PDF 形式の金融文書における 項目-数値間関係を考慮したテーブル情報抽出

Table Information Extraction from Financial PDF Documents Considering Item-Value Relation

青野有華 [1] * 市川幸史 [1] 近藤浩史 [1] 加藤淳也 [1]
Yuka Aono[1] Koji Ichikawa[1] Hirofumi Kondo[1] Junya Kato[1]

[1] 株式会社日本総合研究所
[1] The Japan Research Institute, Limited

Abstract: 有価証券報告書などの金融文書において，重要な情報はテーブル形式で記載されることもあり，テーブル内の情報抽出は金融データの更なる利活用に向けて重要な役割を果たすと期待される．しかし，企業が共通して開示する文書であっても，企業によってテーブル形式が異なることや，情報抽出の難しい PDF 形式で開示される文書も存在することから，現状テーブル情報が抽出され，十分に活用されているとは言い難い．そこで本研究では，PDF 形式で開示されている日本語金融文書内のテーブルからの情報抽出を試みた．我々の手法では，PDF 内の罫線情報を利用しテーブル領域およびテーブル内セルを抽出した．その上でセル内での改行とセルの区切りを区別するために，セル内項目情報および数値情報に着目した BERT ベースの分割判定モデルを構築した．実験では，2 種類の PDF 形式の金融文書に含まれるテーブルを対象とした性能評価実験を行い，我々の提案手法が優れた性能を発揮することを確認した．

1 はじめに

上場企業は有価証券報告書や四半期報告書など様々な文書を開示している．これらに加え近年では，SDGs に対する社会的な意識の高まりなどもあり，従来とは異なる情報開示の流れもある．金融庁は「ディスクロージャーワーキング・グループ」[1]という審議会を開き，企業情報の開示や提供のあり方を述べている．また，新しい開示項目として，新型コロナウイルス感染症や ESG に関する開示例を挙げた「記述情報の開示の好事例集 2020」[2]を発行している．このような官庁の動きもあり，今後は企業による情報開示は更なる広がりをみせると考えられる．

開示情報は，企業の評価や株式投資の意思決定において重要な情報源である．一方，情報開示の広がりに伴い，多くの情報から必要な情報のみを読み取り，評価することは時間やコストを多く必要とする．

この課題に対応するため，近年では機械学習や深層学習を活用した金融文書のテキストマイニングの研究がなされている (例えば [1])．また，企業によって文書の形式は異なり，それらを一覧化，データベース化することは，金融文書に対するテキストマイニング研究分野の最終的な目標の一つともいえる．近年の研究傾向として，金融文書内のテキスト情報に対しての研究は多くなされている（例えば [2])．一方で，金融文書にはテーブル形式で記載された情報もあるが，金融文書を対象としてテーブルからの情報抽出に重点をおいた研究は少ない．

そこで本研究では金融文書内のテーブル情報を正確に抽出することを目標とする．一般に有価証券報告書などは XBRL 形式での開示がなされているが，PDF 形式でのみ開示されている文書もある．そのため本研究では特に，金融文書の中でも，有価証券報告書に添付される「株主総会招集通知」などの，PDF 形式の文書を対象とする．

PDF からのテーブル情報の抽出には主に 2 点の課題がある．1 つめの課題は，罫線やテキストの位置関係のみから，適切にテーブルを検出し，構造を把握することである．この課題に対して本研究では，PDF 内の線分情報と文字情報を用いて解決を試みた．

2 つめの課題は，テーブルのセル内に複数の情報が含まれている場合の分割である．この場合，罫線を区切りとした単純なセル抽出では，これらが 1 つの情報として抽出される (図 1)．この状態では項目 (例えば科目) と数値 (例えば期末残高) の関連が不明確であるため，分析を行う際に特定の科目に関する合計を算出することや，項目のフィルタ検索をすることが難しい．そこで本研究では，BERT [3] を用いて，1 セル内に複数の情報を含むセルを，意味のある項目単位に分割することを試みた．その際に，金融文書のテーブルでは「科目」といった列 (項目列) に対して，「残高」列 (数値列) などの数値情報が対応していることが多いことから，数値情報の数を利用してラベル付きデータの構築や分割数の制限を行った．

提案手法の評価には株主総会招集通知にある「関連当事者との取引に関する注記」にあるテーブルと有価証券報告書にある「大株主の状況」にあるテーブルを対象とした[3]．その結果，我々の手法はそれぞれ F 値 0.995，0.981 を達成し，既存のオープンソースのソフ

*連絡先: aono.yuka@jri.co.jp
[1]https://www.fsa.go.jp/singi/singi_kinyu/tosin/20180628.html
[2]https://www.fsa.go.jp/news/r2/singi/20201106-3.html
[3]「大株主の状況」は XBRL 形式のデータが存在するが，実験のため PDF 形式に変換し評価を行った．

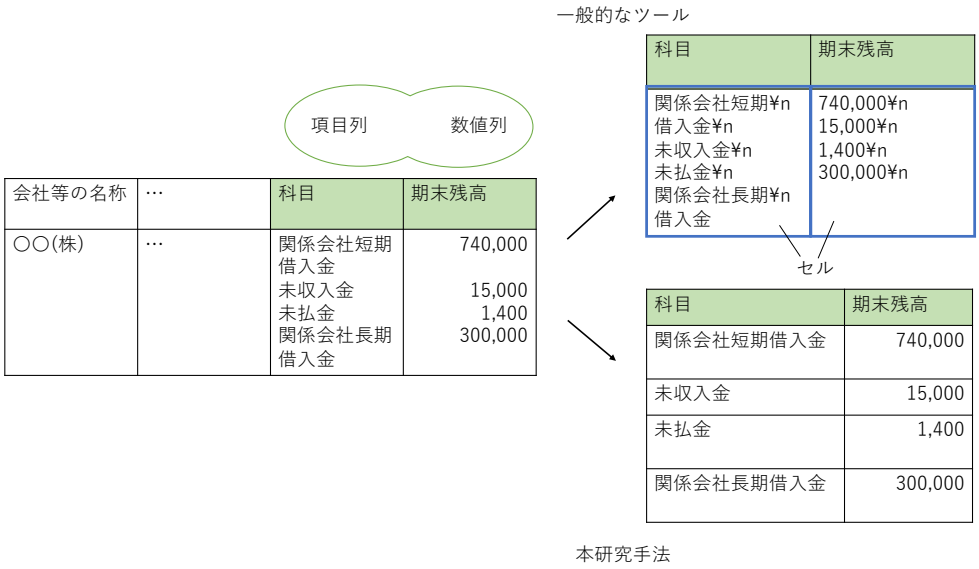


図 1: 本研究で対象とする抽出例

トウェアを利用した場合に比べてそれぞれ 0.35，0.38 ポイントの性能改善を確認した．

以下，2 章では問題設定を述べ，3 章では提案手法について，4 章では実験，5 章では結論，6 章では今後の課題を述べる．

2 問題設定

本研究では PDF 形式で開示された文書内のテーブルからの情報抽出を行う．具体的には，金融文書内の，あるセクションに企業間で共通して記載されるテーブルのうち，「項目」列とそれに紐づく同一テーブル内の「数値」列を抽出し，項目-数値間が 1 対 1 対応した情報を取得する．したがって，前提として以下の情報が与えられることを想定する．

1. 抽出対象のテーブルのおおよその位置（セクション名など）
2. 「項目」列の列名
3. 「数値」列の列名

分かり易さのため, 図 1 を使って具体例を挙げると,「項目」列の列名とは「科目」を指し,「数値」列の列名とは「期末残高」を指す．なお，本問題設定は企業評価等の実業務を想定して定めた．実業務では意味のある数値を企業間で比較する場面が多くあり，テーブル全体の情報よりも，一部の項目と対応する数値を抽出したいというケースが多いと想定する．この想定において，本問題設定は有効と考える．

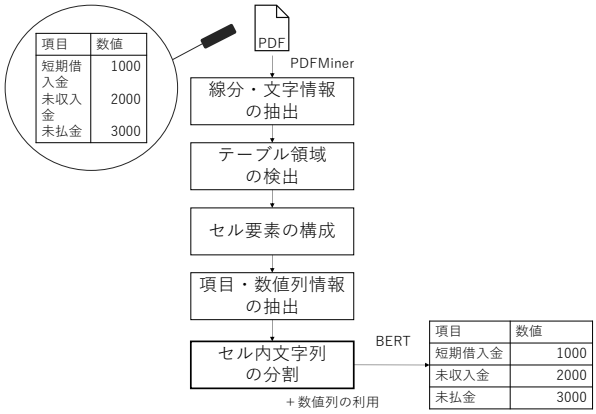


図 2: 提案手法の流れ

3 提案手法

提案手法では，図 2 に示す手順に従いテーブル内情報の抽出を行う．まず，PDF 文書を入力として，PDF 文書の線分情報と文字情報からテーブル領域を検出する．その後，テーブル領域内のデータを抽出するために，セル・列の構成や，機械学習モデルを用いたセル内文字列の分割を行う．最終的に，図 1 で示したような，入力として与えた「項目」列の列名および「数値」列の列名をもつ，2 列のテーブルデータを出力する．以下で，処理の詳細を紹介する．

3.1 線分・文字情報の抽出

第一ステップとして, PDF 文書から, 文書に含まれる線分情報と文字情報の抽出を行う．本研究では PDF 構成要素を PDFMiner[4]を用いて抽出し，抽出された線分要素と文字要素をテーブル情報の抽出のために用いる．この時, セクション名によるキーワードマッチングなどにより, 抽出対象のテーブルが含まれるおおよそのページ範囲は特定できていることを想定とする．以下, 線分要素を，その線分がもつ 2 端点の座標 $(x_1, y_1), (x_2, y_2)$ により

$$l = (x_1, y_1, x_2, y_2) \tag{1}$$

で表す．また文字要素を，その文字要素が存在する領域および文字により

$$c = (b, t) \tag{2}$$

で表す．ただしここで $b = (x_1, y_1, x_2, y_2)$ は文字が存在する左下の点座標 (x_1, y_1) および右上の点座標 (x_2, y_2) により形成される矩形領域を，また t は領域 b 内に含まれる文字を表す．抽出ツールにより，PDF の各ページごとに以下のような線分要素の集合 L，文字要素の集合 C が得られるとする:

$$L = \{l_1, l_2, \dots\}, C = \{c_1, c_2, \dots\}. \tag{3}$$

ただし，本研究では線分要素 l は水平あるいは垂直な線分のみを対象とし，それ以外のものは無視する．また要素に付随する文字フォントや線分の太さといったスタイル情報は使用しない.

3.2 テーブル領域の検出

次に，上記の線分要素集合 L を用いてテーブル領域の抽出を行う．一般にテーブルの構造は縦横の線分で構成されており，各線分は外枠または内罫線のいずれかを表している．そこで，接点あるいは交点のある線分同士を同一集合としたクラスターを作成し，これに基づきテーブル領域の検出を行う．この時，集合内の少なくとも 1 線分と接点あるいは交点が存在する線分もまた同一集合とする．例えば，線分 l_1 と線分 l_2，線分 l_2 と線分 l_3 で接点あるいは交点がある場合，線分 l_1, l_2, l_3 は同一クラスターとなる．この時，各クラスターごとにクラスター内の線分要素の座標の最小・最大値を用いて $(x_{\min}, y_{\min}), (x_{\max}, y_{\max})$ をテーブルが存在する矩形領域 b^{table} の左下および右上の座標として検出する[5]．さらに，クラスターに含まれる線分の数 n とクラスターが作る矩形領域 b^{table} の縦横の長さについてそれぞれに下限

$$n_{\text{lower}} \leq n, \tag{4}$$
$$w_{\text{lower}} \leq x_{\max} - x_{\min}, \tag{5}$$
$$h_{\text{lower}} \leq y_{\max} - y_{\min} \tag{6}$$

を設定し，よりテーブルらしい矩形領域のみを抽出する．ここで $n_{\text{lower}}, w_{\text{lower}}, h_{\text{lower}}$ は事前に定めた閾値である．この操作により，我々は各テーブル領域 b^{table} 内の線分要素の集合

$$L_{\text{table}} = \{l | l \subset b^{\text{table}}, l \in L\} \tag{7}$$

および，文字要素の集合

$$C_{\text{table}} = \{c | c \subset b^{\text{table}}, c \in C\} \tag{8}$$

を得ることができる．ただしここで $a \subset b$ は要素 a が領域 b 内に存在することを表している[6].

3.3 セル要素の構成

次に, 得られたテーブル領域内の線分要素集合 L_{table} および文字要素集合 C_{table} から，テーブルの各セル領域の検出およびセルに内包される文字列を構成し，セル要素を取得する.

まず, テーブル領域内の線分情報 L_{table} により, テーブル中の線分によって囲われた矩形領域 (セル領域) b^{cell} を検出する．これは文字集合 C_{table} の各要素 c の矩形領域 b の中心から最も距離の近い上下左右の線分を特定することで行われる．すなわち，特定された上下左右の線分の交点により形成される矩形領域をセル領域 b^{cell} として検出する．この操作により，テーブル領域 b^{table} 内に存在するセル領域の集合 $\{b^{\text{cell}} | b^{\text{cell}} \subset b^{\text{table}}\}$ および，各セル領域内の文字要素の集合 $C_{\text{cell}} = \{c | c \subset b^{\text{cell}}, c \in C_{\text{table}}\}$ を得ることができる.

その後，セル領域内の文字列 t^{cell} を C_{cell} を用いて以下の手順により構成する: まず文字要素 $c \in C_{\text{cell}}$ の矩形領域中心の y 座標 y_c から，セル内に存在する行ごとの文字列を構成する．具体的には，2 つの文字要素 $c, c' \in C_{\text{cell}}$ について，$|y_c - y_{c'}| < \delta$ ならば, c, c' を同一行に存在する文字とみなす．ここで δ はある閾値である．そして，得られた同一行とみなされた文字要素の集合に対しそれぞれ x 座標の小さい順にソートし，1 行分の文字列とみなすことで，文字列の集合 $\{s_1, s_2, \dots\}$ を構成する.

最後に，各文字列をそれぞれの y 座標の大きい順にソートし，文字列の間に [SEP] トークンを挿入した上で結合し，これをセル要素内の文字列とする．この操作により我々はセル要素

$$e^{\text{cell}} = (b^{\text{cell}}, t^{\text{cell}}) \tag{9}$$

を得る．ただしここでの文字列 t^{cell} は

$$t^{\text{cell}} = s_1[\text{SEP}]s_2[\text{SEP}]s_3 \dots \tag{10}$$

のように表される文字列である.

3.4 項目・数値列情報の抽出

次に, テーブル領域内のセル要素の集合 $\{e^{\text{cell}} | e^{\text{cell}} \subset b^{\text{table}}\}$ および線分要素の集合 L_{table} を用いて，列成分を構成する．これは集合 L_{table} から，垂直な線分要素

[4] https://github.com/pdfminer/pdfminer.six

[5] PDF ではページの左下を起点とし，右上ほど x, y 座標の値が大きくなる.

[6] より厳密には，この包含関係 $\subset$ はある小さい範囲内でのはみ出しを許容することにする．また，PDF によってはページ全体を囲う不可視の線分が抽出されることがあり，上記のテーブル検出方法においてしばしばページ全体がテーブル領域として取得される場合がある．これを取り除くため，実験では検出された 2 つのテーブル領域が $b \subset b'$ となっている場合，b' を除く処理を行っている.

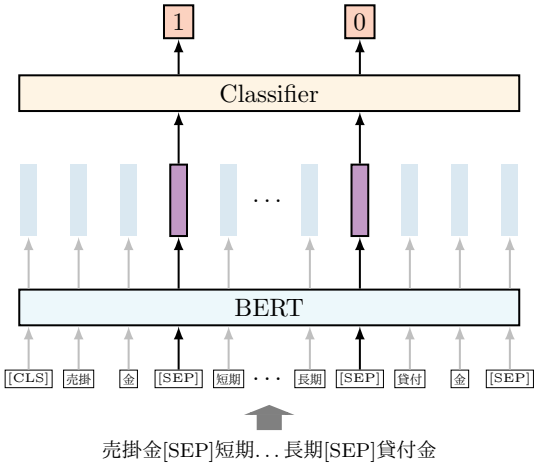


図 3: セル内文字列分割判定モデル概略図

の x 座標を抽出することによって行われる．すなわち，全ての垂直な線分要素 $l_{\text{vertical}} \in L_{\text{table}}$ の x 座標情報から集合 $X_{\text{table}} = \{x_0, x_1, \dots | x_0 < x_1 < \dots\}$ を取得した後，$x_{e^{\text{cell}}} \subset (x_i, x_{i+1})$ を満たす e^{cell} を i カラム目の要素とする．ただしここで $x_{e^{\text{cell}}}$ はセル要素 e^{cell} が x について占める領域を表す．この操作により，テーブル中の列要素の集合

$$\{\text{COL}_1, \text{COL}_2, \dots\} \tag{11}$$

を得る．ただしここで COL_i は

$$\text{COL}_i = \{e^{\text{cell}} | x_{e^{\text{cell}}} \subset (x_i, x_{i+1}), e^{\text{cell}} \subset b^{\text{table}}\} \tag{12}$$

のように表されるセル要素の集合である．

「項目」列および「数値」列情報の抽出は以下のように行う: 上記の手法で得られたテーブル中の各列のうち，列中のセル要素内文字列 t^{cell} に ([SEP] トークンを除いた後) 設定したキーワードがマッチするものを抽出する．本研究においては，我々は項目列名のキーワードおよび数値列名のキーワードにより，2 つのカラム $\text{COL}_{\text{項目}}$ と $\text{COL}_{\text{数値}}$ を取得する．また，ここで我々は項目列と数値列が紐づいている状況を想定する．すなわち，$\text{COL}_{\text{項目}}$ と $\text{COL}_{\text{数値}}$ に属するそれぞれのセル要素について，その中心位置が同一の y 座標をもつものが 1 対 1 対応している状況を想定する．この状況においては，項目列中のあるセル要素内の文字列 $t^{\text{cell}}_{\text{項目}}$ に対して，数値列中の 1 つのセル要素内の文字列 $t^{\text{cell}}_{\text{数値}}$ を 1 対 1 で紐付けすることができる．これにより，各行 $(t^{\text{cell}}_{\text{項目}}, t^{\text{cell}}_{\text{数値}})$ を値にもつ 2 列のテーブルを構成することができる．

3.5 セル内文字列の分割

最後に，得られたテーブルの各行 $(t^{\text{cell}}_{\text{項目}}, t^{\text{cell}}_{\text{数値}})$ に対して分割処理を行う．3.4 節までの手法で抽出された文字列のペア $(t^{\text{cell}}_{\text{項目}}, t^{\text{cell}}_{\text{数値}})$ には，罫線の省略 (すなわち，結合セルの存在) により，本来複数の行に属すべき文字列が結合した状態で抽出されている場合がある (図 1 右上参照)．正確な情報抽出のためにはこれを適切に分割する必要があるが，特に $t^{\text{cell}}_{\text{項目}}$ においては，2 つの理由により分割を機械的に行うことが難しい．第一に，図 1 に示すように，抽出された文字列が折り返しを含んでいる場合があることである．このような状況では，単純な改行記号による分割は項目を過剰に出力してしまうことになる．第二に，行間のマージンが，PDF によって異なりうる点である．例えばマージンサイズを基準とした分割判定では，マージンの閾値を決める必要があるが，この数値を適切に決定することは難しい．また特に，図 1 に示すように折り返しによって生じるマージンと行間のマージンが等しい場合も存在し，マージンのみによってはこれを適切に分割することができない．上記の問題に対処するため，我々は項目の意味内容，および項目に紐づく数値情報を使用した機械学習ベースの分割手法を採用する．

項目セル内文字列の分割に取り組むにあたって，我々は問題を以下のような [SEP] トークンの分類タスクとみなす．すなわち，$t^{\text{cell}}_{\text{項目}}$ を [SEP] トークン部分で区切った系列を

$$X = (s_1, [\text{SEP}], s_2, [\text{SEP}], \dots, s_n) \tag{13}$$

と書いたとき，X 中の各 [SEP] トークンに対し分割・結合のラベルを適切に割り当てることを目的とする．

上記の問題に対し，我々は機械学習ベースの分類モデルを構築することで解決を目指す．すなわち，各 s_i について tokenizer を通して得られるトークン $t_{i1}, t_{i2}, \dots$ と [SEP] トークンによる系列

$$S = (t_{11}, t_{12}, \dots, [\text{SEP}], t_{21}, t_{22}, \dots, [\text{SEP}], \dots) \tag{14}$$

を入力に，各 [SEP] トークンに対し結合・分割の 2 値分類を行うモデルを，教師付き学習により構築する[7]．

本研究ではこのセル内文字列の分割判定モデルに図 3 に示すような BERT をベースとしたアーキテクチャを使用する．すなわち，tokenizer により得られた系列 S を BERT に入力し，各トークンの埋め込み表現を得た後，この埋め込み表現を分類器を通し [SEP] トークンのラベル確率を出力する．このとき，モデルは人手または後述する手法により自動でアノテーションされたラベル付きデータを使用し fine-tuning を行う．

3.6 数値列の利用

多くの場合，項目列中の文字列 $t^{\text{cell}}_{\text{項目}}$ の分割数は，それに紐づいている数値列中の文字列 $t^{\text{cell}}_{\text{数値}}$ から推察することができる．なぜなら，この $t^{\text{cell}}_{\text{数値}}$ 内の数値情報は通常 $t^{\text{cell}}_{\text{項目}}$ 内の各項目に 1 対 1 対応しており，また数値は折り返しを含まないゆえに単純な不要語の除去と [SEP] トークンのカウントによりその分割数 (および $t^{\text{cell}}_{\text{数値}}$ の正確な分割) を得ることができるためである．すなわち，我々は，期待される X の分割数 $n(X)$ を，X に紐づく数値列中の文字列 $t^{\text{cell}}_{\text{数値}}$ により

$$n(X) = \#_{[\text{SEP}]}(\text{PROCESS}(t^{\text{cell}}_{\text{数値}})) \tag{15}$$

[7] 文字列 s_i を tokenize する際，[SEP] トークンの位置によっては不適切な分割やモデルの語彙外のトークンが生じうるが，本研究の実験においてこの影響は軽微であった．より厳密な取り扱いは将来の研究課題とする．

のように取得する．ただしここで $\#_{[\mathrm{SEP}]}(t)$ は文字列 t 中の [SEP] トークンの数であり，PROCESS(t) は文字列 t に対する不要語の除去などの簡単な処理を表す．

我々は，この分割数の情報を弱いラベル付きデータの構築および上記の分割判定モデルの分割数への制約に利用し，アノテーションコストの削減および性能の向上を目指す．

弱いラベル付きデータの自動構築

3.5 節で導入したモデルではラベル付きデータにより fine-tuning を行う必要があるが，このようなラベル付きデータを十分な量用意することが難しい場合もある．そこで我々は数値情報 $t^{\mathrm{cell}}_{\text{数値}}$ より得られる分割数から，自明にラベルが定まるサンプルを教師データとして利用する手法を提案する．すなわち，X を抽出された文字列，$n(X)$ を $t^{\mathrm{cell}}_{\text{数値}}$ より示唆される分割数としたとき，自明にラベルが定まるサンプルとは X が以下の 2 通りのどちらかを満たす場合である: (1) $n(X) = 0$, (2) $n(X) = \#_{[\mathrm{SEP}]}(X)$．ここで (1) の場合は，$X$ 中の全ての [SEP] トークンに結合ラベルを，また，(2) の場合は，X 中の全ての [SEP] トークンに分割ラベルを付与することができる．

分割数への制約

上記の分割判定モデルでは，意味内容によるセルの分割を行うことができるが，現れる文字列が出現頻度の少ない場合や人間でも判断の難しいケースの場合は誤った分割となりうる．また上記の弱いラベル付きデータにより fine-tuning を行った場合，モデルは与えられた文字列に対し文字列内の全ての [SEP] トークンに同一のラベルを付与する傾向が生じ，正確な分割が得られにくい．さらに，モデルが予測した分割が $t^{\mathrm{cell}}_{\text{数値}}$ の分割数と不一致である場合，出力された項目と数値情報を 1 対 1 に対応づけることができない．

そこで，提案手法では数値列から得られる分割数 $n(X)$ を X 内の合計分割数とみなし，この制限の元で適切な分割ラベルを付与する．具体的には，モデルによる各 [SEP] $\in X$ の分割ラベルへのスコアのうち上位 $n(X)$ については分割ラベルを，その他は結合ラベルを付与することでこれを実現する．これにより，より正確なラベル付けや上述した同一ラベル出力の抑制に加え，分割された項目文字列と数値文字列を 1 対 1 で対応づけできるようになり，より正確なテーブル情報抽出ができると期待される．

4 実験

4.1 データセット

本研究では，表 1 に記載するように，開示されている 2 つの金融系 PDF 文書内の項目を対象とし，データセットを作成した (関係当事者データセット，大株主データセット)．その際，2020 年 1 月 1 日から 2020 年 12 月 31 日までの期間に開示された文書をモデル学習用データセット，2021 年 1 月 1 日から 2021 年 6 月 30 日までの期間を評価用データセットの作成に使用した．評価用データセットでは上記期間に開示された PDF 文書の一部をサンプルし，文書内で，表 1 記載のセクション中に存在する項目列および数値列中の値を人手で抽出する形で Ground-truth を作成した．このアノテーションは著名 4 名でそれぞれ別の PDF 文書群に対して行った．また，簡単のため，ページを跨ぐようなテーブルについては，項目名・数値名が存在するページ部分のみを評価対象とした．

モデル学習用データセットは 3.6 節 で述べた自動構築手法に加え，人手で [SEP] トークンにラベルをアノテーションしたデータセットも用意した．人手アノテーションは上記期間内の PDF で関係当事者データセットとして 2000 サンプル ([SEP] トークン数 4829)，大株主データセットとして 800 サンプル ([SEP] トークン数 3180) について行った．また，同期間内の PDF で自動構築したデータセットのサンプル数はそれぞれ 1483 サンプル ([SEP] トークン数 2772)，3225 サンプル ([SEP] トークン数 8606) であった．実験では，自動構築したデータセットにより fine-tuning したモデルと，人手アノテーションしたデータセットにより fine-tuning したモデルの性能をそれぞれ評価した．

4.2 評価指標

実験の評価指標として，(マイクロ)Precision， Recall, F1 スコアを採用した．すなわち，ある PDF i に対し，それがもつ真の項目-数値のペアの集合を T_i，また手法により抽出された項目-数値のペアの集合を P_i としたとき，Precision，Recall を以下のように定義した．

$$\mathrm{Precision} = \frac{|\sum_i T_i \cap P_i|}{|\sum_i P_i|}, \tag{16}$$

$$\mathrm{Recall} = \frac{|\sum_i T_i \cap P_i|}{|\sum_i T_i|}. \tag{17}$$

ここで $|A|$ は集合 A の要素数を，$A \cap B$ は集合 A および B の共通要素の集合を表す． 共通要素数では，T, P の各要素に文字列正規化[8]を行った後，完全にペアが等しくなったもののみを共通要素として数えた．

4.3 ベースライン

我々の手法との比較対象として，PDF テーブル抽出を対象とした標準的なオープンソースである Tabula[9] と Camelot[10]を採用した．実験では，それぞれが出力するテーブルデータのうち，表 1 で指定した項目列・数値列名を列に含むテーブルを抽出し，そこに含まれる項目列・数値列をもって手法の予測とした．本実験では両者ともデフォルトのパラメータ設定を採用した．

4.4 実験設定の詳細

分割判定モデルで使用する BERT としては，東北大学が公開している日本語 Wikipedia による事前学習済みモデル (bert-base-japanese-whole-word-masking)[11] を使用した．BERT により得た各トークンごとの埋め込みを，Dropout 層および 1 層の全結合層を通した後，

[8] Unicode 正規化 (NFKC)，大文字小文字の統一を行った．
[9] https://github.com/chezou/tabula-py
[10] https://github.com/camelot-dev/camelot
[11] https://github.com/cl-tohoku/bert-japanese

表 1: データセット

データセット	文書	セクション	項目列名	数値列名	評価用項目数	学習用データ数 (人手 / 自動)
関連当事者	株主総会招集通知	関連当事者との取引に関する注記	科目	期末残高	1485	2000 / 1483
大株主	有価証券報告書	大株主の状況	氏名又は名称	所有株式数	1131	800 / 3225

表 2: 実験結果

手法	関連当事者			大株主		
	Precision	Recall	F1	Precision	Recall	F1
Tabula	0.540	0.115	0.190	0.410	0.682	0.512
Camelot	0.720	0.580	0.642	0.722	0.520	0.604
自動アノテーション	0.997	0.992	0.995	0.935	0.908	0.921
人手アノテーション	0.997	0.992	0.995	0.995	0.966	0.981

[SEP] トークンに対して softmax 関数を適用し，[SEP] トークンに対するラベル確率とした．学習はクロスエントロピー損失の最小化によって行い，最適化アルゴリズムは AdamW[4] を採用した．実験では学習率を 2×10^{-5} (学習率スケジューリングなし)，ドロップアウト率を 0.1，バッチサイズを 32，トークン最大長は 256 にそれぞれ設定した．学習用データセットは訓練用データセット, 検証用データセットとして 8:2 に分割し，訓練用データセットを使用してモデル訓練を行った．また学習のイテレーションは検証用データセットの分割・結合ラベルに対する正解率に基づき，early-stopping により打ち切った．

4.5 結果

表 2 に実験の結果を示す．関連当事者データセットでは, 人手アノテーションの場合, Precision が 0.997(評価用項目数:1481 行/1485 行)，Recall が 0.992(1481 行/1493 行)，F1 スコアは 0.995 となり，ベースラインを F1 スコアで 0.35 ポイント程度上回る結果となった．また，本データセットにおいては，自動アノテーションによるモデルにおいても，Precision が 0.997(1481 行/1485 行)，Recall が 0.992(1481 行/1493 行)，F1 スコアは 0.995 となり，人手アノテーションと同性能の結果が得られた．

大株主データセットでは，人手アノテーションの場合，Precision は 0.995 (1093 行/1098 行)，Recall は 0.966 (1093 行/1131 行)，F1 スコアは 0.981 となり，ベースラインを F1 スコアで 0.38 ポイント程度上回る結果となった．一方，自動アノテーションによるモデルにおいても，Precision が 0.935 (1027 行/1098 行)，Recall が 0.908 (1027 行/1131 行)，F1 スコアが 0.921 となり, 人手アノテーションと遜色ない結果が得られた．

4.6 考察

実験結果より，Tabula や Camelot などの一般的な PDF テーブル抽出ツールと比べて，数値-項目間の 1 対 1 対応を正解とした時の性能は大幅に改善された．これは，従来のツールでは難しかった適切なセルの分割がなされた結果と考えられ，金融系 PDF 文書における提案手法の有効性を示していると言える．

関連当事者データセットにおいて，適切な情報取得ができなかったテーブルは，図 4a のように 1 つの項目に対して複数の数値が紐づくようなテーブルや，反対に図 4b のように複数項目に対して 1 つの数値が紐づくようなテーブルであった．本手法では，1 項目に対して 1 数値が紐づくことを前提としているため，上記のようなテーブルは対応できなかった．

大株主データセットにおいては，特に自動アノテーションによる結果が関連当事者データセットよりも劣る性能となった．これは，自動アノテーションを行えたデータ中, 分割ラベルをもつデータが非常に少なかったことが原因であると考えられる (3225 サンプル中 85 サンプル)．実際，[SEP] トークンのラベルが自明でないサンプル ($0 < n(X) < \#_{[\mathrm{SEP}]}(X)$ なもの) に対する Precision，Recall はそれぞれ 0.736 (195 行/265 行)，0.714 (195 行/273 行) と低く[12]，BERT がこの自動アノテーションデータで十分に学習されているわけではないことが示唆された．この改善に向けては，例えばランダムに抽出した 2 つのサンプルを，分割ラベルを付与した [SEP] トークンにより結合し，データ増強を行うといった手法の導入が考えられる．このようなデータ増強手法のより詳細な検証については，今後の一つの発展の方向性としておく．

最後に，大株主データセットにおけるテーブル検出の失敗例を図 4c に示す．大株主データセットではページ下部にまでテーブルが存在することが比較的多くあったが，これが PDF 中に存在するその他の線分 (多くは不可視であり, ページを囲うように存在するもの．図 4c ではオレンジ色の線分で表現) と交差した結果，不適切なクラスターが作成される状況がしばしば生じた．この抑制には，線分要素のフィルタリングや，より洗練されたテーブル検出ルールの導入が必要であると考えられる．

5 結論

本研究では，PDF 形式のテーブルに対して,「項目」列とそれに紐づく同一テーブル内の「数値」列の関係を考慮した上で情報抽出を行った．これは，PDF の文

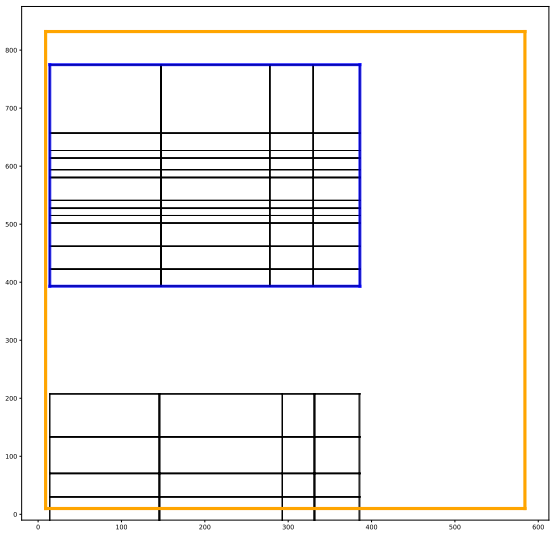
[12] 人手アノテーションの場合，それぞれ 0.985 (261 行/265 行)，0.956 (261 行/273 行) であった．

会社等の名称	…	取引金額(千円)	科目	期末残高(千円)
△△(株)	…	- 3,300 2,00 5,000	1年内回収予定の関係会社長期貸付金 関係会社長期貸付金 未収入金	- 3,300 17,000 300

(a)

会社等の名称	…	取引金額(千円)	科目	期末残高(千円)
□□(株)	…	27,000 70,000 40 200 35,000	売掛金 買掛金 短期貸付金 未収入金	2,000 600 35,000

(b)



(c)

図 4: 本手法で抽出できなかった例

書内から抽出した線分情報や文字情報を用いてテーブル領域を検出し，領域内のセル及び列の構成やセル内文字列の分割を行うことで実現した．セル内文字列の分割には BERT を用い，「数値」列を利用した分割数の制約を課すことでより正確な分割を行った．我々は以上の手法を 2 つの日本語金融文書により作成したデータセットにより評価し，結果，一般的な PDF テーブル抽出ツールと比べ大幅な性能の改善を確認した．

6 今後の課題

本研究のテーブル領域検出では，PDF 内の線分要素の交点や接点を利用しているため，罫線が大きく省略されているとき，検出に失敗する場合がある．今回使用した日本語金融文書データセットにおいては，このような大きな罫線の省略があるテーブルはほとんどなく，今回の手法でも高い精度でテーブル検出を行うことができた．しかし，文書によっては，罫線が大きく省略されたテーブルが割合の多くを占める場合もありうる．特に英語圏のテーブルはしばしば罫線が大きく省略されるため，検出できない状況が多く存在すると予想される．このような状況に対応するためには罫線に依らない表検出手法 (例えば Schreiber ら [5] のような画像からの検出手法) を実施することが必要であろう．

また，我々の手法で項目-数値間の 1 対 1 対応した一覧化を実現することができたが，会社間での比較は項目における表記ゆれや数値の単位の統一化をする必要がある．例えば，金融文書の場合，数値の単位は「百万円」や「千円」などと企業間で異なり，このままでは企業間で数値の合計を算出することはできない．これについては，今回のテーブル内の情報抽出に併せて，テーブル周辺の情報も抽出し，単位らしい文字列をキーワードマッチングすることで実現できると考えられる．

参考文献

[1] Liu, Z., Huang, D., Huang, K., Li, Z., and Zhao, J.: FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining, *IJCAI2020*, pp. 4513–4519 (2020)

[2] 高野 海斗, 酒井 浩之, 北島 良三：有価証券報告書からの事業セグメント付与された業績要因文・業績結果文の抽出, 人工知能学会論文誌,Vol. 34, No. 5, p.wd-A_1–22,(2019)

[3] Devlin, J., Chang, M. W., Lee, K., and Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding, *NAACL-HLT2019*, Vol. 1, pp. 4171–4186 (2019)

[4] Loshchilov, I., and Hutter, F.: Decoupled weight decay regularization, *ICLR2019*, (2019)

[5] Schreiber, S., Agne, S., Wolf, I., Dengel, A., and Ahmed, S.: Deepdesrt: Deep learning for detection and structure recognition of tables in document images, *ICDAR2017*, Vol. 1, pp. 1162–1167, IEEE (2017)