのテキストが括弧の中にあるか	1
トークンのテキストが記号であるか	1
トークンのテキストが記号で開始するか	1
トークンのテキストが記号で終了するか	1
トークンの周辺のトークンの特徴	15
合計	42

表 2　水平結合で利用する隣接 2 トークンの隣接特徴

特徴	次元数
隣接トークン間の距離	1
隣接トークン間のフォントの一致	1
隣接トークン間のスタイルの一致	1
隣接トークン間の重心の一致	1
隣接トークン間の左上の一致	1
隣接トークン間の右上の一致	1
隣接トークンのテキストの類似度	1
隣接トークン間に罫線が引かれているか	1
結合位置	2
隣接方向	1
合計	11

表 3　周辺の各トークンの特徴

特徴	次元数
幅	1
高さ	1
テキストが数値か	1
合計	3

3.6　補助罫線推定

提案する表構造解析手法では，3.5 節の水平結合後に，トークンの位置関係に基づいて補助罫線を推定する．具体的にはまず，垂直方向の補助罫線候補としてトークンの左端，右端，水平方向に隣接する 2 トークンの重心の中点，水平方向の補助罫線候補としてトークンの上端，下端，垂直方向に隣接する 2 トークンの重心の中点のそれぞれ 3 種類の点集合を作成する．トークンの各点の例を図 7 に示す．図 7 の緑の +，×，∗ は，それぞれトークンの左端，右端，水平方向の隣接トークン間の中点であり，青のそれらはトークンの上端，下端，垂直方向の隣接トークン間の中点である．そして，それぞれの点集合を重心法でクラスタリングして得られた点集合のクラスタを補助罫線候補とし，その補助罫線候補が補助罫線であるか推定する．

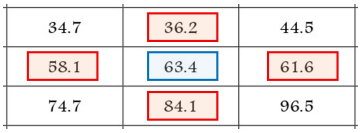

図 6　周辺のトークンの例

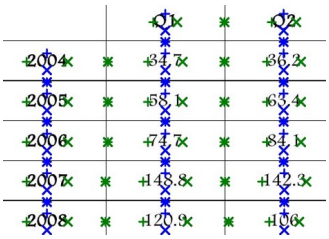

図 7　トークンの上下左右の端点と重心の中点

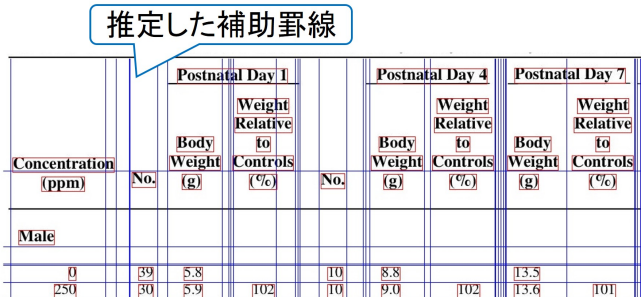

図 8　補助罫線推定の例

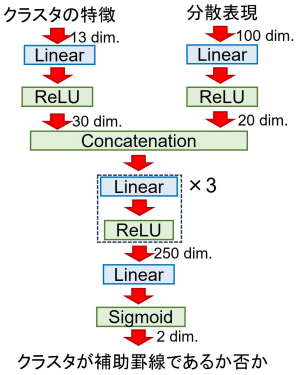

図 9　補助罫線推定の NN モデル

補助罫線推定の例を図 8 に示す．図 8 の青線は提案手法が推定した補助罫線を表す．

この補助罫線推定の NN モデルの概要を図 9 に示す．図 9 の NN モデルの入力は，表 4 に示す補助罫線候補の 13 次元の特徴ベクトルと，補助罫線候補中の各点の生成に関与したトークンのテキストの 100 次元の分散表現の平均である．この分散表現は水平結合で使用するものと同じである．また，出力は補助罫線か非補助罫線かの 2 次元である．中間層の出力次元数は，連結層より前では分散表現を入力とする層は 20，クラスタの特徴を入力とする層は 30 とし，結合層より後では 250 とする．出力層の活性化関数は Sigmoid 関数，それ以外の層は ReLu を用いる．また，損失関数には 2 値クロスエントロピーを用い，最適化関数には Adam，学習率は 0.01 とする．

表 4　補助罫線推定のクラスタの特徴

特徴	次元数
クラスタを構成する点の数	1
表中の水平 (垂直) 方向のトークン数	1
補助罫線候補を挟むトークン間の罫線の有無	1
補助罫線候補の方向	1
クラスタの種類	6
補助罫線候補がセル上を通るか	1
補助罫線候補の位置	1
表のサイズ	1
合計	13

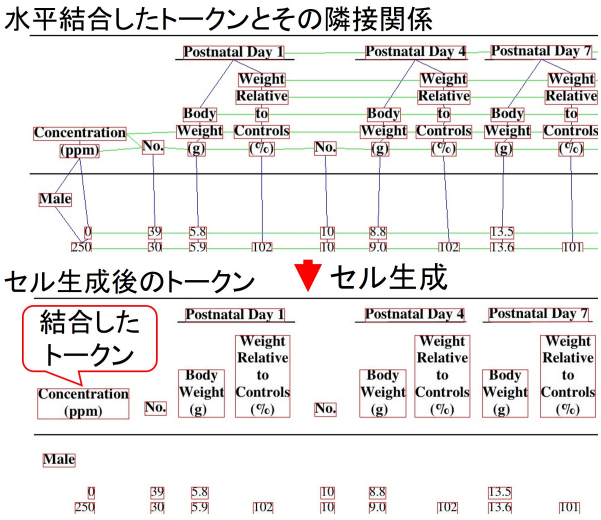

図 10　セル生成の例

3.7　セル生成

3.7.1　セル生成の概要

セル生成では，3.6 節の補助罫線推定後，垂直方向に隣接する 2 トークンと水平方向に隣接する 2 トークンを交互に結合する．なお，この結合は結合できる隣接 2 トークンがなくなるまで反復的に行う．セル生成では，水平結合と同様にグラフを生成し，GNN で隣接する 2 トークンを結合する．図 10 では，例えば，“Concentration” と “(ppm)” の垂直方向の隣接トークンが結合されている．

セル生成の GNN モデルは図 4 に示した水平結合のものと同じである．ただし，セル生成のモデルは入力特徴の次元数が異なる．すなわち入力は，トークンのテキストの 100 次元の分散表現，トークンの 57 次元の特徴ベクトル，隣接 2 トークンの 19 次元の隣接特徴ベクトル，同じセル中にあるか推定する隣接 2 トークンの 114 次元（57 次元 ×2）の特徴ベクトルおよびその 19 次元の隣接特徴のベクトルである．なお，セル生成では垂直方向または水平方向の隣接トークンの結合が推定対象であり，結合を推定する隣接 2 トークンの特徴はそれぞれのトークンの特徴ベクトルを連結したものである．なお，分散表現は水平結合と同じものを使用する．

3.7.2　セル生成の入力特徴

セル生成の GNN モデルへの入力特徴は，3.5 節で説明した水平結合のモデルへの入力特徴と同じである．ただし，次の差

表 5　作成した構造情報付きの表データセット

文書発行元 URL	表数
https://www.ers.usda.gov	265
https://bjs.ojp.gov	167
https://nces.ed.gov	81
https://www.stat.go.jp	80
https://www.bea.gov	65
https://www.census.gov	63
https://www.bls.gov	59
https://www.transportation.gov	15
https://www.cdc.gov	5
合計	800

異がある．まず，セル生成で用いるトークンの特徴では，表 1 のトークンのテキストの品詞を，水平結合されたトークンに関しては最初と最後のトークンのテキストの品詞を連結したものとする．さらに，トークンを構成する単語数とトークンが属する列や行のトークン数を追加する．後者の特徴は，注目トークンの上下の延長上にあるトークンの数を列のトークン数，左右の延長上にあるトークン数を行のトークン数とする．また，セル生成では表 2 の隣接 2 トークンの隣接特徴に隣接トークン間に存在するセパレータを構成する点の数を追加する．セパレータは，罫線，罫線の延長線，数値間の補助罫線，3.6 節で説明した 6 種類の点集合から生成した補助罫線の計 9 種類あり，それぞれのセパレータごとのクラスタを構成する点の数をこの特徴に用いる．

4　評価実験

4.1　表データセット

本稿で提案する表構造解析手法の各 NN モジュールの学習には，[2] で用いた 209 表と表 5 の 800 表を用いる．なお，後者の表データセットは，表 5 に示す日米の公的機関のウェブサイトにおいて 2022 年 1 月から 2022 年 10 月までに発行された PDF 文書を収集し，収集した文書に含まれる表に表構造を本稿の第一著者が付与したものである．付与する表構造の情報は，セルが位置する行や列，セルのテキスト情報などである．例えば，図 11 の表とその表構造では，緑のセルの範囲は 1 行 1 列から 2 行 1 列までであり，このセルは複数行にまたがるセルである．

表構造解析精度の評価には，ICDAR2013 Table competition [3] のテスト用データセットを用いる．この表データセットは，EU，米国政府の発行した文書から 156 の表を収集したものであり，それらの表を含む PDF 文書，表構造の付与された XML からなる．

4.2　評価指標

表構造解析精度の評価指標には，Göbel らが定義した表中のセルの隣接関係の再現性に基づく評価指標 [4] と Zhong らが定義した TEDS [7] を用いる．

セルの隣接関係の再現率と適合率は，それぞれ式 (1)，式 (2) で算出される．

表の一部

Year	Rates per 1,000 population [1]				Total fertility rate [2]	Life expectancy at birth (years)	
	Live births	Deaths	Infant mortality	Natural change		Males	Females
1950	28.1	10.9	60.1	17.2	3.65	a) 59.57	a) 62.97

セルの表構造情報

```
−<cell end-col="0" end-row="1" id="1" multiple="column" start-col="0" start-row="0">
   <content>Year</content>
   <token id="p32_w170"/>
 </cell>
```

図 11　表と表構造の例

$$\text{再現率} = \frac{\text{解析結果の正しい隣接関係の数}}{\text{正解データの隣接関係の数}} \tag{1}$$

$$\text{適合率} = \frac{\text{解析結果の正しい隣接関係の数}}{\text{解析結果の隣接関係の数}} \tag{2}$$

また，F 値はこの再現率と適合率の調和平均である．

一方 TEDS は，HTML 形式で表された正解データの表構造と解析結果の表構造の類似度を木編集距離に基づいて定めた類似度である．TEDS の算出式を式 (3) に示す．

$$\text{TEDS}(T_a, T_b) = 1 - \frac{\text{EditDist}(T_a, T_b)}{\max(|T_a|, |T_b|)} \tag{3}$$

ここで T_a，T_b はそれぞれ正解データと解析結果の表，$\text{EditDist}(T_a, T_b)$ は T_a，T_b の木編集距離，$|T_a|$，$|T_b|$ はそれぞれの表の HTML 木構造のノード数である．

4.3　実験結果

4.3.1　表構造解析精度

表検出を含むエンドツーエンド表構造解析の精度を表 6，表検出を含まない場合の表構造解析精度を表 7 にまとめる．なお，提案手法では表検出に CascadeTabNet を用いており，表検出を含まない場合は手動で表領域を指定している．また，マイクロ平均は解析結果のすべての表のセルの隣接関係を集計しセルの隣接関係の再現率，適合率，F 値を算出したものであり，マクロ平均は表ごとにセルの隣接関係の再現率，適合率，F 値を算出し，最後にそれぞれの平均をとったものである．まず，提案手法の表検出を含む場合の表構造解析精度は，ICPRAM 2022 の手法 [2] の精度と比較して，セルの隣接関係の再現性の F 値のマイクロ平均とマクロ平均がそれぞれ 2.2 ポイント，2.7 ポイント高く，TEDS が 2.1 ポイント高かった．また，商用ソフトウェアの ABBYY FineReader PDF の文字認識誤りを修正した場合と比べて，セルの隣接関係の再現性の F 値がそれぞれ 0.9 ポイント，1.3 ポイント，TEDS が 0.4 ポイント高かった．表検出を含まない場合は表 7 に示す通り，[2] のセルの隣接関係の再現性の F 値よりそれぞれ 1.4 ポイント，1.6 ポイント，TEDS が 1.3 ポイント高かった．

4.3.2　水平結合の精度

表 8 に示す通り，結合すべきだがしなかった誤りが 6，その逆の誤りが 17 あった．よって水平結合は，結合すべきトークンペアの 99.8%の結合に成功する一方，結合すべきでないペアの 20.2%を誤って結合したことになる．

4.3.3　補助罫線推定の精度

実験における訓練データの補助罫線候補のクラスタ数は，補助罫線であるクラスタが 83,776 件，補助罫線でないクラスタが

表 6　表検出を含む場合のエンドツーエンド表構造解析精度

入力	手法	セルの隣接関係（マイクロ平均）			セルの隣接関係（マクロ平均）			TEDS
		再現率	適合率	F 値	再現率	適合率	F 値	
PDF	提案手法	0.984	0.985	0.984	0.976	0.972	0.974	0.969
	ICPRAM 2022 [2]	0.954	0.970	0.962	0.943	0.951	0.947	0.948
画像	ABBYY FineReader PDF	0.950	0.971	0.960	0.943	0.951	0.947	0.958
	ABBYY FineReader PDF（文字認識誤り修正有）	0.965	0.986	0.975	0.958	0.964	0.961	0.965
	GTE [17]	–	–	–	0.927	0.944	0.935	–

表 7　表検出を含まない場合の表構造解析精度

入力	手法	セルの隣接関係（マイクロ平均）			セルの隣接関係（マクロ平均）			TEDS
		再現率	適合率	F 値	再現率	適合率	F 値	
PDF	提案手法	0.986	0.986	0.986	0.978	0.977	0.977	0.982
	ICPRAM 2022 [2]	0.967	0.977	0.972	0.955	0.968	0.961	0.969
	GraphTSR [8]	0.885	0.860	0.872	0.819	0.855	0.837	–
画像	TabStruct-Net [10]	0.976	0.985	0.981	–	–	–	–
	GTE [17]	–	–	–	0.958	0.968	0.962	–

表 8　水平結合の推定結果

		推定	
		結合する	結合しない
正解	結合する	3,033	6
	結合しない	17	67

表 9　補助罫線の推定結果

		推定	
		補助罫線	補助罫線でない
正解	補助罫線	13,423	412
	補助罫線でない	482	3,630

表 10　セル生成の推定結果

		推定	
		結合する	結合しない
正解	結合する	493	179
	結合しない	7	25,127

38,217 件あった．このように，補助罫線でない点のクラスタ数と補助罫線であるクラスタ数が不均衡であったため，本稿では SMOTEENN [21] で同数にしたものを学習に用いる．表 9 に示す通り，補助罫線であるクラスタがそうでないと誤推定されたクラスタの割合は 3.0%，その逆に補助罫線でないクラスタのうち補助罫線であると誤推定されたクラスタの割合は 11.7%となった．

4.3.4　セル生成の精度

セル生成における推定結果は表 10 に示す通り，結合すべきだが結合しなかった誤りが 179，その逆の誤りが 7 だった．セル生成では，結合すべきトークンペアの 26.6%を結合できなかったが，結合すべきでないペアを誤って結合したのはわずかに 0.028%だった．

5　考　　察

5.1　提案手法の表構造解析誤りの分析

5.1.1　水平結合の誤り

水平結合で誤ってトークンを結合した例を図 12 に示す．この図の “than” と “$10,000-” が誤って水平方向に結合されている．これは，そのトークンペア間の距離が非常に小さいことが原因であると考えられる．実際 “than” と “$10,000-” 間の距離はその下の “$10,000” と “$14,999” 間の距離と比較しかなり小さい．また，“Less than $10,000-” は，本来のセルの内容である “Less than $10,000” とほぼ同じテキストでもある．

5.1.2　補助罫線推定の誤り

補助罫線推定で誤って推定した補助罫線の例を図 13 に示す．この図では “Unadjusted” と “odds ratio” の間に水平の補助罫線が誤って推定されている．これは，この補助罫線候補のクラスタが “(%)” や “odds ratio” など多くのトークンの周囲の点から生成されたからである．補助罫線推定では，多くのトークンが関わるクラスタは補助罫線であることが多い．

5.1.3　セル生成の誤り

セル生成で結合できなかったトークンの例を図 14 に示す．この図では，円で囲んだ “American Indian/Alaska” と “Native” の隣接トークンペアが結合されていないことがわかる．“American Indian/Alaska” とその右方向に存在する “352” や “55” といった数値であるトークンの下端がほぼ揃っており，相対的に揃っているトークン数が多いことが要因といえる．

5.2　作成した表データセットの有効性

提案手法の評価においてセルの隣接関係の再現性の F 値のマイクロ平均，マクロ平均および TEDS は，学習データが 209 表の場合それぞれ 0.982，0.970，0.977 だった．これに表 5 の 800 表を加えた 1,009 表で学習した場合，表 7 の通りそれぞれ 0.4 ポイント，0.7 ポイント，0.5 ポイント向上した．

水平結合におけるトークンの誤結合と結合漏れは，学習データが 209 表の場合それぞれ 17 と 4 だった．学習データが 1,009 表の場合は表 8 の通りそれぞれ 17 と 6 であり，結合漏れは 2

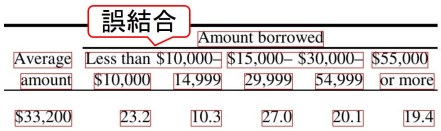

図 12　水平結合で誤って結合されたトークンの例

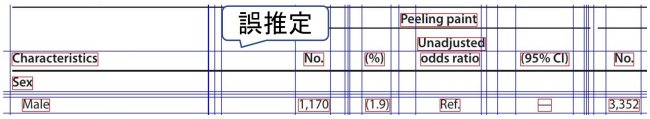

図 13　補助罫線推定で誤って推定された補助罫線の例

図 14　セル生成で結合できなかったトークンの例

増えた．水平結合では，209 表で学習した場合でも比較的高精度でトークンを結合できているといえる．

補助罫線推定における補助罫線の誤推定と推定漏れは，学習データが 209 表の場合それぞれ 881 と 595 だった．学習データが 1,009 表の場合は表 9 の通りそれぞれ 482 と 412 であるため，1,009 表で学習した場合それぞれの誤りが 399 と 183 減少したことになる．このことから，本稿で作成した 800 表の表データセットは補助罫線推定の精度向上に寄与したといえる．

セル生成におけるトークンの誤結合と結合漏れは，学習データが 209 表の場合それぞれ 11 と 270 だった．学習データが 1,009 表の場合は表 10 の通り 7 と 179 であり，それぞれの誤りが 4 と 91 減少した．よって本稿で作成した表データセットはセル生成の精度向上に寄与したといえる．

6　ま　と　め

本稿では，グラフニューラルネットワークを用いたエンドツーエンド表構造解析手法を提案した．提案手法は，水平結合，補助罫線推定，セル生成の 3 つの NN モジュールを持ち，水平結合およびセル生成で GNN を用いて表中の隣接トークンを結合する．評価実験の結果，表検出を含むエンドツーエンド表構造解析でのセルの隣接関係の再現性の F 値が 0.984 となり，これは商用の ABBYY FineReader PDF の F 値より 2.4 ポイント高かった．また，表検出を含まない場合のセルの隣接関係の再現性の F 値は 0.986 であり，[2] の F 値より 1.4 ポイント高かった．さらに，本研究で作成した 800 表の構造付与データを学習データに加えたことで，補助罫線推定およびセル生成モジュールの精度ならびに全体の表構造解析精度が向上した．

今後の課題として，ICDAR 2013 table dataset 以外の表データセットによる提案手法の表構造解析精度の評価や，表構造解析結果を利用したグラフ生成アプリの開発等が挙げられる．

謝　　辞

本研究の一部は，科学研究費補助金基盤研究 (B)(課題番号 22H03904)，同基盤研究 (C)(課題番号 18K11989)，新エネルギー・産業技術総合開発機構 (NEDO) の戦略的イノベーション創造プログラム (SIP) 第二期「ビッグデータ・AI を活用したサイバー空間基盤技術」および 2022 年度国立情報学研究所共同研究 (22FC01) の援助による．

文　　献

[1] M. Ohta, et al. Table-structure recognition method using neural networks for implicit ruled line estimation and cell estimation. *DocEng 2021,* Article 23, 7 pages, 2021.

[2] H. Aoyagi, et al. Table-structure recognition method consisting of plural neural network modules. *ICPRAM 2022*, pp. 542–549, 2022.

[3] M. Göbel, et al. ICDAR 2013 table competition. *ICDAR 2013*, pp. 1449–1453, 2013.

[4] M. Göbel, et al. A methodology for evaluating algorithms for table understanding in pdf documents. *DocEng 2012*, pp. 45–48, 2012.

[5] D. Prasad, et al. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. *CVPR 2020*, pp. 2439–2447, 2020.

[6] 青柳拓志他. 表検出を含むエンドツーエンド表構造解析の評価. 第 18 回 WI2 研究会予稿集，No. 01, pp. 1–8, 2022.

[7] X. Zhong, et al. Image-based table recognition: Data, model, and evaluation. *ECCV 2020*, pp. 564–580, 2020.

[8] Z. Chi, et al. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*, 2019.

[9] S. Schreiber, et al. DeepDeSRT: Deep learning for detection and structure recognition of tables in document images. *ICDAR 2017*, pp. 1162–1167, 2017.

[10] S. Raja, et al. Table structure recognition using top-down and bottom-up cues. *ECCV 2020*, pp. 70–86, 2020.

[11] Z. Cai, et al. Cascade R-CNN: High quality object detection and instance segmentation. *IEEE PAMI*, Vol. 43, No. 5, pp. 1483–1498, 2019.

[12] L. Gao, et al. ICDAR 2019 competition on table detection and recognition (cTDaR). *ICDAR 2019*, pp. 1510–1515, 2019.

[13] A. Shahab, et al. An open approach towards the benchmarking of table structure recognition systems. *DAS 2010*, pp. 113–120, 2010.

[14] J. Fang, et al. Dataset, ground-truth and performance metrics for table detection evaluation. *DAS 2012*, 2012.

[15] M. Li, et al. Tablebank: Table benchmark for image-based table detection and recognition. *LREC 2020*, 2020.

[16] S. Paliwal, et al. TableNet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. *ICDAR 2019*, pp. 128–133, 2019.

[17] X. Zheng, et al. Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. *WACV 2021*, pp. 697–706, 2021.

[18] C. Tensmeyer, et al. Deep splitting and merging for table structure decomposition. *ICDAR 2019*, pp. 114–121, 2019.

[19] P. Veličković, et al. Graph attention networks. *ICLR 2018*, 2018.

[20] T. Mikolov, et al. Efficient estimation of word representations in vector space. *ICLR 2013*, 2013.

[21] G. Batista, et al. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, Vol. 6, Issue 1, pp. 20–29, 2004.