ニューラルネットワークによる表構造解析の誤り分析

Error Analysis of Table Structure Analysis Using Neural Networks

細谷 亮太

Ryota Hosoya

岡山大学 太田研究室

Ohta Laboratory, Okayama University

概要 学術論文において，表は実験結果を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においては，グラフのほうが適している．よって，表データをグラフに自動変換できれば，論文閲覧者の論文の理解の助けになる．一方，表をグラフに自動変換するには，表構造を正しく解析する必要がある．そのため，青柳らは [2] において，3 種類のニューラルネットワーク（NN）を用いる表構造解析手法を提案した．本稿では，青柳らが提案した手法において，正しく解析できない表などを分析した．

1 はじめに

近年，学術論文データベースの充実により，学術論文を手軽に入手できるようになった．学術論文では，様々な実験結果をまとめるために，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，表をグラフに自動変換できれば，論文閲覧者が論文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．山田らは，トークンの位置関係から補助罫線を推定するニューラルネットワーク（NN）と，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 [1]．ここで，トークンは表中の文字列を意味している．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た表構造解析の精度評価では，再現率，適合率，F 値がそれぞれ 0.951，0.960，0.955 となり，これは ICDAR2013 Table Competition の参加者の中の最良の記録を上回った．また，青柳らは [1] の手法を改良した表構造解析手法を提案した [2]．具体的には，NN の特徴量として，トークンの分散表現を追加した．解析のプロセスも改良しており，表中のトークンをマージする NN，補助罫線を推定する NN，表中のトークンをさらにマージすることでセルを生成する NN を用いて表構造解析を行った．ICDAR2013 Table Competition のデータセットと評価指標を用いた精度評価では，再現率，適合率，F 値がそれぞれ 0.967，0.977，0.972 となり，山田らの手法 [1] を上回った．本稿では，青柳らが提案した [2] の解析手法において，正しく構造解析することができなかった表について，その原因と改善策を検討する．

	Method A	Method B	Method C
Dataset 1			
Data X	0.45	0.65	0.43
Data Y	0.76	0.19	0.55
Dataset 2			
Data Z	0.56	0.98	0.59

図 1: 表と表の構成要素

2 表の構成要素

図 1 に本稿で扱う表の例を示す．まず，実線と点線は，それぞれ罫線，補助罫線である．なお，補助罫線とは，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罫線のことである．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，表中の文字列に対応している．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．

3 青柳らの表構造解析手法

青柳らの表構造解析手法 [2] の概要を図 2 に示す．まず，入力された PDF 文書を，pdfalto[*1] 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[*2] を用いて罫線の情報を検出する．次に，検出した情報をもとに，ニューラルネットワークを

[*1] https://github.com/kermitt2/pdfalto

[*2] https://github.com/pdfminer/pdfminer.six

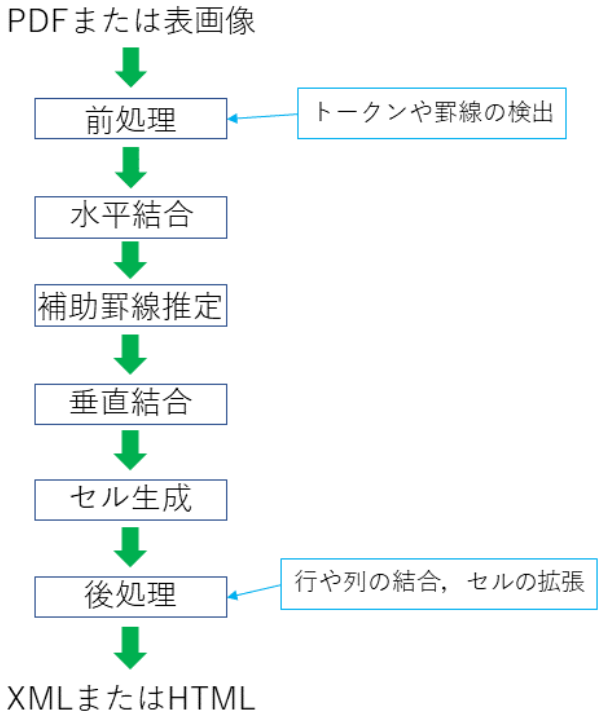

図 2: 青柳らの表構造解析手法 [2] の概要

用いて隣接した 2 つのトークンを水平結合する．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，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．次に，入力にトークンの特徴量と，トークンの文字列の分散表現を用いて，補助罫線を推定する．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．また，水平結合と垂直結合した 2 トークンをさらに結合することで，セルを生成する．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，セルの拡張を行うことで表構造を決定する．

4 表構造解析の実験

4.1 実験の概要

今回の表構造解析の実験で使用する学習データとして，ICDAR2013 training dataset の表 98 件を含む合計 209 件 [3] の表で学習した．

テストデータとして，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）論文集の PDF 文書の表を解析する．解析結果は XML ファイル，HTML ファイルとして出力する．

		0～20%	21～40%	41～60%	61～80%	81～100%
強制 UI	肯定記事	0	1	45	5	6
	否定記事	2	2	44	2	0
コスト UI	肯定記事	0	7	54	11	16
	否定記事	13	15	57	4	0
任意選択 UI	肯定記事	1	15	47	26	16
	否定記事	7	33	48	11	0

図 3: 論文 [4] の表 1

		0～20%	21～40%	41～60%	61～80%	81～100%
強制 UI	肯定記事	0	1	45	5	6
	否定記事	2	2	44	2	0
コスト UI	肯定記事	0	7	54	11	16
	否定記事	13	15	57	4	0
任意選択 UI	肯定記事	1	15	47	26	16
	否定記事	7	33	48	11	0

図 4: 図 3 の表の構造解析結果

4.2 実験結果

表構造解析の成功例として，図 3 に論文 [4] の表 1 と，図 4 にその構造解析結果を示す．この表は青柳らの方法 [2] で正しく構造解析された．

図 5 に論文 [5] の表 4 を示す．この構造解析結果の図 6 をみると，青色でハイライトした，“ソース”と，“Python”にあたるセルの中に何も要素が入っていないことがわかる．これは pdfalto がこれらの文字列を XML ファイルに出力しなかったためである．しかし，空白となったこれらのセルの位置は正しい．また，オレンジ色でハイライトした，“Python”，“擬似コード（英/日）”のセルが結合して，“Python 擬似コード（英/日）”となっていた．これらのトークン間の距離は，他と比べて近い．なお，3 行 4 列目の“￥$”は，空白相当の文字列を置換したものである．

次に，図 7 に論文 [6] の表 1 を示す．この構造解析結果の図 8 をみると，オレンジ色でハイライトした右側のセルが誤って結合している．これは“正解率，“適合率”，“再現率”のトークン間の距離が比較的近いからである．また，オレンジ色でハイライトした左側のセルも誤って結合している．これは“再現率”，“F1”，“￥$”のトークン間の距離が比較的近いからである．

これらの実験結果では，正しく構造解析できなかったセルは，その中のトークンと隣接トークンとの距離が他と比べて近いことがわかる．解決策として，トークンの位置関係や分散表現の他に，表の意味構造を考慮したセル生成を行うことが挙げられる．

また，ニューラルネットワークの学習データに英語の表を利用しているため，日本語のデータセットを用

いた学習を行えば，精度の向上が期待できる．

5　まとめ

青柳らの手法を用いて表構造解析を行った．正しく表構造解析できない表には，隣接トークンとの距離が他と比べて近いトークンが存在するという共通点があった．今後は，日本語の表を学習させて，日本語の表の表構造解析精度を評価したい．

参考文献

[1] 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.

[2] 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.

[3] Hiroyuki Aoyagi，Teruhito Kanazawa，Atsuhiro Takasu，Fumito Uwano，and Manabu 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.

[4] 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 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 (DEIM2022)，I44-13，2022.

[5] 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 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 (DEIM2022)，H41-4，2022.

[6] 吉井碧海，山本泰生， 西村雅史，“時系列センサーデータによる鋳造品質の要因分析に向けた深層学習モデルの検討,” 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 (DEIM2022)，H21-4，2022.

目的	ソース	生成物	想定利用者	実現手法
プログラミング支援 [9]	英語	形式言語	エンドユーザ	構文木マッピング
Excel アドイン [10]	英語	表計算プログラム	エンドユーザ	意味解析
コーディング支援 [11]	英語	Python	全てのプログラマー	NN モデル
コード理解支援 [12]	Python	擬似コード (英/日)	プログラミング初学者	SMT
コード理解支援 [13]	Java	コメント (英)	開発者	NMT(seq2seq)
プログラミング体験の向上 [14]	英語	LOGO プログラム	エンドユーザ	ルールベース
例外指示 [15]	日本語	SDC(意味構造)	自動運転システム	SDC/CCG
本研究: プログラミング支援	日本語	Python	プログラミング初学者	NMT(mT5)

図 5: 論文 [5] の表 4

目的		生成物	想定利用者	実現手法
プログラミング支援 [9]	英語	形式言語	エンドユーザ	構文木マッピング
Excel アドイン [10]	英語	表計算プログラム	エンドユーザ¥$	意味解析
コーディング支援 [11]	英語	Python	全てのプログラマー	NN モデル
コード理解支援 [12]	Python	擬似コード (英/日)	プログラミング初学者	SMT
コード理解支援 [13]	Java	コメント (英)	開発者	NMT(seq2seq)
プログラミング体験の向上 [14]	英語	LOGO プログラム	エンドユーザ	ルールベース
例外指示 [15]	日本語	SDC(意味構造)	自動運転システム	SDC/CCG
本研究: プログラミング支援	日本語		プログラミング初学者	NMT(mT5)

図 6: 図 5 の表の構造解析結果

	テストデータ (手法別)				テストデータ (共通)			
	正解率	適合率	再現率	F1	正解率	適合率	再現率	F1
提案手法	0.80	0.79	0.79	0.79	0.80	0.79	0.79	0.79
ランダム抽出 (比較 1)	0.49	0.48	0.34	0.39	0.51	0.35	0.35	0.34
JOF-LN (比較 2)	0.62	0.62	0.60	0.61	0.73	0.72	0.67	0.69

図 7: 論文 [6] の表 1

	テストデータ (手法別)				テストデータ (共通)			
	正解率	適合率	再現率	F1 ¥$	正解率	適合率	再現率	F1
提案手法	0.80	0.79	0.79	0.79	0.80	0.79	0.79	0.79
ランダム抽出 (比較 1)	0.49	0.48	0.34	0.39	0.51	0.35	0.35	0.34
JOF-LN (比較 2)	0.62	0.62	0.60	0.61	0.73	0.72	0.67	0.69

図 8: 図 7 の表の構造解析結果