機械学習を用いた表構造解析の一手法

山田　凌也† 　太田　学† 　金澤　輝一†† 　高須　淳宏††

† 岡山大学大学院自然科学研究科　〒 700–8530 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

E-mail: †yamada@de.cs.okayama-u.ac.jp, ohta@cs.okayama-u.ac.jp, ††{tkana,takasu}@nii.ac.jp

あらまし　正確な数値を読み解くのに適している表は学術論文等において実験結果を表すのに頻繁に用いられる．しかし，数値の比較や変化を視覚的に読み取るにはグラフの方が適しているため，我々は表データからのグラフの自動生成を提案した．本稿では，グラフを自動生成するために機械学習を用いた表構造解析手法を提案する．表の構造解析では，表中の罫線は非常に重要であるが，セル間に罫線が必ずしも引かれているとは限らない．そこで機械学習を用いて引かれていない罫線を補完し，さらに表中のトークンを結合することでセルを生成する．実験では，ICDAR2013 において提供されたデータセットを用い，その精度を評価する．結果として F 値が 0.955 となり，比較手法をすべて上回った．

キーワード　表構造解析，グラフ自動生成，XML

1 はじめに

近年，CiNii[1]や Google Scholar[2]等の学術論文データベースの充実により，膨大な論文を手軽に入手できるようになった．それゆえ，膨大な論文を活用するためには入手した論文の内容や実験結果を効率的に把握する必要があり，学術論文の自動要約やキーワード抽出などの研究が盛んに行われている [1,2]．とりわけ，実験結果などの情報は表やグラフにまとめられている．Kamola ら [3] は論文から情報抽出する前処理として論文構成要素を分析，分類した．論文構成要素の中でも，表は効率的かつ手軽に数値情報を示すことができ，文書解析の著名な国際会議である International Conference on Document Analysis and Recognition (ICDAR) 2013 では表構造解析コンペティション [4] が開催された．

Nurminen [5] は同コンペティションの中で，空白の検出とエッジと呼ばれる線分に基づく表構造解析手法を提案した．エッジとは行の左端，右端，重心が同一垂直線上に 4 つ以上ある時，そこに引く線分である．Nurminen らはこのエッジを行や列の名前を表すヘッダを検出するために利用した．評価実験では，同コンペティションで提案されたセル間の隣接関係に基づく評価指標において再現率が 0.941，適合率が 0.952，F 値が 0.946 を達成し，これは参加者中最高の成績であった．後続のコンペティションは，解析対象を表が含まれる文書 PDF から表が含まれる画像へと変え，ICDAR2019[3]においても開催された．

一方，実験結果を表現する方法としてグラフもしばしば用いられる．グラフは数値の比較や変化を視覚的に把握するのに適している．そのため，我々は効率的に内容を把握するにはグラフの方が適していると考え，表構造解析に基づくグラフの自動生成手法 [6] を提案した．しかし，表の罫線の引き方やセルの構成など表の構造は著者によって異なる．特に，罫線は表の構造解析にあたって非常に重要な情報となるが，罫線を引くのか，それとも罫線でなくセルの配置によって行，列を表現するのかは著者による．そのため，我々は Nurminen の提案したエッジを行ではなく，単語であるトークンを用いて作成し，表中の単語を結合していくことでセルを生成する表構造解析手法を提案した [7]．この実験では ICDAR2013 Table Competition において提供されたデータと評価指標を用いて表構造を解析し，F 値が 0.912 となった．これは同コンペティション参加者のトップである Nurminen の結果に迫るものであった．

しかし，DocEng2019 [7] で提案した表構造解析手法ではエッジの作成やトークンの結合にヒューリスティクスを用いており，拡張性や汎用性が乏しかった．また，誤ったエッジを作成してしまうなど，ヒューリスティックスのみでは対応できない表もあり，より柔軟な手法が必要であった．そこで本稿では，機械学習を用いた汎用的な表構造解析手法を提案する．具体的には，引かれていないがセルを分割するために必要な線分である補助罫線の予測とトークンの結合を行うニューラルネットワーク (NN) モデルを含む表構造解析手法を提案する．

本稿の構成は次の通りである．まず，2 節で表構造解析に関する研究を紹介する．次に 3 節で本稿で提案する表構造解析手法とそれに含まれる 2 つのモデルについて述べる．4 節ではそれぞれのモデルの性能と表構造解析結果を評価する．続く 5 節で実験結果について考察する．最後に 6 節でまとめる．

2 関連研究

Nurminen [5] は ICDAR2013 において PDF 中の表の構造解析のアルゴリズムを提案した．まず，Poppler PDF rending library[4]を利用し，PDF 中のテキストデータをテキストボック

1 : https://ci.nii.ac.jp/

2 : https://scholar.google.co.jp/

3 : http://sac.founderit.com/

4 : http://poppler.freedesktop.org/

スとして抽出し，PDF 画像から水平，垂直両方向の罫線を見つける．その後，同一 Y 座標のテキストボックスが同じ行にあるとし，行中でテキストボックス間の距離を用いて，テキストボックスのマージを繰り返す．つづいて，複数の行に対して，その左端，右端，重心が同一垂直線上にあればそれをエッジとして検出する．エッジと行間や列間の空白を基に行，列をマージしていくことで最終的な表の構造を決定した．ICDAR2013 の Table Competition において，彼らのシステムによる解析結果は，セル間の隣接関係の再現率が 0.941，適合率が 0.952，F 値が 0.946 で参加者中最高の結果であった．

Shingrov ら [8] は単語の矩形領域を用い，表構造を再構成する手法を提案した．彼らの手法は 3 段階に分けられる．最初に，前処理として PDF からテキストチャンクと罫線を生成する．次に，テキストチャンク間の距離と大きさや矩形領域の位置関係を基にテキストチャンクを結合し，テキストブロックを生成する．最後にテキストブロックを分析し，行と列に分割していくことでセルを構築し，表の構造を解析した．適切な閾値や計算方法を探索し，よりよい表構造解析手法を探索した．彼らも評価のために ICDAR2013 の Table Competition で提供されたデータセットを用い，もっとも良かった結果で再現率が 0.923，適合率が 0.950，F 値が 0.936 であった．

Chi ら [9] は graph neural network (GNN) model を用いて PDF の表の構造を解析するモデル GraphTSR を提案した．このモデルはまず前処理として，Shingrov ら [8] と同様の手法でテキストチャンクの取得，結合を行いそれをセルとした．セルをグラフの頂点とし，任意の 2 頂点間に辺を持つ完全グラフを作成した．それらの辺に対して，“vertical”，“horizontal”，“no relation” の 3 種類のタグを GraphTSR を用いて付与することにより表構造を解析した．評価実験には，データセットとして，ICDAR2013 のデータセットと彼らの作成したデータセット (SCiTSR)，その中でも複数行，列にまたがるセルを含む複雑な表のみを抽出したデーセット (SCiTSR-COMP) を用いた．その精度を ICDAR2013 における隣接関係に基づく評価指標で評価した．結果としてそれぞれ F 値のマイクロ平均がそれぞれ 0.872，0.953，0.955 となった．

3 提案する表構造解析手法

3.1 表の定義

本稿では，隣接するセルを分割する線分のうち，実際に書かれているものを罫線，書かれていないが分割するために必要なものを補助罫線と呼ぶ．また，これらをまとめてセパレータと呼ぶ．表中のセルはこのセパレータによって囲まれているものと定義する．

図 1 に本稿で扱う表の例を示す．図 1 中では罫線を実線，補助罫線を点線で示している．また，表中の単語をトークンと呼ぶ．トークンは単語のテキスト自身に加え，フォントやそれを囲む矩形領域の座標や幅，高さなどを属性として持つ．赤い矩形で囲んだ文字列がトークンである．さらに，各行の名前を表してる列を「ヘッダ列」，各列の名前を表している行を「ヘッダ行」，それ以外の数値等を「データ」と呼ぶ．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データを持たない行を「サブヘッダ行」と呼ぶ．図 1 では青色ハイライトした列がヘッダ列，緑色でハイライトした行がヘッダ行，黄色でハイライトした行がサブヘッダ行である．

	method A	method B	method C	method D
CorpusA				
a1	0.839	0.876	0.814	0.815
a2	0.902	0.769	0.858	0.883
CorpusB				
b1	0.741	0.955	0.746	0.843
b2	0.734	0.950	0.966	0.891
b3	0.935	0.930	0.979	0.917

図 1: 表の例

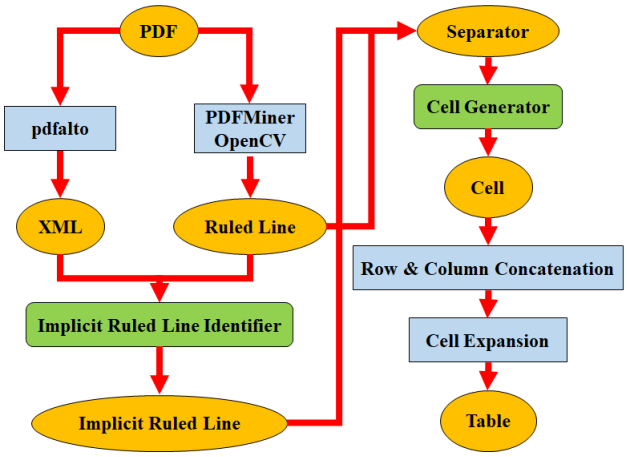


図 2: 表構造解析手法の概略図

3.2 表構造解析手法の概要

本稿で提案する機械学習を用いた表構造解析手法について述べる．本手法の概要を図 2 に示す．まず PDF から表を抽出するために pdfalto[5]により文書 PDF を XML ファイルへと変換する．この XML ファイルには各単語がトークンとして，ページの左上から順に並んでおり，そのトークンを囲む矩形領域の左上の座標や幅，高さ，フォントなどが共に記されている．本稿では表の範囲は人手で定める．また，罫線の抽出は PDFMiner.six[6]による抽出と opencv-python[7]による直線検出によって抽出する．

Implicit Ruled Line Identifier はトークンの位置情報と罫線の特徴を用いて補助罫線を推定する．ここで推定された補助罫線と罫線をまとめたものがセパレータである．つづいて，Cell Generator で隣接する 2 トークンを再帰的に結合することによりセルを生成する．Cell Generator はセパレータとトークンの特徴を入力とする．また，特徴量としての座標はすべて表の左上を原点とする．

セル生成後にルールによって行や列を結合し，複数や複数列

5 : https://github.com/kermitt2/pdfalto
6 : https://github.com/pdfminer/pdfminer.six
7 : https://opencv.org/

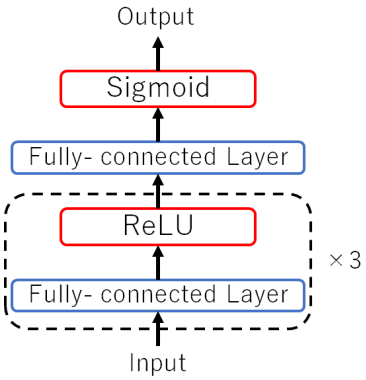


図 3: Implicit Ruled Line Identifier のモデル

図 4: トークンの上下左右の端点と重心の中点

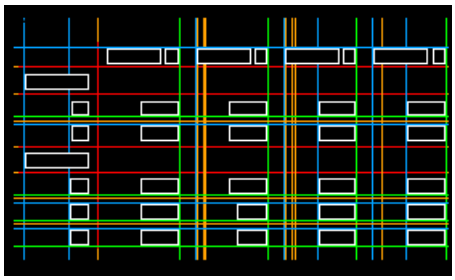


図 5: 補助罫線推定結果

にまたがるセルはそのセルの文字揃えに基づくルールによってその範囲を拡張する．

3.3 補助罫線推定

3.3.1 補助罫線

本稿では 3 種類の補助罫線を定義する．

- トークンの位置関係に基づく補助罫線
- 垂直方向に隣接した数値間の補助罫線
- 罫線の延長線

まず，トークンの位置関係に基づく補助罫線について述べる．我々は補助罫線の推定のためにトークンの文字揃えが重要であると考えた．例えば，図 1 では，“method A” の “A” とそれより下のセルの右端が揃っているため，そこに補助罫線が存在すると推測できる．

2 つ目は垂直方向に隣接した数値間の補助罫線である．我々は 1 つのセル中で数値が垂直方向に隣接することはないと仮定した．そこで，垂直に数値が並んでいる場合はその間に必ず補助罫線があるとする．例えば，図 1 においては，3 行 2 列目の “0.839” と 4 行 2 列目の “0.902” は双方とも数値であり，垂直に隣接している．そのため，この 2 トークンは結合することはなく，その間には補助罫線が存在する．

最後に罫線の延長線について説明する．表の一部，例えばヘッダ行中にのみ罫線を引いている表が存在する．そのような表に対応するために罫線の延長線もすべて補助罫線として用いる．本稿では，1 つ目のトークンの位置関係に基づく補助罫線を 3.3.2 節で説明する Implicit Ruled Line Identifier によって推定する．

3.3.2 Implicit Ruled Line Identifier

Implicit Ruled Line Identifier はトークンから得た特徴を用いて，トークンの位置関係に基づく補助罫線を推定するためのモデルである．図 3 に Implicit Ruled Line Identifier のモデル概要を示す．本モデルの中間層は 3 層の全結合層からなり，入力は補助罫線候補の 13 次元の特徴ベクトル，出力は補助罫線候補が “補助罫線” か “非補助罫線” かの 2 次元である．中間層の出力次元は 30 次元とする．出力層の活性化関数は Sigmoid 関数，それ以外の層には 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て，2 値分類を行う．最適化関数は Adam，学習率は 0.01 とした．これらのパラメータは学習データを用いた 5 分割交差検証によって決定した．パラメータの探索範囲は，中間層の出力次元が 20，30，50 とドロップアウト層の有無である．

3.3.3 Implicit Ruled Line Identifier の入力特徴量

補助罫線を引く場所の候補作成のために，垂直方向の補助罫線はトークンの左端，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はトークンの上端，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 種類ずつ合計 6 種類の点の集合を作成する．その後，それぞれの集合でクラスタリングし，各クラスタの重心を通るような線分を補助罫線の候補とする．ただし，水平方向の補助罫線を推定する場合，各集合中の点の y 座標のみを用いてクラスタリングする．一方，垂直方向の補助罫線を推定する場合は x 座標のみを用いてクラスタリングする．

クラスタリングには階層的クラスタリングの 1 つである重心法を用いて，クラスタの重心とそのクラスタ中の各点とのユークリッド距離が 1 以下のクラスタを生成する．図 4 に図 1 の左下の 2 つのトークンの上下左右の端点と，その重心の中点を示す．青色の点がトークンの左端と上端，緑色の点が右端と下端，オレンジ色の点がこの 2 トークンの重心の中点である．

表 1 に Implicit Ruled Line Identifier への入力特徴量を示す．図 1 中の “method A” の “A” の右側に引かれる補助罫線を例に表 1 の特徴量について説明する．この補助罫線候補は “A”，“0.839”,“0.902”,“0.741”,“0.734”,“0.935” の 6 つのトークンの右端によって成るクラスタの重心を通る．また，表の垂直方向のトークン数は 6 であり，この補助罫線を挟むトークン “A” とその右に位置する “method B” の “method” の間には罫線が存在せず，この補助罫線は他のセル上を通らない．よってこの補助罫線候補の入力特徴量は [点の数: 6, 垂直方向のトークン数: 6, 罫線: 0, 方向: 1, クラスタの種類 [0, 0, 0, 1, 0, 0], セル上を通るか: 0, x 座標, 幅] となる．Implicit Ruled Line Identifier はこの補助罫線の候補に対して “補助罫線” か “非補助罫線” かを推定する．

図 5 に図 1 に対して推定した補助罫線を示す．図 5 中では白色の矩形がトークンを，赤色の実線が罫線を表している．青色

表 1: Implicit Ruled Line Identifier への入力特徴量

特徴量	次元数	説明
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	水平方向の補助罫線推定では水平方向 垂直方向の補助罫線推定では垂直方向
補助罫線候補を挟むトークン間の罫線	1	0: 存在しない，1: 存在する
補助罫線候補の方向	1	0: 水平，1: 垂直
クラスタの種類	6	[下，重心の中点 (垂直)，上，右，重心の中点 (水平)，左] の one-hot ベクトル
補助罫線候補がセル上を通るか	1	0: 通らない，1: 通る
補助罫線候補の位置 (比率)	1	水平方向の補助罫線推定では y 座標 垂直方向の補助罫線推定では x 座標
表のサイズ	1	水平方向の補助罫線推定では表の高さ 垂直方向の補助罫線推定では表の幅

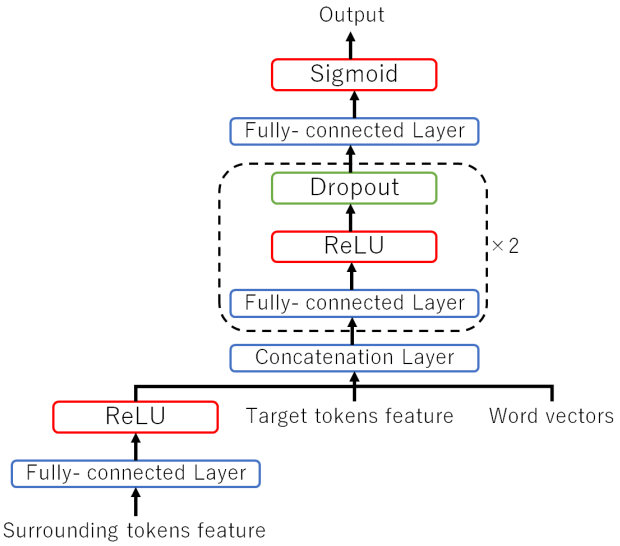


図 6: Cell Generator のモデル

の実線がトークンの左端または上端の集合から作成した補助罫線，緑色が右端または下端の集合から作成した補助罫線，オレンジ色が隣接するトークンの中点の重心の集合から作成した補助罫線である．

3.4 セルの生成

罫線，補助罫線，トークン等の情報を Cell Generator へ入力し，隣接トークンの結合を推定させる．さらに後処理を加え，セルを生成する．

3.4.1 Cell Generator

Cell Generator は 3.3 節で述べた補助罫線とトークンの特徴に基づいてセルを生成する．Cell Generator は隣接する 2 トークン A,B に対して，それらの 2 トークンが結合するかどうかを判定するモデルである．トークン A は水平方向に結合する時は左側に位置するトークンとし，垂直方向に結合するときは上側に位置するトークンとする．このモデルの概要図を図 6 に示す．このモデルの入力ベクトルは 3 つある．まず結合するかどうかを推定する隣接 2 トークンの特徴量 (Target tokens feature)，周囲のトークンの特徴量 (Surrounding tokens feature)，そしてトークン中のテキストの特徴を得るために word2vec [10, 11] を用いて得たトークン A，B のテキストの分散表現 (Word Vectors) である．入力ベクトルの大きさは，トークン A,B の特徴量が 23 次元，周囲のトークンの特徴量が 50 次元，テキストの分散表現が A, B で合わせて 200 次元である．これらを入力として，出力は隣接 2 トークンの関係が “結合” か “非結合” の 2 次元である．中間層の出力次元数は Concatenatio Layer 以前は 128 次元，以後は 200 次元とする．

活性化関数は Implicit Ruled Line Identifier と同様に出力層には Sigmoid 関数を，それ以外の層には ReLU 関数を用い，損失関数には 2 値クロスエントロピーを用いる．最適化関数は Adam，学習率は 0.001 とする．これらのパラメータは学習データを用いた 5 分割交差検証によって決定した．探索範囲は中間層の出力が 150，200，250，ドロップアウト層の不活性化確率が 0，0.2，0.5 である．

単語の分散表現を得るための word2vec のフェイクタスクには，隣接トークンの結合判定に利用するため，隣接単語を予測するタスクである Skip-gram を用いる．出力は 1 単語に対して 100 次元，学習に用いる単語の出現回数の下限値である mincount は，表中には固有名詞が含まれているため 1 とする．

3.4.2 Cell Generator への入力ベクトル

Cell Generator への入力ベクトルのうち隣接 2 トークンの特徴量を表 2 に示す．

トークン A，B 間の距離は水平方向の結合の場合，トークン A の右端とトークン B の左端の x 軸方向の距離であり，垂直方向の結合では下端と上端の y 軸方向の距離である．トークン中のテキストはそれが 0-9 の数字と “.”，“-”，“%”，“$”，数値に関連する単語である “greater”，“smaller”，“more”，“less” で構成されていれば，数値と判定する．トークンが属する行や列のトークン数は，その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．

本稿ではセパレータを構成するクラスタが大きいほどその補助罫線は有力と考え，セパレータの種類ごとのクラスタ中の点の数を特徴量として用いる．なお罫線とその延長線はそれを構成する点が存在しないため，構成する点の数を 1 とする．トー

表 2: 隣接 2 トークンの特徴量

特徴量	次元数	説明
トークン A，B 間の距離	1	-
トークン A，B のテキストのフォントの一致	1	0: 不一致，1: 一致
トークン A，B のテキストのスタイルの一致	1	0: 不一致，1: 一致
トークン A のテキストのフォントサイズ	1	-
トークン B のテキストのフォントサイズ	1	-
トークン A のテキストが数値か	1	0: 数値ではない，1: 数値
トークン B のテキストが数値か	1	0: 数値ではない，1: 数値
結合位置	2	トークン A,B 間の中点の座標 [x, y]
表のサイズ	2	[幅，高さ]
トークンが属する列，行のトークン数	2	[列のトークン数，行のトークン数]
結合の方向	1	0: 水平，1: 垂直
間に存在するセパレータを構成する点の数	9	[罫線，罫線の延長線，右，重心の中点，左，下，重心の中点，上，数値間]

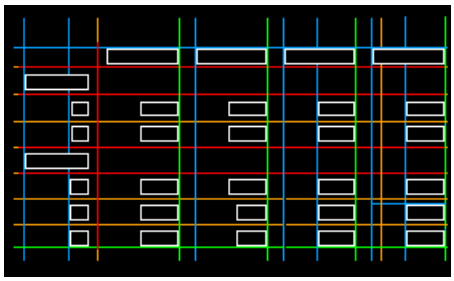


図 7: セル生成結果

クン間に同種類のセパレータが複数ある場合はその中で最大のクラスタの点の数をセパレータを構成する点の数とする．

周囲のトークンの特徴量はトークンの矩形の左上の座標と幅，高さ，テキストが数値であるか否かを並べたものである．周囲のトークンとしてトークン A，B のそれぞれで上下左右の隣接 4 トークンとそれ自身で 5 トークン× 2 =10 トークンを用いた．

word2vec の学習には，Leipzig Corpora[8] より English News(2016) と表を含む文章 PDF を pdfalto によってテキスト化したものを用いる．

3.4.3 Cell Generator の後処理

Cell Generator の推定結果に基づきトークンを再帰的に結合することで，セルを生成する．この時，間に罫線がある場合は結合を行わない．また，[7] において提案したルールのうち，以下の 2 つを用いてトークン間の補助罫線を整理する．

- 補助罫線と罫線が両方存在する場合，罫線のみを残す
- 補助罫線間の距離が 1 文字分以下ならば 1 本にまとめる

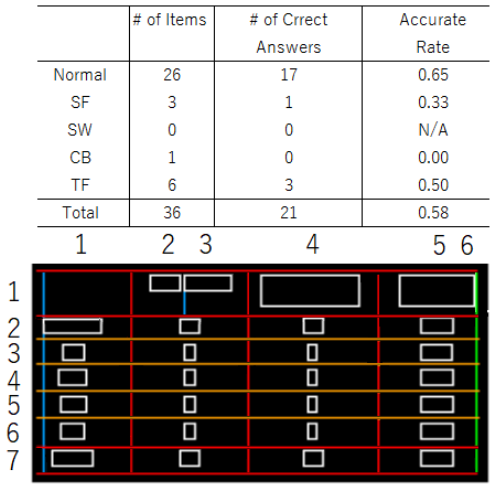
Cell Generator の結果に基づくトークンの結合と補助罫線の整理を行った結果を図 7 に示す．補助罫線の整理後，それに基づきセルが何行，何列に位置するか同定する．

3.5 行や列の結合

Cell Generator により生成したセルを以下のルールにより修正する．

（1） データを共有する行や列は結合する

8 : http://wortschatz.uni-leipzig.de/en/download/



	# of Items	# of Crrect Answers	Accurate Rate
Normal	26	17	0.65
SF	3	1	0.33
SW	0	0	N/A
CB	1	0	0.00
TF	6	3	0.50
Total	36	21	0.58

図 8: データが同一な列を持つ表 [12] とその解析結果

（2） ヘッダを共有する行や列は結合する

（3） ヘッダ部分にまたがるセルがあった場合，その範囲が 1 行や 1 列となるように行や列を結合する

本稿ではヘッダ行とヘッダ列はそれぞれ 1 行目と 1 列目から数えて数値が出てきた行や列の直前までとする．ただし，数値を含まない表では 1 行目と 1 列目をヘッダ行とヘッダ列とする．

まず，図 8 の表を例にルール (1) を説明する．図 8 の上が解析する表であり，下が解析結果である．解析結果の上の番号は列番号を表している．図 8 を見ると，“# of Items” が 2 列目と 3 列目に過分割されている．この時，この 2 列は 2 行目以降のセル，“26” から “36” を共有している．よって，これら 2 列を結合し 1 つの列とする．

一方，ヘッダ部分を共有している場合も，行や列を 1 つの行や列に結合する．

ルール (3) は過分割されたヘッダを修正するためのルールである．図 9 にこのルールを適用する表の一部とその解析結果を示す．上が元の表の一部であり，下が解析結果である．この表では 1 行のヘッダ行が過分割され 3 行となっており，この 3 行にまたがるセルが存在している．我々はヘッダ行中には複数行のセルは少ないと考え，これらのセルの範囲が 1 行となるよう

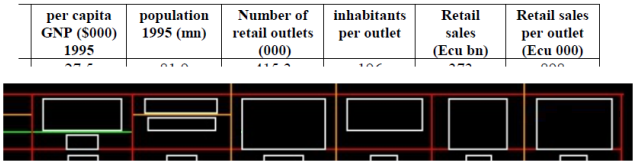


図 9: ヘッダ行にまたがるセルを持つ表とその解析結果

にこの 3 行を結合する．

3.6 セルの拡張

次に DocEng2019 [7] で提案したルールを基に複数行や複数列にまたがるセルの範囲を拡張する．

まず，サブヘッダ行の拡張について説明する．[7] では，サブヘッダ行のセルも他のセル同様に文字揃えに応じて拡張した．しかし，サブヘッダ行のセルはその位置によって修飾する対象が異なる．例えば，図 1 中のサブヘッダ行のセルである “CorpusA” と “CorpusB” はヘッダ列に属するセルであり，それより下のヘッダ列のセルを修飾している．よって，その範囲はヘッダ列の範囲である 1 列目のみとする．

サブヘッダ行以外のセルでは，[7] のルールを基にしたルールでセルの範囲を拡張する．まず，拡張範囲を求める．セルの範囲の拡張は，拡張対象のセルの周囲の空白のセルを埋めるように行う．しかし，複数行や複数列にまたがるセルでは，その範囲が下や右の罫線の長さによって表現されていることがある．これを利用するため，拡張範囲として周囲の空白と罫線の両方を用いる．つづいて，セルの範囲を拡張する．拡張方法は拡張対象のセルの文字揃えによる．例えば，水平方向の拡張においては，拡張対象のセルの下のセルと左端，右端を比較し，左端が一致していれば左揃え，右端が一致してれば右揃え，両端のどちらも一致，もしくはどちらも一致していなければ中央揃えとした．垂直方向の拡張では，拡張対象の右側のセルと上端，下端を比較する．本手法では水平，垂直双方向の文字揃えを 1 つではなく，取りうる文字揃えすべてで中央揃え，右 (下) 揃え，左 (上) 揃え拡張を試み，拡張できた拡張方法を採用する．具体的な拡張方法は，中央揃え以外では，拡張対象のセルと同一列や同一行にあるヘッダ行やヘッダ列のセルの範囲に合わせて拡張し，中央揃えでは隣接する行や列の同範囲のセルの重心と拡張対象のセルの重心が一致するように拡張する．

4 評価実験

4.1 データセット

本稿で Implicit Ruled Line Identifier と Cell Generator の学習に用いた論文集とデータセットを表 3 に示す．

学習に用いた表は合計で 209 件である．Implicit Ruled Line Idenitfier の入力である補助罫線候補の数は 25,642 であり，その $\frac{1}{6}$ が補助罫線，残りの $\frac{5}{6}$ が補助罫線ではなかった．割合が不均衡であったため，Oversampling の一手法である Synthetic Minority Over-sampling Technique(SMOTE) と Undersampling の一手法である Edited Nearest Neighbor(ENN) の 2 つを組み合わせた手法である SMOTEENN [16] によりこれらの数を同数にして学習を行う．

表 3: 学習に使用した表データ

論文集，データセット名	文書数	表数
NTCIR9 Spoken Doc [13]	9	27
NTCIR12 QA Lab-2 [14]	8	25
NTCIR12 SpokenQuery & Doc-2 [15]	6	16
Journal of Machine Learning Research Vol.18[9]	9	43
ICDAR2013 training dataset(EU)[10]	34	63
ICDAR2013 training dataset(US)[10]	25	35
合計	81	209

Cell Generator への入力は表中の隣接する 2 トークンとその周囲のトークンの特徴量であり，学習データにおけるその数は 31,235 件であった．また，その比率は結合:非結合=1:4 であった．不均衡であるが，表構造解析の精度を下げる原因となるのは，トークンを結合できないことよりも，結合すべきでない 2 トークンを結合することなため，サンプリングによりサンプル数を揃えることはしない．

モデルの評価と表構造解析実験には，ICDAR2013 Table Competition の test dataset[11] を用いた．このデータセットは EU，米国政府の発行した様々なドメインの文書 PDF から以下の条件にあう表を収集したものである．

- 明確な矩形領域とセル構造を持っている
- 上付き，下付き文字が含まれていない
- 表の大きさが 1 ページ以内である

このデータセットには EU の 27 文書の 76 の表，米国政府の 40 文書の 80 の表，計 67 の文章，156 の表が含まれている．

4.2 評価指標

Implicit Ruled Line Identifier における評価では，そのクラスタから生成される補助罫線候補のうち，実際にセルを分割するものを補助罫線とする．例えば図 5 では図 1 の表の “0.839” と “0.876” の間には複数本の補助罫線候補が推定されるがそれらはすべて正しい．Implicit Ruled Line Identifier はこのラベル予測の再現率，適合率と，セル間の補助罫線の重複を削除した上で，補助罫線が必要なセル間に実際にどれだけ補助罫線を推定できたかによって評価する．Cell Generator は隣接 2 トークンに対する “結合”，“非結合” のラベル予測の再現率，適合率によって評価する．

表構造解析精度の評価は ICDAR2013 を共催した Göbel らが提案したセルの隣接関係に基づく評価指標 [17] を用いる．図 10 に元の表と解析結果の表の隣接関係を示す．図中の黒い四角は正しい隣接関係を，白い四角は間違った隣接関係を表している．隣接セルが空白セルの場合，そこには隣接関係を定義せず，空白のセルを飛ばして隣接関係を持つと定められている．図 10 の (b) の表では “Increase” の右のセルが空白のため，空白のセルを飛ばし “Decrease” との間に隣接関係を持つ．下に

9 : http://www.jmlr.org/
10 : https://roundtrippdf.com/en/downloads/

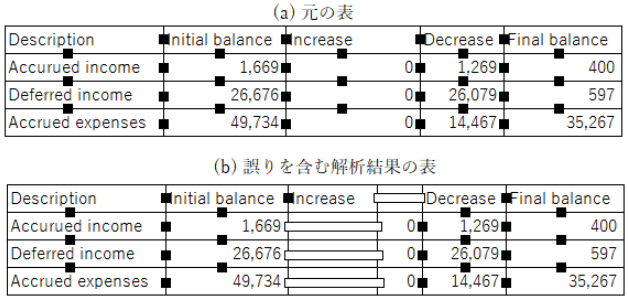


図 10: 元の表と誤りを含む解析結果の表の隣接関係 [17]

表 4: 補助罫線推定の再現率と適合率

	再現率	適合率
補助罫線	0.91	0.74
非補助罫線	0.92	0.98

表 5: トークン結合推定の再現率と適合率

	再現率	適合率
結合	0.95	0.96
非結合	0.99	0.99

は空白のセルしかないので，隣接関係を持たない．この隣接関係の再現率と適合率を次式で算出する．

$$\text{再現率} = \frac{\text{解析結果の正しい隣接関係数}}{\text{正解データのすべての隣接関係数}}$$

$$\text{適合率} = \frac{\text{解析結果の正しい隣接関係数}}{\text{解析結果のすべての隣接関係数}}$$

4.3 実験結果

表 4 に補助罫線推定結果の再現率，適合率を示す．また，セル間の補助罫線の重複を削除すると，補助罫線が必要なセル間の 98 %に補助罫線を予測できていた．表 4 を見ると補助罫線クラスの適合率が他と比べて低くなっている．この原因については，5.1 節で考察する．

次に Cell Generator によるトークンの結合推定結果の再現率，適合率を表 5 示す．

最後に表構造解析精度を表 6 に示す．ICDAR2013 の参加者の結果と我々の DocEng2019 [7] での結果，Sigarov ら [8] と Chi ら [9] の結果を表 6 にまとめる．本稿の提案手法は，他の手法全ての F 値を上回ることができた．特に ICDAR2013 において最高の成績を収めた Nurminen [5] の手法と比較すると，その結果を 0.9 ポイント上回った．

5 考　　察

5.1 Implicit Ruled Line Identifier の分析

表 4 では補助罫線の適合率が低くなっている．特にトークンが 2 行以上になるなどトークン数が多いセル中に誤って補助罫線を推定することが多かった．つまり，複数行のトークンからなるセルの過分割が大きな原因となっている．

また，データ不足も低適合率の原因である．4.1 節で述べたように SMOTEENN によって学習時の補助罫線と非補助罫線のサンプルの構成比はおおよそ同じにした．しかし，補助罫線候補のクラスタの種類ごとにデータ数の偏りが存在する．表 7 に学習データのクラスタの種類ごとのデータ数を，表 8 にそれぞれクラスタの種類ごとの推定結果の F 値を示す．表 7 を見ると，水平方向の補助罫線候補は補助罫線と非補助罫線の数の比率がせいぜい 1 : 2 であるのに対し，垂直方向の補助罫線候補は非補助罫線がはるかに多く，その数は 10 倍以上である．また，表 8 を見ると，他のクラスタに比べて右端や左端の点のクラスタにおける補助罫線推定の F 値が低くなっている．そのため，補助罫線推定の精度向上には，垂直方向の補助罫線の推定精度を向上させることが必要である．

表 6: ICDAR2013 データセットにおける表構造解析結果

手法	再現率	適合率	F 値
提案手法	0.951	0.960	0.955
Nurminen [4] (1st ranked)	0.941	0.952	0.946
Shigarov [8](2016)	0.923	0.950	0.936
DocEng'2019 [7]	0.916	0.917	0.917
GraphTSR [9](2019)	-	-	0.872
2nd ranked [4]	0.640	0.614	0.627
3rd ranked [4]	0.481	0.570	0.522

表 7: クラスタの種類ごとの学習データ数

	垂直方向の補助罫線候補			水平方向の補助罫線候補		
	右端	左端	重心の中点	下端	上端	重心の中点
補助罫線	325	481	317	1120	1119	1134
非補助罫線	5431	4937	5419	1747	1880	1732

表 8: クラスタの種類ごとの Implicit Ruled Line Identifier による補助罫線推定結果 (F 値)

	垂直方向の補助罫線候補			水平方向の補助罫線候補		
	右端	左端	重心の中点	下端	上端	重心の中点
補助罫線	0.62	0.52	0.78	0.88	0.87	0.89
非補助罫線	0.97	0.96	0.96	0.90	0.91	0.92

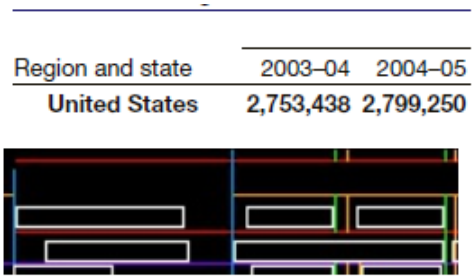


図 11: トークン間の距離が小さい表

5.2 Cell Generator の分析

Cell Generator の失敗は主に 2 種類あった．まずトークン間の距離の小さいトークンを誤って結合してしまうことである．誤結合した表の一部を図 11 に示す．上が元の表であり，下が解析結果である．

図中の “2,753,438” と “2,799,250” の 2 つのセルの間には補

助罫線が推定されていた．しかし，補助罫線の誤推定があるため，Cell Generator は補助罫線があったとしても距離が近い，つまり単語間の空白であると判断されれば表 11 のように結合してしまう．

次にサブヘッダ行中のトークンの誤結合である．サブヘッダ行は 1 つのセルしか持たない特殊な行である．Cell Generator は結合対象トークンの周囲のトークンを考慮しているが，表全体を見ないと判断することが難しいサブヘッダ行を誤結合することがしばしばあった．そのため結合対象の周囲のトークンだけでなく表全体を考慮する Cell Generator，もしくは後処理が必要である．

5.3 後処理の分析

表構造解析の失敗の原因は補助罫線の推定とセルの生成の失敗以外には以下がある．

まず，ヘッダ部分の判定の失敗である．データに数値と文字の入り混じった表では，正しくヘッダ行やヘッダ列の判定ができなかった．そのため，行や列の結合やセルの範囲の拡張に失敗した表があった．

また，セルの拡張において，周囲のセルとの位置関係によってその拡張方法を決定する．しかし，偶然周囲のセルと左端や右端のみが揃っていたために，中央揃えにもかかわらず誤って左揃えや右揃えと判定され，拡張に失敗した表があった．

6 おわりに

本稿では，機械学習を用いた表構造解析手法を提案した．提案手法はまず，文章 PDF から単語であるトークンと罫線を抽出し，それを基に Implicit Ruled Line Identifier を用いて表中の補助罫線を推定する．次にこの結果とトークンの情報を用いて Cell Generator により隣接 2 トークンが結合するか否かを推定する．最後にルールによって行と列の結合と複数行または列に隣接するセルの範囲を拡張して表構造を決定する．文書解析の著名な国際会議である ICDAR2013 の表構造解析タスクにおいて提供された 156 の表の構造解析を行い，同コンペティションにおいて用いられたセルの隣接関係に基づく評価指標で評価した結果として，我々の手法は再現率，適合率，F 値がそれぞれ，0.951，0.960，0.955 となった．これは他の手法を上回る結果であった．特に DocEng2019 [7] において我々の提案したルールに基づく表構造解析手法と F 値によって比較すると 4 ポイントほど高く，ICDAR2013 で最高の結果であった Nurminen の結果を 0.9 ポイント上回った．

今後の課題としては，補助罫線予測の予測精度の改善と周囲のトークンだけでなく表全体の特徴を加味することのできるトークン結合モデルの構築が挙げられる．また，解析した表構造を利用したグラフ生成アプリケーションの開発も行いたい．

謝　辞

本研究の一部は，科学研究費補助金基盤研究 (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」および国立情報学研究所共同研究の援助による．ここに記して深謝する．

文　献

[1] 衣川和尭, 鶴岡慶雅. 学術論文の章構造に基づくニューラル自動要約モデル. 言語処理学会第 23 回年次大会発表論文集, pp. 150–153, 2017.

[2] I. Augenstein, et al. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proc. of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 546–555, 2017.

[3] G. Kamola, M. Spytkowski, M. Paradowski, and U. Markowska-Kaczmar. Image-based logical document structure recognition. *Pattern Analysis and Applications*, Vol. 18, No. 3, pp. 651–665, 2014.

[4] M. Göbel, et al. ICDAR 2013 table competition. In *Proc. of the 12th International Conference on Document Analysis and Recognition*, pp. 1449–1453, 2013.

[5] A. Nurminen. Algorithmic extraction of data in tables in PDF documents. Master's thesis, Tampere University of Technology, 3 2013.

[6] R. Yamada, M. Ohta, and A. Takasu. An automatic graph generation method for scholarly papers based on table structure analysis. In *Proc. of MEDES'18*, pp. 25–28, 2018.

[7] M. Ohta, R. Yamada, T. Kanazawa, and A. Takasu. A cell-detection-based table-structure recognition method. In *Proc. of the ACM Symposium on Document Engineering 2019*, 2019.

[8] A. Shigarov, A. Mikhailov, and A. Altaev. Configurable table structure recognition in untagged PDF documents. In *Proc. of the 2016 ACM Symposium on Document Engineering*, p. 119–122, 2016.

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

[10] T. Mikolov, et al. Distributed representations of words and phrases and their compositionality. In *Proc. of the 26th International Conference on Neural Information Processing Systems - Volume 2 (NIPS 2013)*, p. 3111–3119, 2013.

[11] T. Mikolov, et al. Efficient estimation of word representations in vector space. *CoRR*, Vol. abs/1301.3781, , 2013.

[12] T. Chang and Y. Tsai. Asee: An automated question answering system for world history exams. In *Proc. of the 12th NTCIR Workshop Meeting*, pp. 445–450, 2016.

[13] A. Tomoyosi, et al. Overview of the IR for spoken documents task in NTCIR-9 workshop. In *Proc. of the 9th NTCIR Workshop Meeting*, pp. 223–235, 2011.

[14] H. Shibuki, et al. Overview of the NTCIR-12 QA Lab-2 task. In *Proc. of the 12th NTCIR Workshop Meeting*, pp. 392–708, 2016.

[15] T. Akiba, et al. Overview of the NTCIR-12 SpokenQuery&Doc-2 task. In *Proc. of the 12th NTCIR Workshop Meeting*, pp. 167–179, 2016.

[16] G. Batista, R. Prati, and M. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, Vol. 6, pp. 20–29, 2004.

[17] M. Göbel, E. Oro, and G. Orsi. A methodology for evaluating algorithms for table understanding in PDF documents. In *Proc. of the ACM Symposium on Document Engineering 2012*, pp. 45–48, 2012.