ニューラルネットワークによる日本語を含む表の構造解析の一手法

細谷　亮太† 　金澤　輝一†† 　上野　史††† 　太田　学†††

† 岡山大学工学部情報系学科　〒 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: †pkyf9d8a@s.okayama-u.ac.jp, ††tkana@nii.ac.jp, †††{uwano,ohta}@okayama-u.ac.jp

あらまし　表は数値情報を集約して表すために頻繁に用いられるが，一目でデータの変化や差異を確認できるという点ではグラフのほうが適しており，数値を含む表のデータをグラフに自動変換できれば文書の理解の助けになる．しかし，このような自動変換にはまず表構造を正しく解析する必要がある．文書認識の国際会議である ICDAR 2013 では，表検出および表構造解析のための構造情報が付与された表データが公開されたが，それらの表は EU や米国政府の発行した文書に含まれる英語の表であった．そこで，本稿では日本語を含む表の構造をニューラルネットワークにより解析する手法を提案し，日英の言語の違いが表構造解析に与える影響について考察する．本稿では，DEIM2022 の論文中の日本語を含む表 280 件に表構造情報を付与したデータを作成し，学習及び表構造解析精度の評価を行った．

キーワード　表構造解析，ニューラルネットワーク，グラフ自動生成

1 はじめに

学術論文などの様々な文書において，実験結果や統計データをまとめるために，表やグラフが頻繁に用いられる．しかし，一目でデータの変化や差異を確認できるという点においては，表よりグラフのほうが適している．そのため，数値を含む表をグラフに自動変換できれば，閲覧者が文書の内容をより容易に理解することができる．表からグラフを自動生成するには，まず表構造を正しく解析する必要がある．

ICDAR 2013 で提供された表データは，EU や米国政府の発行した文書に含まれる英語の表であり，[1] の手法は英語の表の構造解析を想定している．一方本稿では，[1] の手法を基に日本語を含む表の構造をニューラルネットワークにより解析する手法を提案し，日英の言語の違いが表構造解析に与える影響について考察する．また，提案手法で [1] の手法を改め，日本語に対応するために MeCab[1] や IPAdic，Wikipedia2Vec[2] を利用した．実験では，第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022) の論文中の 280 の日本語を含む表に表構造情報を付与したデータを作成し，表構造解析手法の NN モジュールの学習及び [1] の手法と比較する形で表構造解析精度の評価を行う．

2 関連研究

山田らは，トークンの位置関係から実際には引かれていない補助罫線を推定する NN と，罫線，補助罫線，周囲のトークンなどの特徴量を考慮して，隣接したセルの結合を推定する NN を用いた表構造解析手法を提案した [2]．ICDAR 2013 table competition のデータセットと評価指標を用いた精度評価では，セルの隣接関係の再現性の再現率，適合率，F 値がそれぞれ 0.951，0.960，0.955 となった．

青柳らは表中の隣接トークンを結合する 2 種類のニューラルネットワーク (NN) と，補助罫線を推定する NN を組み合わせた表構造解析手法を提案した [1]．文書解析の国際会議である，International Conference on Document Analysis and Recognition (ICDAR) 2013 の table competition において提供された表データセットと評価指標を用いて表構造解析を行った結果，セルの隣接関係の再現性を示す再現率，適合率，F 値がそれぞれ 0.967，0.977，0.972 となり，これは ICDAR 2013 table competition の参加者の中の最良の記録を上回った．その表構造解析手法の概要を図 1 に示す．この手法ではまず，入力された PDF 文書を，pdfalto[3] を用いて XML に変換する．得られた XML 文書には，トークンの座標，大きさ，フォントなどの情報が含まれている．さらに，PDFMiner[4] を用いて罫線の情報を検出する．次に，トークンの大きさ，フォントなどの検出した情報をもとに，ニューラルネットワークを用いて水平方向に隣接した 2 つのトークンを結合するかどうか推定する．具体的には，水平に隣接する 2 つのトークンの，位置やフォントサイズ，品詞などの特徴量と，その周辺のトークンの座標や幅などの特徴量を入力として，それらを水平結合するかどうか判定する．次に，トークンの特徴量と，トークンの文字列の分散表現を入力として，補助罫線の有無を推定する．さらに，垂直に隣接する 2 トークンの特徴量と，その周辺のトークンの特徴量，罫線と補助罫線の情報を入力することで，それらの 2 トークンを垂直結合するかどうかを推定する．その後，結合されたトークンをさらに隣接トークンと結合することで，セルを生成

1：https://taku910.github.io/mecab
2：https://github.com/singletongue/WikiEntVec
3：https://github.com/kermitt2/pdfalto
4：https://github.com/pdfminer/pdfminer.six

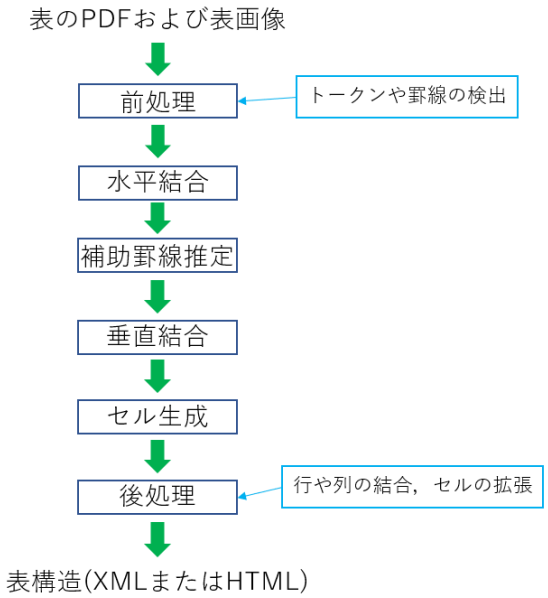

図 1　青柳らの表構造解析手法 [1] の概要

	方法 A	方法 B	方法 C
データセット 1			
データ X	0.45	0.65	0.43
データ Y	0.76	0.19	0.55
データセット 2			
データ Z	0.56	0.98	0.59

図 2　表と表の構成要素

する．なお，このセル生成では，隣接するトークンが水平方向の場合は水平方向に結合し，垂直方向の場合は垂直方向に結合し，それらを交互に行い，結合できるトークンがなくなるまで処理を続ける．最後に，後処理として，行や列を結合し，またセルの拡張を行うことで表構造を決定する．本研究では, [1] の手法を日本語を含む表に適応させる．

3　日本語を含む表の構造解析

3.1　表の構成要素

図 2 に表の例を示す．まず，実線と点線は，それぞれ罫線，補助罫線である．なお，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な線のことである．罫線もしくは補助罫線で囲まれたものをセルと呼ぶ．また，赤い短形で囲まれているものがトークンで，表中の単語などに対応している．青色でハイライトした列をヘッダ列と呼び，各行の名前を表している．同様に，緑色でハイライトした行をヘッダ行と呼び，各列の名前を表している．また，黄色でハイライトされた，ヘッダ行ではないが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ぶ．

3.2　日本語を含む表と英語の表の特徴

日本語を含む表と英語の表の特徴の特徴について述べる． 英

Funder	Delay Period	Funding
NIH Public Access Requirement (Sec. 218)	up to 12 mo	allowable cost for grants
Howard Hughes Medical Institute	up to 6 mo	dedicated fund
European Research Council	up to 6 mo	allowable cost for grants
UK Medical Research Council	up to 6 mo	allowable cost for grants
Wellcome Trust	up to 6 mo	dedicated fund

図 3　英語の表の例 [3]

自由参加型ウェブ百科事典	説明文数
Wikipedia[2]	1,950
ニコニコ大百科（仮）[3]	1,281
ピクシブ百科事典 [4]	1,879
アニオタ Wiki（仮）[5]	1,140
計	6,250

図 4　日本語を含む表の例 [4]

手法	1 回目	2 回目	3 回目	平均
HAN	100.00	96.88	100.00	98.96
RT_1	90.63	40.63	100.00	77.09
RT_2	100.00	96.88	96.88	97.92

図 5　日本語を含む表の例 [5]

語の表と日本語を含む表の例をそれぞれ図 3，図 4 に示す．図 3 は ICDAR2013 table competiton の訓練用データセットに含まれる表である．英語の表では通常単語間に空白がある．一方で日本語を含む表では日本語の単語間に空白がない場合が多く，また全角文字と半角文字が混在しているといった特徴がある．図 5 は図 4 とは別の日本語を含む表の例だが，ここでは “1” と “回目” が別のトークンと指定されている．これは，数字である “1” と漢字である “回” の間が，見やすさのために漢字同士の間隔よりも大きく空けられていることが一因である．以降では日本語を含む表に出現する英語アルファベットのことを単に「英語」と表現する．

3.3　表構造解析手法の概要

この節では図 1 の各モジュールを，日本語を含む表への適応のための改良点をを挙げながら順に説明する．

3.3.1　前　処　理

表構造解析の前処理として, [1] と同様に，入力された PDF 文書を，pdfalto を用いて XML に変換する．さらに，PDFMiner と OpenCV[5] を用いて罫線の情報を検出する．なお，表領域は，変換された XML ファイルにおいて表に該当するトークンの座標を人手で指定することで与える．

3.3.2　水 平 結 合

図 1 に示すように前処理の後にトークンの水平結合を行う．水平結合では，水平方向の隣接 2 トークンの特徴量とその周辺

5 : https://opencv.org

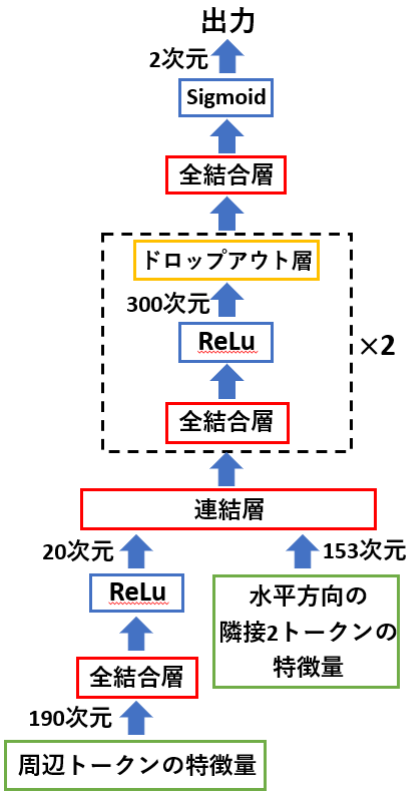

図 6　水平結合のモデル図 [1]

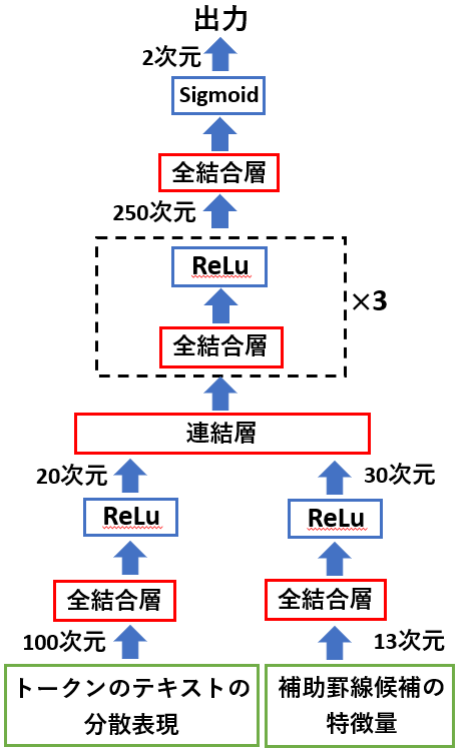

図 7　補助罫線推定のモデル図 [1]

のトークンの特徴量を用いて 2 トークンを結合する．水平結合のモデル図を図 6 に示す．水平方向に隣接している 2 トークンの 153 次元の特徴ベクトルとその周辺のトークンの 190 次元の特徴ベクトルを入力とし，それらの 2 トークンを結合するか否かの 2 次元のベクトルを出力とする．使用する活性化関数は，出力層には Sigmoid 関数を用い，それ以外の層には ReLu 関数を用いる．中間層の出力次元数は，周辺トークンの特徴量を入力とする層は 20 次元，連結層より後の層は 300 次元とする．損失関数は 2 値クロスエントロピーを用いる．また，最適化関数には Adam [6] を用い，学習率は 0.01 とする．ドロップアウト層の不活性化確率は 0.2 とする．

水平結合の対象となる隣接 2 トークンの特徴量を表 1 にまとめる．水平方向に隣接する 2 つのトークンのうち左のトークンをトークン A，右のトークンをトークン B としているため，トークン A，B 間の距離は，x 軸方向におけるトークン A の右端の座標とトークン B の左端の座標の差の絶対値である．トークン中のテキストが数値かどうかは，テキストが 0-9 の数字と，“.”，“-”，“%”，“$”，また，“greater”，“smaller”，“more”，“less” といった数値に関連する単語で構成されていれば，トークン中のテキストを数値と判断する．トークンのテキストの品詞は，MeCab によって取得した 70 次元の one-hot ベクトルである．[1] では表 1 に示したトークンの特徴量の 1 つである，トークンのテキスト (英語) の品詞を，Natural Language Toolkit (NLTK)[6] を用いて取得しているが，日本語を含む表を対象とする本稿では，MeCab を用いて取得する．MeCab は日本語の形態素解析エンジンであり，日本語に関して，NLTK よりも細かく品詞を判定することができる．具体的には NLTK で分類される品詞は 47 種類であるが，MeCab では 70 種類となる．なお，MeCab では IPAdic の辞書を利用する．また，MeCab において英語アルファベットはすべて名詞として判定される．

隣接 2 トークンの周辺トークンの特徴量を表 2 に示す．まず周辺トークンとは，水平方向に隣接するトークン A，B の上下左右の隣接 4 トークンと，トークン A，B 自身を含めた 10 トークンのことである．表 2 に示す個々の周辺トークンの特徴量は，トークンの座標，幅，高さ，テキストが数値か数値でないかの 5 次元ベクトルである．よって周辺トークンの特徴量は 5 次元 ×10 トークンの 50 次元の特徴ベクトルに，トークン A，B のそれぞれのテキストの品詞情報 70 次元 ×2 トークンの 140 次元の特徴ベクトルをあわせた合計 190 次元の特徴ベクトルとなる．

3.3.3　補助罫線推定

図 1 に示すように，トークンの水平結合の後で補助罫線を推定する．補助罫線を推定する NN モジュールのモデル図を図 7 に示す．この NN モジュールは，トークンの周囲の点の集合である補助罫線候補の 13 次元の特徴ベクトルと，補助罫線候補に関係するトークンのテキストの分散表現の 100 次元の特徴ベクトルを入力とし，補助罫線であるか否かの 2 次元のベクトルを出力とする．使用する活性化関数は，出力層には Sigmoid 関数を用い，それ以外の層には ReLu 関数を用いる．中間層の出力次元数は，トークンのテキストの分散表現を入力とする層は 20 次元，補助罫線候補の特徴量を入力とする層は 30 次元，

6：https://www.nltk.org

表 1　水平方向の隣接 2 トークンの特徴量

特徴	次元数	説明
トークン A，B 間の距離	1	トークン A の右端とトークン B の左端の距離
トークン A とトークン B のフォントの一致	1	0:不一致，1:一致
トークン A とトークン B のスタイルの一致	1	0:不一致，1:一致
トークン A のフォントサイズ	1	-
トークン B のフォントサイズ	1	-
トークン A のテキストは数値か	1	0:数値でない，1:数値
トークン B のテキストは数値か	1	0:数値でない，1:数値
結合位置	2	トークン A，B の中点の座標
表のサイズ	2	表の幅と高さ
トークンが属する列，行のトークン数	2	トークンの上下左右の延長上にあるトークンの数
トークン A のテキストの品詞	70	-
トークン B のテキストの品詞	70	-
合計	153	-

表 2　個々の周辺トークンの特徴量 [1]

特徴	次元数
座標	2
幅	1
高さ	1
テキストが数値か否か	1
合計	5

図 8　トークンの上下左右の端点と重心の中点

表 3　補助罫線候補の特徴量 [1]

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に罫線があるか否か	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否か	1
補助罫線候補の位置	1
表のサイズ	1
合計	13

連結層より後の層は 250 次元とする．損失関数は 2 値クロスエントロピーを用いる．また，最適化関数には Adam [6] を用い，学習率は 0.01 とする．

補助罫線候補は，垂直方向の補助罫線の場合，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の 3 種類の点集合である．水平方向の補助罫線候補は，トークンの上端，下端，垂直方向に隣接するトークンの重心の中点の 3 種類の点集合である．図 8 にトークンの上下左右の端点と水平方向に隣接している 2 トークンの重心の中点を示す．トークンの上下左右の端点と中点の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ている． 補助罫線候補の特徴量を表 3 に示す．クラスタの種類は，補助罫線候補がトークンの右端，左端，上端，下端，水平方向の重心の中点，垂直方向の重心の中点の 6 種類のいずれかを表す 6 次元の one-hot ベクトルである．

トークンのテキストの分散表現の取得に, [1] では Word2vec [7] を使用していたのに対して，提案手法は Word2vec の日本語学習モデルである Wikipedia2Vec を使用する．

3.3.4　垂 直 結 合

図 1 に示すように，補助罫線推定の後に，トークンの垂直結合を行う．トークンの垂直結合を行う NN モジュールのモデル図を図 9 に示す．なおこのモデルは，3.3.5 項で説明するセル生成で使用するモデルと同じである．垂直方向に隣接している 2 トークンの 165 次元の特徴ベクトルとその周辺のトークンの 190 次元の特徴ベクトルを入力とし，垂直結合を行うか否かの 2 次元のベクトルを出力する．使用する活性化関数は，出力層には Sigmoid 関数を用い，それ以外の層には ReLu 関数を用いる．中間層の出力次元数は，周辺トークンの特徴量を入力とする層は 20 次元，連結層より後の層は 200 次元とする．損失関数は 2 値クロスエントロピーを用いる．また，最適化関数には Adam [6] を用い，学習率は 0.01 とする．ドロップアウト層の不活性化確率は 0.2 とする．

隣接 2 トークンの特徴量を表 4 にまとめる．水平方向もしくは垂直方向に隣接する 2 つのトークンをトークン A，トークン B としている．水平結合と同様に，トークンのテキストの品詞の取得方法を NLTK から MeCab に変更する．セパレータは，罫線，罫線の延長線，トークンの上下左右それぞれの端点の集合から作成した 4 種類の補助罫線，水平方向，垂直方向に隣接した 2 トークンの重心の中点の集合から作成した 2 種類の補助罫線，隣接する 2 トークンが数値である場合のその間の補助罫線の合計 9 種類ある．また，周辺のトークンの特徴量は水平結合と同じく，190 次元の特徴ベクトルである．

3.3.5　セ ル 生 成

図 1 に示すように，垂直結合の後にセル生成を行う．セル生成のモデル図は図 9 の垂直結合のモデル図と同じである．セル生成では，隣接トークン A，B の水平方向の結合と垂直方向の結合を，結合ができなくなるまで交互に繰り返す．入力特徴量も表 4 と表 2 に示すものであるが，水平方向の結合の場合，トークン A は左，トークン B は右のトークンを，垂直方向の結合の場合，トークン A は上，トークン B は下のトークンを

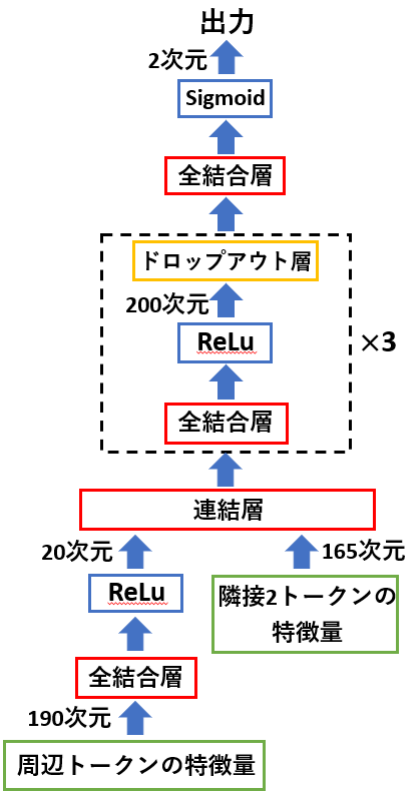

図 9　垂直結合とセル生成のモデル図 [1]

表 4　垂直結合とセル生成で利用する隣接 2 トークンの特徴量

特徴	次元数
トークン A，B 間の距離	1
トークン A とトークン B のフォントの一致	1
トークン A とトークン B のスタイルの一致	1
トークン A のフォントサイズ	1
トークン B のフォントサイズ	1
トークン A のテキストは数値か	1
トークン B のテキストは数値か	1
結合位置	2
表のサイズ	2
トークンが属する列，行のトークン数	2
結合の方向	2
間に存在するセパレータを構成する点の数	9
トークン A のテキストの品詞	70
トークン B のテキストの品詞	70
トークン A とトークン B のテキストの一致度合い	1
合計	165

表す．なお，図 1 にある後処理は, [1] と同じ処理である．

4　実　　験

4.1　実験の概要

実験では [1] の手法と本稿の手法の 2 つの手法で表構造を解析してその精度を評価する．日本語を含む表の構造解析とその精度評価を行うため，DEIM2022 の論文中の日本語を含む 280 の表に表構造情報を付与した．この作成した表データを，4 つ

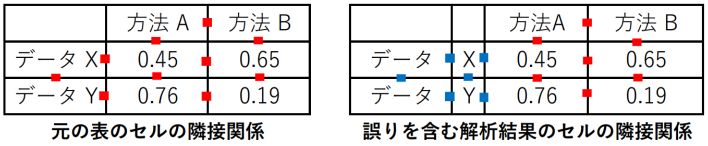

図 10　表のセルの隣接関係

表 5　表構造解析結果

手法	再現率	適合率	F 値
[1] の手法	0.825	0.908	0.865
提案手法	**0.834**	**0.938**	**0.883**

表 6　水平結合の結果：[1] の手法

		推定	
		結合する	結合しない
正解	結合する	444	4
	結合しない	88	1561

表 7　水平結合の結果：提案手法

		推定	
		結合する	結合しない
正解	結合する	432	16
	結合しない	53	1,596

の NN モジュールの学習用に 230 表，精度評価用に 50 表に無作為に分け，表構造解析の実験を行う．

表構造解析結果の評価には，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 [8] を用いる．図 10 にセルの隣接関係の例を示す．図中の赤い四角で示された箇所は正しい隣接関係を，青い四角で示された箇所は間違った隣接関係を表している．セルの隣接関係の再現率，適合率は次の式で算出される．また F 値は再現率と適合率の調和平均である．

$$\text{再現率} = \frac{\text{解析結果の表の正しい隣接関係の数}}{\text{元の表の隣接関係の数}} \quad (1)$$

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

4.2　実 験 結 果

4.2.1　表構造解析精度

表構造解析の実験結果を表 5 に示す。提案手法の再現率は 0.834，適合率は 0.938，F 値は 0.883 となり, [1] の手法をそれぞれ 0.9，3.0，1.8 ポイント上回った．

4.2.2　水平結合の結果

[1] の手法で隣接 2 トークンの水平結合を行った場合の結合結果を表 6 に示す．結合するべきなのに結合されなかったペアが 4，結合するべきでないのに結合したペアが 88 あった．全体としての精度は 95.6%であった．また，提案手法で隣接 2 トークンの水平結合を行った場合の結合結果を表 7 に示す．結合するべきなのに結合されなかったペアが 16，結合するべきでないのに結合したペアが 53 あった．全体としての精度は 96.7%であった．

表 8　補助罫線推定の結果：[1] の手法

		推定	
		補助罫線	補助罫線でない
正解	補助罫線	789	95
	補助罫線でない	408	2,233

表 9　補助罫線推定の結果：提案手法

		推定	
		補助罫線	補助罫線でない
正解	補助罫線	704	185
	補助罫線でない	273	2,388

表 10　セル生成の結果：[1] の手法

		推定	
		結合する	結合しない
正解	結合する	38	33
	結合しない	74	3,079

表 11　セル生成の結果：提案手法

		推定	
		結合する	結合しない
正解	結合する	35	42
	結合しない	41	3,120

4.2.3　補助罫線推定の結果

[1] の手法で補助罫線推定を行った結果を表 8 に示す．補助罫線を誤って補助罫線でないと推定する誤りは 95，補助罫線でないものを誤って補助罫線と推定する誤りは 408 あった．全体としての精度は 85.7%であった．また，提案手法で補助罫線推定を行った結果を表 9 に示す．補助罫線を誤って補助罫線でないと推定する誤りは 185，補助罫線でないものを誤って補助罫線と推定する誤りは 273 あった．全体としての精度は 87.1%であった．

4.2.4　セル生成の結果

[1] の手法で隣接 2 トークンの結合すなわちセル生成を行った結果を表 10 に示す．結合するべきなのに，結合しなかった隣接 2 トークンは 33，結合するべきではないが誤って結合した隣接 2 トークンは 74 あった．全体としての精度は 96.7%であった．また，提案手法で隣接 2 トークンの結合を行った結果を表 11 に示す．結合するべきなのに，結合しなかった隣接 2 トークンは 42，結合するべきではないが誤って結合した隣接 2 トークンは 41 あった．全体としての精度は 97.4%であった．

5　考　　察

5.1　表構造解析の誤り

空白によって大きなセルが過分割された表の例を図 11 に，その解析結果を図 12 に示す．図 11 の赤い四角で示された箇所に，入力の PDF ファイルを変換した XML ファイルには U+3000，いわゆる全角空白文字があり，全角空白文字とその右の”冬”を含む 5 行目が，2 行目から 4 行目とは独立していると解析された．元の表では “ライトアップ”，“桜”，“夜”のセルは 2 行目から 5 行目にまたがるためこれは誤りである．同様に，“無料”，“早朝”，“開放”が含まれる 6 行目から 9 行目においても，9 行目が独立した行として解析されている．

クラス	トピック	季節	レビュー数	正解数	適合率
1	ライトアップ 桜 夜 □	春	304	276	0.908
		夏	51	33	0.647
		秋	74	68	0.919
		冬	152	113	0.743
5	無料 早朝 開放 □	春	36	29	0.806
		夏	39	32	0.821
		秋	35	31	0.886
		冬	64	51	0.797

図 11　空白によってセルが過分割された表の例 [9]

クラス	トピック	季節	レビュー数	正解数	適合率
1	ライトアップ 桜 夜	春	304	276	0.908
		夏	51	33	0.647
		秋	74	68	0.919
	□	冬	152	113	0.743
5	無料 早朝 開放	春	36	29	0.806
		夏	39	32	0.821
		秋	35	31	0.886
	□	冬	64	51	0.797

図 12　図 11 の表の解析結果

小室圭	ネット侮辱罪	アフガン撤退	ワクチン 3回目接種	10万円分 クーポン配布
7人	6人	5人	4人	8人

図 13　[1] の手法でのみ誤って水平結合した表の例 [10]

日本語を含む表の XML ファイルでは，全角空白文字があった場合これがトークンとして検出され，トークンの結合や補助罫線の推定に影響する．全角空白文字がトークンとして検出される表は他にも見つかっており，全角空白文字を取り除いて解析を行うことによる推定精度改善が期待できる．

水平結合において [1] の手法では隣接 2 トークンを誤って結合したが，提案手法は正しく解析できた表の例を図 13 に示す．[1] の手法は図 13 の赤い四角で示された 2 つのトークンを誤って結合したが，提案手法は結合しなかった．

図 14 に垂直結合において誤って結合された表の例を示す．図 14 で赤い四角で示された 3 つのトークンが誤って結合された．これは “short” と “long” の 2 つのトークンがまず誤って水平結合され，“推定ラベル”のトークンが “short long” のトークンと誤って垂直結合された．

5.2　日英の言語の違いの影響

表 12 に英語の 156 表からなる ICDAR 2013 table competition のテスト用データセットに対して，[1] の手法で水平結合を

表 12　ICDAR 2013 table competition のテスト用データセットに対する [1] の手法の水平結合の結果 [1]

		推定	
		結合する	結合しない
正解	結合する	3,031	8
	結合しない	19	66

		推定ラベル		
		short	long	再現率
正解ラベル	short	2364	636	0.788
	long	1911	1089	0.363
適合率		0.533	0.631	0.576

図 14　誤って垂直結合した表の例 [11]

		0~20%
強制 UI	肯定記事	0
	否定記事	2
コスト UI	肯定記事	0
	否定記事	13

図 15　日本語＋英語のトークンの組み合わせの例 [12]

英語	Python
Python	擬似コード (英/日)
Java	コメント (英)
英語	LOGO プログラム
日本語	SDC(意味構造)
日本語	Python

図 16　英語＋日本語のトークンの組み合わせの例 [13]

行った結果を示す．これを表 6 に示した日本語を含む表に対する結果と比較すると，英語の表のほうが日本語を含む表と比べて水平結合すべきトークンが多い．日本語であれば 1 つのトークンとなるテキストを，英語では複数のトークンで表すことが多いためこのような結果になった．

また，図 14 のような日本語を含む表においては，英語は “short” と “long” のように結合されずに独立したトークンとして使われていることが英語の表よりも多いため，水平結合するトークンが少ないという理由も考えられる．また，水平結合されるトークンには，図 15 の青い四角で示した箇所のような “強制 UI” や “コスト UI” といった日本語＋英語もしくは図 16 の青い四角で示した箇所のような “LOGO プログラム” といった英語＋日本語のような使われ方がみられた．

6　おわりに

本稿では，日本語を含む表を対象とする，NN を用いた表構造解析手法を提案した．提案手法は，青柳らが英語の表を構造解析するために提案した手法 [1] に基づいており，まず入力された表の PDF を XML に変換し，検出したトークンの座標，大きさ，フォントなどの情報をもとに水平方向に隣接した 2 つのトークンを結合するか推定する．次に水平結合したトークンや周辺のトークンの特徴量を用いて補助罫線を推定する．その後垂直方向に隣接した 2 つのトークンをトークンの特徴量や補助罫線の情報を入力として結合するか推定する．そして垂直に結合されたトークンをさらに隣接するトークンと結合することでセルを生成する．本研究では，水平結合や垂直結合，セル生成の NN の入力とするトークンのテキストの品詞情報を，日本語形態素解析解析システム MeCab により獲得するよう改良した．また，補助罫線推定の NN の入力とするトークンの分散表現も，Word2vec の日本語学習モデルである Wikipedia2Vec を用いて獲得するよう改めた．

また実験のため，DEIM2022 の論文中の日本語を含む 280 の表に表構造情報を付与したデータを作成し，各 NN モジュールの学習，精度評価に利用した．提案手法の表構造解析精度は，セルの隣接関係の再現性に基づく評価指標において再現率が 0.834，適合率が 0.938，F 値が 0.883 となった．この結果は青柳らの手法をそれぞれ 0.9，3.0，1.8 ポイント上回った．水平結合，補助罫線推定，セル生成のモジュールごとの評価結果も，それぞれ精度が 96.7%，87.1%，97.4%となり，青柳らの手法をそれぞれ 1.1，1.4，0.7 ポイント上回った．日本語を含む表と英語の表の構造解析における違いとして，日本語は英語が複数のトークンを使って表す内容を 1 つのトークンで表すことが多いため，日本語を含む表では水平結合すべきトークンが英語の表に比べてかなり少ないことがわかった．また，日本語を含む表では全角空白文字のトークンが空のセルを形成し，その結果表構造に誤りが生じる事例があった．

今後の課題として，日英の言語の要素以外の表の違いの分析が挙げられる．

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)，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] 青柳拓志，金澤輝一，高須淳宏，上野史，太田学，“ニューラルネットワークを用いた表構造解析の一手法，” 第 13 回データ工学と情報マネジメントに関するフォーラム (DEIM2021)，E25-3，2021.
[2] 山田凌也，太田学，金澤輝一，高須淳宏，“機械学習を用いた表構造解析の一手法，” 第 12 回データ工学と情報マネジメントに関するフォーラム (DEIM2020)，E6-4，2020.
[3] M. Göbel et al. ICDAR 2013 Table Competition. ICDAR 2013, pp. 1449-1453, 2013.
[4] 樋口亮太，山西良典，松下光範，“単語の頻度と意味に基づいたコミックに関するテキスト情報源の特性分析，” 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022)，E21-2，2022.
[5] 神田凌弥，杉山一成，吉川正俊，“リツイートの時系列データを用いた Hierarchical Attention Networks に基づく誤情報検出，” 第 14 回データ工学と情報マネジメントに関するフォーラ

ム (DEIM2022)，B34-6，2022.
[6] D. P. Kingma and J. Ba. Adam，“A method for stochastic optimization,” In Proceedings of International Conference on Learning Representations (ICLR), 2015.
[7] T. Mikolov et al.，“Efficient estimation of word representations in vector space,” CoRR. vol. abs/1301.3781, 2013.
[8] M. Göbel, E. Oro, and G. Orsi，“A methodology for evaluating algorithms for table understanding in PDF documents,” In Proceedings of the ACM Symposium on Document Engineering 2012, pp. 45-48, 2012.
[9] 鳥山実桜，灘本明代，“観光レビューから季節特有な情報と季節によって変化する情報の抽出手法の提案,” 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022)，B43-1，2022.
[10] 三倉温樹，荘司慶行，Martin J. Dürst，“言及箇所と賛否に注目したソーシャルメディア上でのニュース信憑性判断支援,” 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022)，B34-3，2022.
[11] 岡田渓，新田直子，中村和晃，馬場口登，“気象センサ情報を用いた屋外画像における人物の服装変換,” 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022)，C31-2，2022.
[12] 益田匡史，北山大輔，“情報偏食の軽減における検索結果 UI の評価,” 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022)，I44-13，2022.
[13] 小原百々雅，秋信有花，梶浦照乃，倉光君郎，“リアルタイムコード翻訳によるプログラミング学習支援 AI に向けて,” 第 14 回データ工学と情報マネジメントに関するフォーラム (DEIM2022)，H41-4，2022.
[14] H. Aoyagi，T. Kanazawa，A. Takasu，F. Uwano，and M. Ohta，“Table-structure Recognition Method Consisting of Plural Neural Network Modules,” 11th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2022)，pp. 542-549，2022.