テキスト情報を用いた表構造の修正

鈴木 祥子[1] 那須川 哲哉[1] 吉田 一星[1] Lihong He[2]
[1] 日本アイ・ビー・エム株式会社 東京基礎研究所 [2]IBM Research - Almaden
{e30126,nasukawa,issei}@jp.ibm.com lihong.he@ibm.com

概要

画像や PDF として保存された文書から表の構造を抽出する試みは多く行われており、特に画像認識モデルを用いた手法の性能が良いことが知られている。しかし、多種多様な表の抽出において、一律に高い抽出性能を出すことは難しい。本論文では、表の性質によらず共通して現れる抽出誤りを修正する、テキスト情報を活用する方法を提案する。多様な形式の表データセットで評価することで、提案手法がロバストに表構造の抽出性能を向上させることを示す。

1 はじめに

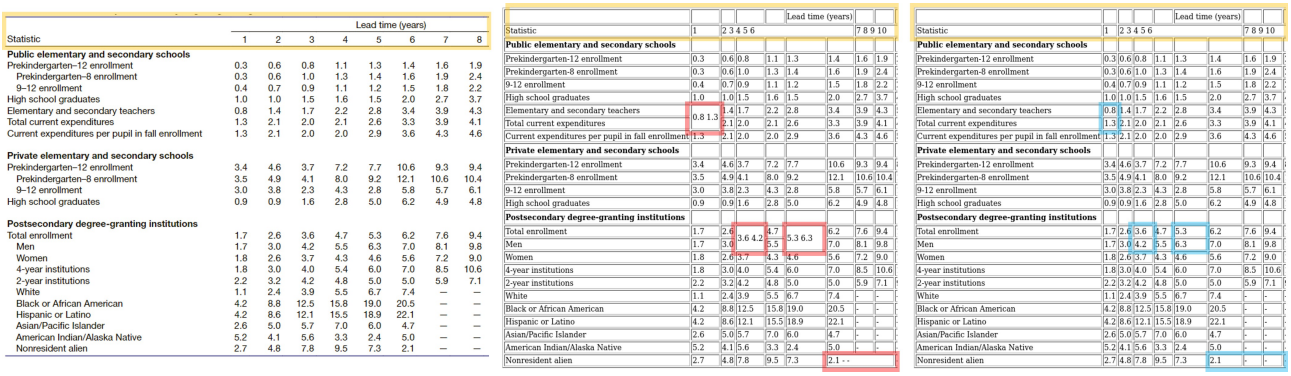
文書中の表（テーブル）には有用な情報が豊富に含まれているが、画像や PDF として保存された表からそれらを確実に抽出することは容易ではない。文書からの表抽出のタスクは、大きく「表領域の特定」と「表構造の抽出」から成る [4, 2]。ここで表構造とは、各セルが表の m 列 n 行に属する、という情報である。セルは空白セルであるか、あるいは文書中の座標と紐づいたテキストを含む。ここでは数字を含む文字列全般をテキストとして扱う。表構造が抽出されれば、例えば html で表を再構築することができる。図 1 は文書中の表と表抽出結果の一例である。表抽出タスクを解くアプローチとして、深層学習による画像認識モデルの応用が近年盛んである [10, 7, 5]。

しかし、画像認識モデルによるアプローチには 2 つの課題がある。1 つは、表領域と表構造の抽出性能が表の形式に大きく依存して変化することである。表抽出向け画像認識モデルの訓練データとして、科学論文や経済レポート中の表などが通常利用される [11, 10, 7]。これらのデータで訓練されたモデルは、実ビジネスでの適用対象となる請求書やバランスシート (例えば [12, 13]）に対しては抽出精度が高くない。この問題を解決するために対象分野の表を含む文書を用意してモデルをファインチューニングすることは可能であるが、表抽出の訓練データ作成には多大なアノテーションコストを要するため [9]、多数の表形式に対して都度ファインチューニングを実施することは現実的ではない。もう 1 つの課題は、従来の画像認識モデルによる表抽出では、表のセル内テキスト情報が表構造を読み解く手がかりを与えるにも関わらず利用されないことが多い点である。

本研究では、テキスト情報を利用して表構造の抽出性能を向上させる方法を検討する。特に、表抽出の SotA である画像認識モデルによりあらかじめ抽出された表に対して、表構造の誤りを修正する後処理の手法を提案する。提案手法では入力の表構造から異常セルを特定し、テキスト情報を用いて可能な表変形を検討することで表構造の誤りを修正する。実用上、後処理の実行時間は、画像認識モデルによる表構造の抽出時間に比べて小さいことが望ましい。このため、自然言語処理においても様々な分野で効果を発揮する深層学習モデルは使用せず、軽量かつ表データのドメインに依存しない手法を考案した。種々のドメインの表データでの実験を行い、提案手法を用いることで表構造の抽出性能が画像認識モデルによる結果よりも向上することが確かめられた。本手法はまた、修正の対象となる表がどのようなアプローチによって抽出されたものなのかを問わない。例えば人手で作成した表に対して誤りが含まれている可能性があった時、それを修正するという場面でも利用が可能である。

2 画像認識モデルによる表抽出

文書が html の table タグなどにより構造化されていないケースでの表抽出は、画像認識モデルの性能が良いことが知られている。[10] では文書内の表の領域、及び表内のセルの領域の特定に画像認識モデルを利用している。PDF もしくは画像データを入力とし、出力は表領域、表構造、各セル内テキストの



Statistic	Lead time (years)							
	1	2	3	4	5	6	7	8
Public elementary and secondary schools								
Prekindergarten–12 enrollment	0.3	0.6	0.8	1.1	1.3	1.4	1.6	1.9
Prekindergarten–8 enrollment	0.3	0.6	1.0	1.3	1.4	1.6	1.9	2.4
9–12 enrollment	0.4	0.7	0.9	1.1	1.2	1.5	1.8	2.2
High school graduates	1.0	1.0	1.5	1.6	1.5	2.0	2.7	3.7
Elementary and secondary teachers	0.8	1.4	1.7	2.2	2.8	3.4	3.9	4.3
Total current expenditures	1.3	2.1	2.0	2.1	2.6	3.3	3.9	4.1
Current expenditures per pupil in fall enrollment	1.3	2.1	2.0	2.0	2.9	3.6	4.3	4.6
Private elementary and secondary schools								
Prekindergarten–12 enrollment	3.4	4.6	3.7	7.2	7.7	10.6	9.3	9.4
Prekindergarten–8 enrollment	3.5	4.9	4.1	8.0	9.2	12.1	10.6	10.4
9–12 enrollment	3.0	3.8	2.3	4.3	2.8	5.8	5.7	6.1
High school graduates	0.9	0.9	1.6	2.8	5.0	6.2	4.9	4.8
Postsecondary degree-granting institutions								
Total enrollment	1.7	2.6	3.6	4.7	5.3	6.2	7.6	9.4
Men	1.7	3.0	4.2	5.5	6.3	7.0	8.1	9.8
Women	1.8	2.6	3.7	4.3	4.6	5.6	7.2	9.0
4-year institutions	1.8	3.0	4.0	5.4	6.0	7.0	8.5	10.6
2-year institutions	2.2	3.2	4.2	4.8	5.0	5.0	5.9	7.1
White	1.1	2.4	3.9	5.5	6.7	7.4	—	—
Black or African American	4.2	8.8	12.5	15.8	19.0	20.5	—	—
Hispanic or Latino	4.2	8.6	12.1	15.5	18.9	22.1	—	—
Asian/Pacific Islander	2.6	5.0	5.7	7.0	6.0	4.7	—	—
American Indian/Alaska Native	5.2	4.1	5.6	3.3	2.4	5.0	—	—
Nonresident alien	2.7	4.8	7.8	9.5	7.3	2.1	—	—

図 1　左：文書中の表の例、中央：左の表を [10] で表抽出した出力（赤の囲みはデータセルが行/列方向に over-merged となる誤り）、右：中央の表出力を本手法で修正した結果（青の囲みは修正箇所）。橙の囲みは列ヘッダーを表す。

位置情報である。ここではテキストの文字列情報および位置情報は、文書がそれらを自明に保持している場合には（例えば programmatic PDF の場合）そのまま利用し、そうでない場合は OCR によって抽出されている。しかしテキスト情報は表の領域特定、あるいは表構造抽出に利用されていない。[7] でも表構造抽出に画像認識モデルを利用するが、モデルを OCR に依存しない形にデザインしているため、セル内のテキスト情報は利用されない。

これらの手法は表抽出の性能において SotA といえる。しかしビジネス上表抽出が必要な請求書やバランスシートに対しては抽出精度が高くない。特に、これらの画像認識モデルを適用の際、セルの境界検出に失敗すると、最終的な表にセルの over-merged あるいはセルの over-split というタイプの誤りとなって現れる。over-merged は本来属しているよりも広い範囲の行または列にわたって複数セルが結合されてしまう誤りである。図 1 の左は文書中の表の例、中央はこれを画像認識モデルを用いた表抽出ツール [10] で抽出した結果を表す。中央の表には over-merged な誤りが存在する。セルの over-split とは、単独のセルが複数のセルに分割されてしまう誤りである。

3　提案手法の説明

本手法では、画像認識モデルで利用されていないテキスト情報に着目し、2 節で述べたセルの over-merged あるいは over-split の誤りを解消することで表構造の抽出性能を上げることを試みる。本手法は文書内から画像認識モデルなどによって表が抽出された後、後処理として適用することを想定している。本手法の入力となる表の持つ情報として以下を想定する。

- 各セルの文書内座標情報
- token の文書内座標情報
- 表の構造（各セルに対する行、列の id）

ここで token とは、空白を含まず一塊になっている水平方向に並んだ文字の集合を表す。

本手法では十分一般的な仮定として、表は列ヘッダーを持つこととする。列ヘッダーは各列のラベルを示すセルの並びである。

提案手法が着目する表の性質は、表中の各列において、列ヘッダーを除くセル内のテキストはその列内で類似性を持つことである [1]。例えば図 1 の左の表では、最上部の橙の囲みが列ヘッダーであり、列ヘッダー以外の第 1 列（左端の列）のセル内テキストはアルファベットと数字の組み合わせである。第 2 列以降はどの列も小数点第一位までの数値、あるいは空の値を表す記号で占められている。本研究ではこの同一列内類似性を前提とし、類似性を破るセルを異常セルとして抽出する。次に、この異常セルを減らす表の変形を探索し表の修正を行う。この操作を繰り返すことで、入力の表に含まれる誤りを除去し、表構造の抽出性能を高めることを目的とする。

提案手法は次のステップからなる。

1. 表の列ヘッダーの特定
2. 各セルの異常度スコアを算出
3. 各列ごとにセル内テキストの正当性予測モデルを学習
4. 異常セルの特定
5. 異常セルの変形パターンをスコア化
6. 最大スコアを持つ変形パターンに従い表を修正
7. 4-6 のステップの繰り返し

3.1 列ヘッダーの特定

列ヘッダーは列内セル類似性の前提に含まれないため、除く必要がある。ここでは列ヘッダーは表の上部の行にのみ含まれると仮定し、また列ごとに、列ヘッダーのテキストはそれ以外のセルのテキストと特徴が異なること、複数列ヘッダーがある場合にはそのヘッダーの組み合わせが各列について一意になること、を仮定し列ヘッダーの特定を行う。

3.2 セルの異常度スコア算出

2 節で述べたように、画像認識モデルでは表構造抽出においてセルの over-split あるいは over-merged という誤りが発生しやすいという問題がある。このタイプの誤りが起きた場合、同一列に属するセル内テキスト類似性が壊れる可能性が高い。このようなセルを入力の表から検出するために異常度スコアを次のように定義する。

1. 各セルの形状やテキストを特徴量化
2. 同一列に属するセル集合に対し、各特徴量の外れ値を決定
3. 外れ値を持つ特徴量をセルごとに特定し、該当する特徴量の個数をそのセルの異常度スコアとして算出

セルごとの特徴量は Appendix の A 節で示すようにセルの形状、あるいはセル内テキストの特徴からデザインできる。異常度スコアが閾値を超えたセルを異常セルとし、異常セルの集合を Y とする。

3.3 予測モデルの学習

前述のように本研究では、表内の各列において列ヘッダーを除くセル内テキストには類似性があることを前提としている。この前提が満たされる状況では、あるテキストが与えられた時、それがある列内に属するセルのテキストとして適切かどうか、という予測モデルを構築できる。

列 i に属するセル集合のうち、3.2 節で導出した異常セル集合 Y に含まれないセル集合を X_i と表す。各列 i の予測モデル M_i は入力をテキスト、出力を $[0,1]$ のスコアとして入力テキストの正当性を確率値で表すものを採用する。決定木やロジスティック回帰などが挙げられる。入力テキストの特徴ベクトル導出には各種アプローチがあるが、例えば各 token の以下のような表層上の特徴を利用する。

- 大文字のみ
- 小文字のみ
- 数値のみ
- 数値＋ピリオド
- 数値＋記号
- 文字＋数値
- 最初の 1 文字が大文字、残りが小文字
- 表に複数回出現する token
- 先頭文字
- 末尾の文字
- 文字数

学習の際は各セル $x \in X_i$ 内のテキストを予測モデルの正例とする。負例の入力テキストには、x の隣接セル（上下セル、左右セル）をランダムに結合させたもの、及びその部分文字列を用いる。

3.4 異常セルの変形パターンのスコア化

3.2 節で得た異常セル集合 Y から異常度スコアの最も高いセルを取り出し、変形を検討する。これをすべての異常セル $y \in Y$ が検討されるまで、あるいは決められたステップ数が完了するまで繰り返す。

異常セル y が特定された後、変形パターン候補の集合 $P(y)$ を導出する。セル y に over-merged の可能性のある時（行 span あるいは列 span の異常がある時）には、変形パターンは、y の token 列 $\{w_1, w_2, ...\}$ の行あるいは列方向への分割方法を列挙することになる。セル y に行方向の over-split の可能性がある時（同一行の空白セルの数に異常がある時）には、セル y と隣接する上下のセルを結合した token 列を考えた上で、この結合した token 列の行方向の分割方法を列挙すれば良い。本提案手法では、$P(y)$ を y の周辺セル情報から推定するアプローチをとる。Appendix の B 節及び図 2 に算出の詳細と例を示す。

次に、このようにして得られた各変形パターン候補 $P \in P(y)$ に対し、スコアを計算する。パターン P は、分割された token 列である仮想的なセル p の集合、及び各 p がどの列（行）j に属するかの情報を持つ。例えば図 2 では分割パターン 1 において、仮想セル $\{w_1, w_2\}$ は列 1 に属し、もう一つの仮想セル $\{w_3, ..., w_7\}$ は列 2 と列 3 に属している。3.3 節の予測モデルから $M_1(w_1, w_2) \times M_2(w_3, ..., w_7) \times M_3(w_3, ..., w_7)$ はこのパターンの正当性の確率値と解釈できる。そこで、列方向に対して over-merged な異常セルの変形パターン P のスコア $S(P)$ を式 (1) で定義する。ここで、p

表 1　提案手法適用前後の表構造抽出性能比較（数値は F1 スコアのパーセント表示）

表抽出モデル	データセット	A	B	C	D	E	F	G	H	I	J	K	L
	文書数	67	43	36	110	54	29	9	22	126	39	250	41
EfficientnetB0	適用前	77.2	40.3	38.6	80.5	23.3	79.3	**54.8**	43.7	75.0	57.0	72.2	67.2
	適用後	**77.3**	**42.4**	**41.7**	**80.8**	**25.1**	**81.9**	54.2	**45.5**	**77.1**	**58.6**	**73.8**	**71.0**
Resnet50	適用前	70.7	36.1	35.8	84.1	**26.9**	74.6	**58.7**	42.6	76.5	57.3	73.7	71.6
	適用後	**72.2**	**38.0**	**36.5**	**84.5**	**26.9**	**76.7**	58.6	**43.8**	**77.4**	**57.7**	**75.2**	**75.4**

は P に含まれる仮想セル、j は p が属する列番号、c は正の定数である。

$$S(P) = \sum_{p \in P} \ln\{M_j(\text{token sequences in } p) + c\} \quad (1)$$

同様に、行方向に over-merged な異常セル y のパターン P のスコアを式 (2) で定義する。行方向の over-merged ではパターン P は列方向と同様に仮想セル p を要素として持つが、予測モデルはセル y の属する列 k のものを適用する。

$$S(P) = \sum_{p \in P,k} \ln\{M_k(\text{token sequences in } p) + c\} \quad (2)$$

$P(y)$ 中の最大スコアのパターンに従い表を修正する。このステップを各異常セルに対して繰り返す。

4　評価実験

提案手法の効果を確かめるため、画像認識モデルを用いた表抽出手法 [10] の出力結果に対して、本手法を適用した。画像認識モデルは複数の backbone を採用し、結果を比較した。

4.1　データと設定

表の形式や内容はドメインに大きく依存するため、多様な表データセットを収集し評価に利用した。一部のデータは表抽出タスクのデータとしてアノテーション付与された状態で公開されている [4]。その他のデータは web 上で公開された表あるいは非公開の表に対しアノテーションを追加で付与している。セルの異常度スコア算出には、Appendix の A 節で示すセルの特徴量を利用した。セル内テキストの正当性予測モデルとしては決定木を用いた。予測モデルにおいて、セル内テキストの特徴ベクトルには 3.3 節で示した token 単位のものを利用する。token ごとの特徴ベクトルを導出後、先頭 token の特徴ベクトル、最終 token の特徴ベクトル、全 token の特徴ベクトルの和、を結合したものを予測モデルの入力として用いた。

4.2　評価結果

表構造の評価指標として、正しい表のセル隣接関係と出力した表のセル隣接関係から算出した F1 スコアを利用する [3]。12 種類の異なるドメインから成るデータセットについて、Resnet50[6] と EfficientNetB0[8] の 2 種の backbone の画像認識モデルを用いた表抽出手法 [10] を適用し、表領域の特定、および表構造の抽出結果を得た。この出力された表を入力として提案手法を適用した結果を表 1 に示す。なおデータセット D および I が公開データ [4] にあたる。多くのドメインにおいて、どちらの画像認識モデルでも提案手法適用後の F1 スコアが適用前より高く、提案手法の有効性が認められた。但し、データセット G のように適用後にスコアが悪くなるケースも存在する。このようなケースへの対応が今後の課題である。

5　まとめ

本論文では、表構造の誤りを表内のテキスト情報を用いて修正する手法を提案した。特に、文書からの表抽出タスクにおいて SotA である画像認識モデルを用いた手法の後処理として本手法を適用し、各種ドメインの表データに対して表構造の抽出性能が上がることを示した。

本手法では各列のテキスト表記に類似性があることを前提とする。この前提を満たす表であれば、今回の実験で利用した画像認識モデルの表抽出手法の出力に限らず、あらゆる表に対して本手法は適用可能である。

本研究では実行速度を重視したため、テキスト情報として表層上の特徴のみを利用し、シンプルな予測モデルを用いたが、深層学習の言語モデルを利用することで表構造抽出の性能は更に上がることが期待される。

参考文献

[1] Taha Ceritli, Christopher K. I. Williams, and James Geddes. ptype: probabilistic type inference. *Data Mining and Knowledge Discovery*, Vol. 34, No. 3, pp. 870–904, mar 2020.

[2] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Lang. Icdar 2019 competition on table detection and recognition (ctdar). In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1510–1515, 2019.

[3] Max C. Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. A methodology for evaluating algorithms for table understanding in PDF documents. In *Proceedings of the 12th Symposium on Document Engineering*, pp. 45–48. ACM, 2012.

[4] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pp. 1449–1453, 2013.

[5] Khurram Azeem Hashmi, Marcus Liwicki, Didier Stricker, Muhammad Adnan Afzal, Muhammad Ahtsham Afzal, and Muhammad Zeshan Afzal. Current status and performance analysis of table recognition in document images with deep neural networks. *CoRR*, Vol. abs/2104.14272, , 2021.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, Vol. abs/1512.03385, , 2015.

[7] Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, and Peter Staar. Tableformer: Table structure understanding with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4614–4623, June 2022.

[8] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, Vol. abs/1905.11946, , 2019.

[9] Nancy Xin Ru Wang, Douglas Burdick, and Yunyao Li. Tablelab: An interactive table extraction system with adaptive deep learning. *CoRR*, Vol. abs/2102.08445, , 2021.

[10] Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. Global table extractor (GTE): A framework for joint table identification and cell structure recognition using visual context. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pp. 697–706. IEEE, 2021.

[11] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno-Yepes. Image-based table recognition: data, model, and evaluation. *CoRR*, Vol. abs/1911.10683, , 2019.

[12] 宏田中. 全体最適化戦略に基づく複雑な帳票画像の自動認識に関する研究.

[13] 藤江翔太郎, 白松俊, 大囿忠親, 新谷虎松. フォーマット分類機構に基づく帳票管理支援システムの試作. 第 76 回全国大会講演論文集, Vol. 2014, No. 1, pp. 303–304, 03 2014.

A　セルの異常度スコア算出時の特徴量例

3.2 節では各セルの異常度スコアを算出するための全体的な流れを紹介した。利用される特徴量としては、セルの形状を反映したものとして

- 最左端 token の x 座標
- 最右端 token の x 座標
- セル行 span
- セル列 span
- セル内 token 数
- セルと同一行の空白セルの数

などがある。またテキストの特徴を反映したものとして、各 token に対し

- 大文字のみ
- 小文字のみ
- 数値のみ
- 数値＋ピリオド
- 数値＋記号
- 文字＋数値
- 最初の 1 文字が大文字、残りが小文字

などが考えられる。例えばセル形状の特徴量、セル内最初の token の特徴量、セル内最後の token の特徴量を結合して各セルの特徴量として扱うことができる。

B　token 列分割パターン算出の詳細

3.4 節では、異常セル y の変形パターンを探索し、各パターンにスコアを付与することを説明した。ここでは、y 内の token 列の変形パターンを導出する方法の詳細を述べる。over-split なセルを変形する際には、3.4 節で述べたように、一旦隣接セルと merge させた上で分割パターンを考えればよい。そこで以降は token 列の分割方法のみ考えることとする。また、以下では token 列の列方向の分割方法のみ説明する。行方向の分割パターンも同様の方法で抽出できる。

各 token には座標情報が紐づいている。token 列 $\{w_1, w_2, ...\}$ を列方向に分割する可能性を考えたとき、座標情報から許される分割と許されない分割とが存在する。このとき、各列に属する異常でないセルの座標情報から、列の右端、左端座標を推定し、可能な区切り箇所でのみ token 列の分割を許すことで、token 列が長い場合にも効率的な探索が可能になる。

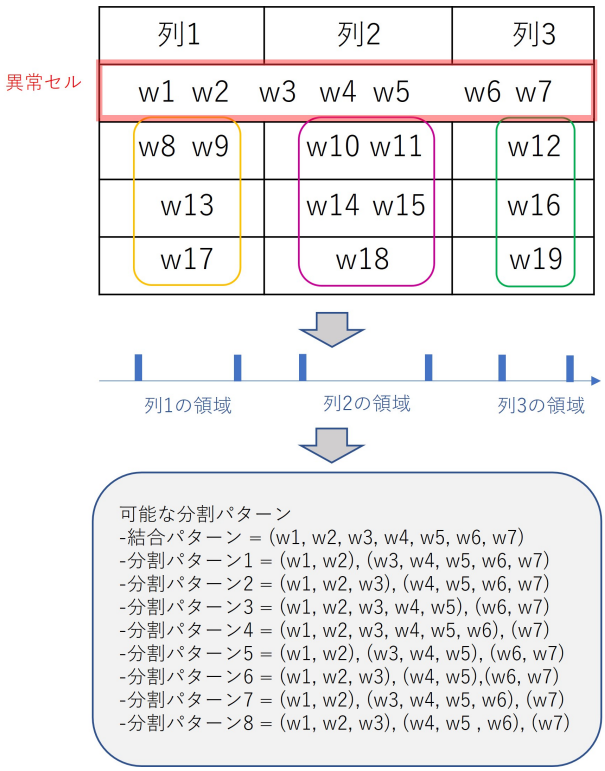


図 2　token 列分割パターン作成の例

図 2 を例に説明する。列 1 に属する異常でないセル内の token である w_8, w_9, w_{13}, w_{17} の座標情報から、列 1 に属するセルが必ず満たす x 領域が推定できる。この領域と token$w_1, w_2, ...$ の x 座標を比較したとき、w_1, w_2 は必ず列 1 に属していなければならないことが分かる。同様の処理で、w_4, w_5 は必ず列 2 に属し、w_7 は必ず列 3 に属する。これらの拘束条件を満たしつつ、token 列の分割を考えると全て merge された状態となんらかの分割された状態、合わせて 9 パターンが許される。例えば図 2 の分割パターン 2 は、$\{w_1, w_2, w_3\}$ が列 0 に属するセルであり、$\{w_4, ..., w_7\}$ が列 1 および列 2 に属する結合セルである。