学術論文の表の解析によるグラフの自動生成の一手法

山田　凌也† 　太田　学†† 　高須　淳宏†††

† 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
†† 岡山大学大学院自然科学研究科　〒 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
E-mail: †ppeq9d0u@s.okayama-u.ac.jp, ††ohta@de.cs.okayama-u.ac.jp, †††takasu@nii.ac.jp

あらまし　実験系の論文において，どのような実験を行い，どのような結果が得られたかという情報は非常に重要である．特に表は実験結果を表すのに頻繁に用いられる．しかし，表は正確な数値を読み解くには適しているが，数値の比較や変化を視覚的に読み取るには不向きである，そこで，本研究では表からグラフを自動生成する手法を提案する．具体的には，論文を XML ファイルに変換して，それを基に表の構造を推定する．また，視覚的に理解しやすいグラフについて検討する．

キーワード　表構造解析，XML データ，グラフ化，自動変換

1. はじめに

近年，CiNii(注1) や Google Scholar(注2) 等の学術論文データベースの充実により，膨大な数の論文を手軽に入手できるようになった．しかし，論文は機械が処理できるデータと異なり，人が読まなければ有効に活用できない．そのため論文中から有用な情報を自動で抽出するために論文の構成要素を分類する様々な研究 [1] が行われている．その中でも表は情報をコンパクトかつ効率的に表示でき，実験結果を表す手段として頻繁に用いられる．そのため，これまでに学術論文中の表の解析，抽出に関する研究 [2,3] が行われてきた．一方，値の変化を確認する場合や，値同士を比較する場合など，表よりもグラフのほうが視覚的に理解しやすい場合がある．

そこで，本稿では論文 PDF を変換した XML 中の表を解析し，その構造を推定することによってグラフを自動生成する方法を提案する．

本稿の構成は次の通りである．まず，2. 節で表の構造解析と論文中の情報の可視化に関する研究を紹介し，3. 節で本研究で行う表の構造解析について説明する．つづく 4. 節でグラフの自動生成について説明し，5. 節では，複数のグラフを集約し，1 つにまとめる方法について説明する．6. 節で表の構造解析とグラフ化に関する評価実験について述べる．7. 節では，実験結果について考察し，8. 節でまとめる．

2. 関連研究

2.1 表構造解析

Kieninger らは表の構造解析のために T-Recs [4] と T-Recs++ [5] というシステムを実装した．T-Recs は文章中の全単語の矩形領域の位置情報を入力とし，表を再現するものである．彼らは各矩形領域を垂直方向に前後の行へと射影し，重なる単語を再帰的に結合することで，列を検出した．

一方，T-Recs++は T-Recs が適切に再現することのできなかったビジネス文章を対象とする．T-Recs++は T-Recs の手法に加えて表の典型的な類似性を考慮して表の構成を予測するシステムである．T-Recs++は T-Recs と比較して，表の行の適合率が 0.45 から 0.86 に，再現率が 0.77 から 0.96 に向上した．一方，表の列の適合率は 0.34 から 0.91 に，再現率は 0.79 から 0.97 に向上した．

Yildiz ら [6] は PDF から表を抽出し，セルに分割するシステム pdf2table を考案した．こちらも Kieninger [5] と同じく単語を囲む矩形領域を利用するが，垂直方向ではなく水平方向に結合する．彼らは表には複数の行が必要であると考え，表構造解析のために複数の行を含むブロックを作成する．その後，行を列に分解，垂直方向に結合していく．こうすることで表の罫線を引き，表をセルに分割する．実験用データセットとして単純な表 50 件と複雑な表 100 件を用い，セルの分割に関してそれぞれで再現率と適合率を算出した．結果は単純な表で再現率が 0.88，適合率が 0.97，複雑な表で再現率が 0.81，適合率が 0.83 となった．

Shingrov ら [7] は単語を囲む矩形領域を利用して表の構造を再現する手法を提案した．彼らの手法は 3 段階に分かれている．まず，前処理として論文中からテキストチャンクと罫線を生成する．つづいて，テキストチャンクを組み合わせてテキストチャンクの集合であるテキストボックスを再現する．最後にテキストボックスから行，列を構築する．彼らはその際のパラメータやセルの分割手法を変えて 2 つのシステムを構築した．これらを用いて ICDAR 2013 の表構造認識コンペティション(注3) において提供された表の各セルの矩形領域の座標のデータセットについてセルの罫線を予測し，再現率が 0.92，適合率が 0.95，F 値が 0.94 となった．

(注1)：http://ci.nii.ac.jp/
(注2)：https://scholar.google.co.jp/
(注3)：http://www.tamirhassan.com/competition.html

2.2 論文中の情報の可視化

村田ら [8] は，自然言語処理に関する論文の概要から重要な情報を抽出する手法を提案した．彼らは重要な情報を「精度表現」,「自然言語処理における分野」,「言語名」,「組織・人名」の 4 つに分類した．これらの情報は関連論文の検索や，自然言語処理分野の論文のサーベイの自動構築等の様々な目的に役立つ．彼らは，教師あり機械学習を用いたテキストチャンカである YamCha(注4) を利用してこの 4 分野の単語を抽出した．抽出精度は，抽出した情報の完全一致で F 値の平均が 0.8，部分一致で 0.85 であった．また，村田らは抽出した各分野の単語を表やグラフで可視化する手法を提案した．表の 1 行目に 4 つの分類，1 列目に論文のタイトルをおくことで重要情報をまとめ，論文の特徴や状況を一度に把握できるようにした．さらに，主要な分野における論文の分布と論文に含まれる言語名の分布をグラフ化する可視化ツールを構築した．これにより自然言語処理分野の論文のサーベイや，各言語についての研究動向の理解に役立つと報告されている．

また，平井ら [9], [10] は論文中の実験に関連のある論文構成要素を実験情報と呼び，これらの情報を論文から自動抽出した．彼らは論文 PDF を XML ファイルに変換し，XML ファイルから論文構成要素を抽出した．その後ルールと Conditonal Random Feild (CRF) [11] によってそれぞれ実験情報を抽出した．NTCIR9(注5) の Spoken Documet タスクの論文の 10 件に対して評価実験を行い，論文構成要素に関するタグ付与実験は再現率，適合率，F 値がいずれも 0.94 程度となった．実験情報抽出実験は, ルールでは再現率が 0.83，適合率が 0.73，F 値が 0.78 となり，CRF では再現率が 0.66，適合率が 0.87，F 値が 0.74 となった．また論文構成要素から表を抽出し，1 行ずつ処理することによってグラフを自動生成した．実験では NTCIR9 の Spoken Documet タスクの論文の実験情報の 27 件の表のグラフ化を試み，そのうち 18 件の表のグラフ化に成功した．

しかし，平井ら [9,10] の手法によってグラフ化できるのは単純な表のみであった．例えば，1 つのセルの中の文字列が複数行であったり，1 つのセルが表の複数行にわたり存在していたりするような表はグラフ化することができなかった．そこで本研究ではセル単位で表を解析し，複雑な表のグラフ化を試みる．

3. 表の構造解析

表を論文から抽出するにあたって，まず pdf2xml(注6) を使用して XML ファイルを作成する．しかし，この XML ファイルでは表の罫線の情報が失われており，グラフ化するには表の構造を推定する必要がある．

本稿では，平井ら [10] の提案した表の行を利用した表の解析を基に，セルを利用した表の解析を提案する．まず，3.1 部で平井らの手法である表の行を利用した表の解析について述べる．次に 3.2 部でセルを利用した表の解析について述べる．

(注4) : http://chasen.org/ taku/software/yamcha/
(注5) : http://research.nii.ac.jp/ntcir/ntcir-9/index-ja.html
(注6) : http://soueceforge.net/projects/pdf2xml

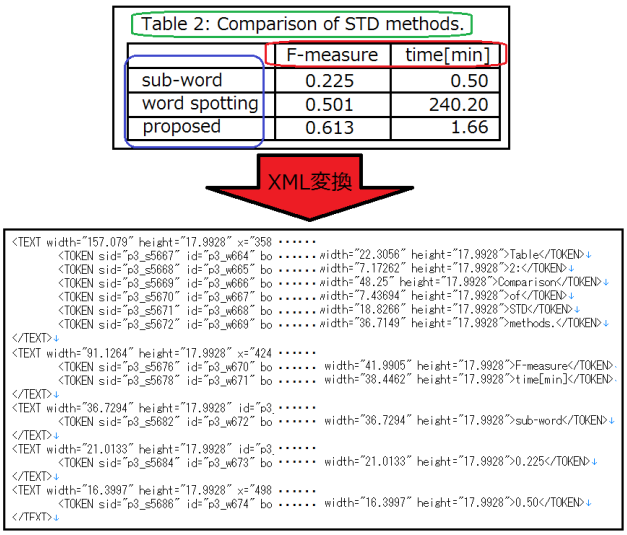

図 1: 論文 [12] の表と XML ファイルの一部

3.1 行を利用した表の解析

平井ら [10] は表からグラフを自動生成するにあたって，表を 1 行ずつ解析した．

まず，論文 PDF から表を抽出するために，pdf2xml によって XML ファイルを作成する．図 1 に論文中の表の例と pdf2xml によって出力されるその XML ファイルの一部を示す．XML ファイルでは各単語はトークンとして表され，トークンを囲む矩形領域の左上の頂点の座標や幅，高さ，トークンのフォントなどが記されている．これらのトークンは論文 PDF の左上を原点とし，そこから右下に向かって並べられる．

平井らの手法では，この XML ファイルからトークンを行ごとに抽出し，それを解析することで表の構成要素を推定する．

表の構造解析にあたって，表の構成要素のうち表のタイトルを「表キャプション」，列の名前を表している行を「ヘッダ行」，行の名前を表している列を「ヘッダ列」，それ以外の数値を「データ」と呼ぶ．図 1 の表においては，緑色で囲んだ要素が表キャプション，赤色で囲んだ要素がヘッダ行，青色で囲んだ要素がヘッダ列である．平井らは表の 1 行目をヘッダ行，2 行目以降の数値をデータ，2 行目以降で数値と判定されなかったものはヘッダ列とし，表を解析した．そのため，それに外れる表は解析できなかった．

3.2 セルを利用した表の解析

平井らは XML 中の 1 つのトークンを 1 つのセルとみなした．しかし，平井らの手法ではグラフ化出来ない表も存在した．そこで本稿では行単位ではなくセル単位で表の構造を解析する方法を提案する．

また，表の解析にあたって提案手法では，表中のヘッダ行とヘッダ列はそれぞれ数字のセルを含まない行と列としている．また，ヘッダ行，ヘッダ列にあたる行，列が存在しない場合は 1 行目，1 列目をそれぞれヘッダ行，ヘッダ列として扱う．表キャプション，データの定義は 3.1 節において定義したものと同様である．

3.2.1 項では文字列を利用した表の罫線の予測について述べ，

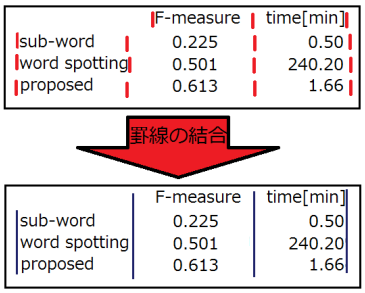

図 2: 垂直方向の罫線の予測の例

3.2.2 項では垂直方向の罫線の再予測について述べる．また，3.2.3 項では表の特徴を利用した罫線の修正について述べる．

3.2.1 罫線の予測

まず，左右に隣接するトークンを結合することでセルを作成する．トークンは PDF 中での空白によって分割されている．トークン間に存在する空白にはセル内の単語間の空白とセル間の空白が存在する．同じセルに含まれるトークンを結合するために，セル間の空白は単語間の空白よりも大きいと考え，直後の文字よりも大きい空白はセル間の空白であるとする．トークン間の空白が単語間のものであると判定されたトークンは結合し，1 つのトークンとする．

次にこのようにして作成したセルを基に罫線を推定する．表の罫線は通常隣接するセルの文字列の重心の中点を通る．pdf2xml は論文を左上から右下の方向に変換していくため，基本的に表のセルは 1 行目から順に変換される．そのため，XML ファイル中から 1 行ずつセル内の文字列を抜き出すことは容易である．

そこで，XML ファイル中のセルの位置を基に各行内でその行に存在するセルの垂直方向の罫線を予測する．その後，隣接する行の一番近い罫線と結合していくことで垂直方向の罫線を作成し，その罫線の位置によってそれらの罫線の間にあるセルがどの列に属するかを予測する．図 1 の表に対して重心を利用して垂直方向の罫線を予測した例を図 2 に示す．図 2 の上の図が各行での罫線の予測であり，予測された罫線は赤線で示した．下の図がその罫線を隣接する行の一番近い罫線と結合し，垂直方向の罫線を予測した結果である．

列の予測後，その結果を基に各列でその列の中に存在するセルの水平方向の罫線を予測する．その後，隣接する列の一番近い水平方向の罫線と結合する．セルの上下の水平方向の罫線の位置を予測することにより，そのセルがどの行に属するかを決定する．

3.2.2 罫線の再予測

3.2.1 項で示した手法により垂直，水平両方向の罫線を予測することができるが，pdf2xml では単語をトークンへと変換する際に，複数行にわたるセル内の文字列はそれよりも y 座標が小さい文字列に先んじて変換される．そのため，垂直方向の罫線の予測位置と実際の位置による垂直方向の罫線の位置にずれが生じる．そこで水平方向の罫線の予測後，予測したセルの行と列を基にもう一度垂直方向の罫線を予測することでこのずれを解消する．pdf2xml により y 座標が大きいにも関わらず先んじて変換される表の例を図 3 に示す．図 3 中の破線は論文中の

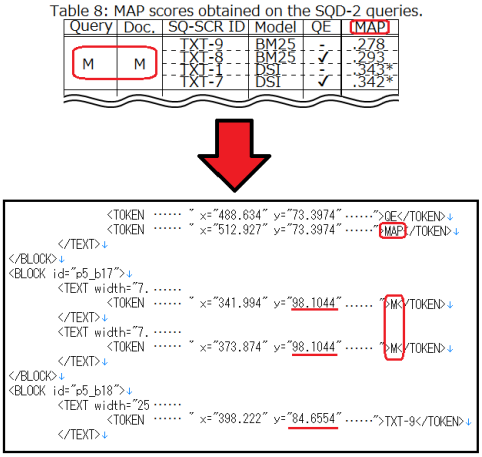

図 3: y 座標が大きいにも関わらず先にトークンに変換される例 [13]

system ID	micro ave.		macro ave.				
	Actual F.	Max F.	Actual F.	Max F.	MAP	ATWV	MTWV
ALPS-1	0.5793	0.6054	0.5965	0.6646	0.7445	0.4641	0.6420

図 4: 論文 [14] 中の表

		micro ave.		macro ave.			
System ID	Actual F.	Max F.	Actual F.	Max F.	MAP	ATWV	MTWV
ALPS-1	0.5793	0.6054	0.5965	0.6646	0.7445	0.4641	0.6420

図 5: 図 4 の表構造の予測罫線 (失敗例)

表では省略されているが，罫線の予測の際は存在しているものとして予測された罫線である．以降ではこの予測した罫線に基づいて説明する．y 方向を優先して行指向で解析すれば，表の 1 行目の最後である “MAP” の次にはその行のすぐ下に位置する “TXT-9” が出力されると考えられる．しかし，図 3 ではその下に存在している “M” が 2 つ出力されている．

この XML ファイルを用いた 1 度目の垂直方向の罫線の予測では，2 つの “M” の左右の罫線は 2 行目のセルの罫線の予測の時に予測される．これは，図 3 の XML ファイルでは “M” は両者ともに “MAP” と “TXT-9” の間，つまり，2 行目の先頭に存在していると判定されるためである．一方，水平方向の罫線を予測した時，各セルの上の罫線は実際の座標に従って引かれるため，“M” の上の罫線は両者とも表の 2 行目の “TXT-9” とその下の “TXT-8” の間に引かれ，“M” は 3 行目に属すると予測される．つまり，“M” の上端を表す水平方向の罫線よりも上の行に左右の罫線が存在していることになり，矛盾が生じる．

そこで，水平方向の罫線の予測によって予測した各トークンが何行目に属するかを基に垂直方向の罫線を再予測する．すると，“M” の左右の罫線は 3 行目における垂直方向の罫線の予測の際に予測されることなり，矛盾が解消される．

このように罫線を予測するが，文字列の重心による単純な罫線の予測と垂直方向の罫線の再予測だけでは，正しくセルの罫線を予測できない表が存在する．図 4 と図 5 にこの方法で予測に失敗した表とその予測された構造をそれぞれ示す．

図 5 を見ると，1 行目のセル “micro ave.”，“macro ave.” は

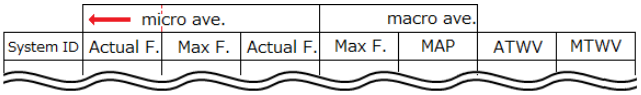

図 6: 制約条件を加えて予測した罫線

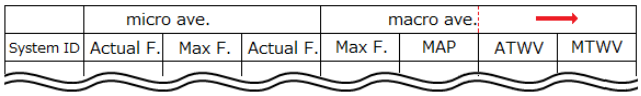

図 7: 引かれるべき罫線を引き，セルを拡張した後の罫線

双方ともその範囲が適切に予測できていない．例えば，“micro ave.” の左の罫線は，各行での垂直方向の罫線の予測の後，左から 2 本目の罫線と結合されるべきである．しかし，“micro ave.” の左の罫線は 2 本目よりも 3 本目に近いため，間違った罫線と結合される．

これを解決するために，制約条件として罫線は他の文字列の上を通らないという制約条件を加える．また，この制約条件を適用するのは罫線の予測において複数行，列にわたるセルを含む行，列と判定されたセルに限る．制約条件を加えて予測した罫線の一部を図 6 に示す．破線は図 5 における “micro ave.” の左罫線である．この制約条件により当該罫線を適切な位置に予測することが出来た．

3.2.3 罫線の修正

文字列の重心を用いて罫線を予測するが，文字列の大きさに対してセルが極端に大きい場合，罫線を正しく予測できない．例えば，図 4 で “macro ave.” のセルの大きさは文字列のそれに比べて大きいため，“micro ave.” との間の予測罫線の位置がずれる．また，“macro ave.” の右側の垂直方向の罫線の位置も適切ではない．そこで次のような表の特徴を利用して予測罫線を修正する．

（1） 表を囲む罫線，ヘッダ行およびヘッダ列とデータを区切る罫線は必ず存在する

（2） 中央揃えの場合，複数の行，列にわたるセルの文字列の重心は隣接する行，列の同範囲にある各セル中の文字列の重心と一致する．

罫線の予測に際して表を囲む罫線やヘッダ行，ヘッダ列とデータを区切る罫線が予測されなかった場合は，そこに罫線が予測されるように隣接するセルを拡張する．例えば，図 6 においては，1 行目の右上に表を囲む罫線が予測されるように拡張する．しかし，この場合のセルの拡張の方法は 2 行目のセルの上への拡張と，“macro ave.” のセルの右への拡張の 2 つが考えられる．拡張する方向が複数存在する場合は拡張する方向を決定するために，次のようなコストを定義する．

$$\text{周期性コスト} = \sum_{i=1}^{n} |E - R_i| \tag{1}$$

$$\text{隣接コスト} = |E - R_{prev}| + |E - R_{next}| \tag{2}$$

n は表の行または列の数である．また，E は拡張するセルのある行/列の罫線の数，R_i は i 行/列目の罫線の数，R_{prev}，R_{next} はそれぞれ直前，直後の行/列の罫線の数である．

式 (1)，(2) によって定義したコストは行の罫線の数を用いた場合，他の行の罫線の数に近いほどその値が小さくなる．しかし，1 行目は水平方向にセルが結合され，他の行とは異なったセルの構成となることも多い．そこで，1 行目においては行ではなく列の罫線の数を用いてセルを拡張する．

セルを広げうるすべての方向について式 (1)，(2) のコストを計算し，その和が最も小さくなる方向にセルを拡張する．図 6 において “macro ave.” のセルを右側へと拡張するコストと “ATWV” と “MTWV” のセルを上へと拡張するコストを計算し，拡張した罫線の一部を図 7 に示す．図 7 の破線は図 6 の “macro ave.” の右罫線である．この拡張により適切な位置に予測罫線が移動した．図 7 を見ると，表の左上には空白のセルが存在しているが，本研究では左上のセルのみ空白を許している．

次に制約 (2) を用いて罫線を修正する．この修正が適用されるのは，セルの文字列が中央揃えである場合のみである．中央揃えであるかどうかは上下の隣接するセルの文字列の座標を比較することによって判定する．例えば，水平方向の中央揃えの判定は，判定対象となるセルと重心を比較するセルの左端と右端の座標を比較し，左端，あるいは右端の x 座標の差が 1 以下であるならばそれぞれ左揃え，右揃えとし，それ以外を中央揃えとしている．

図 7 においては，複数列にわたるセルである “micro ave.” と “macro ave.” が存在する．この 2 つのセルについてそれぞれ，その文字列の重心とそのセルの範囲にある下の行の文字列の重心の差が小さくなる方向に 2 つのセルの罫線を移動させる．この場合，移動の候補は “micro ave.” の左端の罫線の左右の移動と “macro ave.” との間の罫線の左右の移動で 4 つある．例えば正解である 2 つのセルの間の罫線の左への移動の場合，“micro ave.” の重心と 2 行 2 列目の “Actual F.”，2 行 3 列目の “Max. F” の重心の平均の差，および “macro ave.” の重心と 2 行 4 列目の “Actual F.” から 2 行 8 列目の “MTWV” までの重心の平均の差が他の移動候補の計算結果よりも小さくなる．よって，この罫線を左に動かす．

4. グラフの自動生成

3. 節の方法で構造を解析した表のグラフを自動生成する．本稿では Google Chart API を用いて HTML としてグラフを表示する．Google Chart API は数値データや軸の情報などから自動的にグラフを生成し，Google Chrome 上で見ることができるサービスである．

グラフ化の例として，図 1 のグラフ化結果を図 8 に示す．

表とグラフの要素を囲んだ色は対応している．表キャプションをグラフ名，ヘッダ行の要素を凡例，ヘッダ列の要素をラベル名と定める．また，グラフの左右にある縦軸の目盛りのタイトルは，どの凡例の目盛りであるかを表している．

ここではまず，4.1 部でグラフ化の対象とする表について説明し，つづく 4.2 部でグラフ化で用いるグラフの種類について述べる．4.3 部では凡例とラベルの取り扱いについて述べ，4.4 部ではグラフ中のスケール調整について述べる．

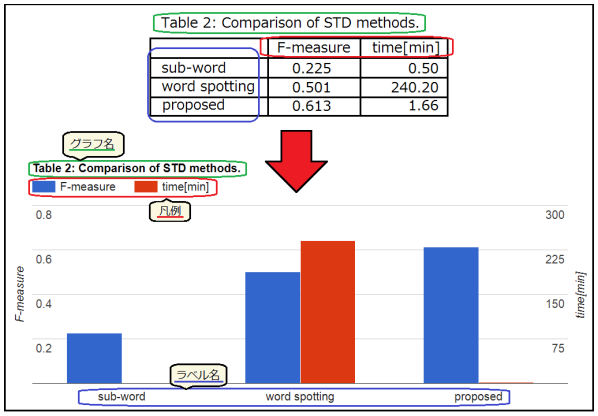

図 8: 図 1 の表 [12] のグラフ化結果

Task	Task1
Spoken Doc.	ref_word
Query	n queries
Training Data	corpus1

(a) 数値のセルがない表の例

Eassay	Pri-oryity	ROUGE-N 1	ROUGE-N 2	N/A
Task1				
Secondary Exam	1	0.36	0.11	0/5
	2	0.41	0.13	0/5
	3	0.39	0.12	0/5
Task2				
Secondary Exam	1	0.36	0.11	0/5
	2	0.41	0.13	0/5
	3	0.39	0.12	0/5

(b) セルが長方形でない表の例

図 9: グラフ化対象から除外した表の例

4.1 グラフ化対象の表

表の中にはグラフになりえないものや，グラフ化しないほうが良いものが存在する．グラフの自動生成の前処理として解析した表をグラフ化するかどうかをまず判定する．平井らは以下のルールのいずれかに該当する表はグラフ化しなかった．

- 表中に数値のみが記されているセルが存在しない
- 表の大きさが 2 行 2 列以下である

本稿では，このうち 1 つ目のルールのみを採用し，2 行以下かつ 2 列以下の表はグラフ化する．これは，小さい表であってもグラフ化することによって視覚的に理解しやすくなると考えたからである．

一方，空白のセルが存在する表や 1 つの表の中に複数の表をまとめた表，セルの形が長方形でない表はグラフ化対象から除外した．除外した表の例を図 9 に示す．

4.2 グラフの種類

本稿では，棒グラフと折れ線グラフで表をグラフ化する．以下に棒グラフと折れ線グラフの選択基準を示す．

- 折れ線グラフ：ヘッダ列が数値である表
- 棒グラフ：ヘッダ列が数値でない表

この選択基準に従って折れ線グラフへと変換される表とその変換結果を図 10 に示す．図 10 の表はヘッダ行が数値となっているが，この表はグラフ化の際に行と列の転置が行われる．行と列の転置については，4.3 部で詳しく述べる．

図 10 を見ると ”Score ” の異なる値に対する他の指標の変化の様子が見て取れる．

4.3 凡例とラベルの判定

グラフを理解する上で凡例とラベルは非常に重要である．

本稿では，凡例に評価指標などの比較基準，ラベルに手法名やデータセットなどの比較対象を割り当てると定める．

Table 1: Processing time of our methods in the CORE experiment.

Score	1.00	0.96	0.92	**0.886**	0.88	0.84	0.80
Recall	26.5	27.7	33.8	**43.8**	44.4	53.9	65.6
Precision	88.0	88.4	90.3	**84.6**	68.8	57.3	24.7
F-measure	40.8	42.1	49.2	**57.0**	54.0	55.5	35.6
Time[ms]	0.00	0.00	0.00	**0.62**	0.96	2.18	10.92

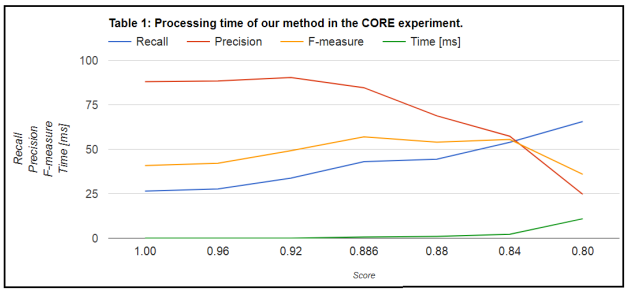

図 10: 折れ線グラフへと変換される表 [15] とグラフ

Table 5: NUL score of formal run

Phase	Exam	Priority of runs 1	2	3
1	National Center Test(1999)	43	**46**	36
2	Benesse mock exam (2015 Jun/All/out of 175)	121	121	118
(2)	Benesse mock exam (2015 Jun/Pattern 1)	**76**	76	76
(2)	Benesse mock exam (2015 Jun/Pattern 2)	64	64	61
3	National Center Test(2011)	65	65	**68**
3	Benesse mock exam (2014 Sep/All/out of 125)	77	76	76
(3)	Benesse mock exam (2014 Sep/Pattern 1)	**60**	57	60
(3)	Benesse mock exam (2014 Sep/Pattern 2)	58	60	54

図 11: 複数行と列にわたる凡例とラベルが存在する表 [16]

4.3.1 項では行と列を転置する場合について述べる．4.3.2 項ではヘッダ行やヘッダ列が複数行，列に存在する時のグラフ化について述べる．

4.3.1 行と列の入れ替え

表を解析した情報を基にグラフを生成するが，“Recall” や “Precision” などの評価指標を凡例とラベルのどちらにするかは通常グラフの作成者に委ねられる．

本稿では，デフォルトで表のヘッダ行をグラフの凡例に，表のヘッダ列をグラフのラベルに割り当てる．しかし，この割り当てで評価指標がラベルとなる場合，表の転置により評価指標を凡例としてグラフ化する．

例えば，図 10 の表は転置しなかった場合，ヘッダ行である 1 行目が凡例，ヘッダ列である 1 列目がラベルとなる．しかし，この表はヘッダ列に評価指標である “Recall” や “Precision” が存在する．そこで，凡例を評価指標とするためにこの表は転置してから，グラフを生成する．

4.3.2 複数行，列に存在するヘッダ行，ヘッダ列の結合

単純な表はヘッダ行とヘッダ列がそれぞれ 1 行，1 列であるが，ヘッダ行が複数行からなる表や，ヘッダ列が複数列からなる表も存在する．ヘッダ行，ヘッダ列ともに複数行，列に存在する表の例を図 11 に示す．

図 11 を見るとヘッダ行は 2 行，ヘッダ列も 2 列である．このような表は平井らの手法ではグラフ化することができなかった．本稿でもデフォルトではヘッダ行，ヘッダ列はそれぞれ 1 行，1 列である．そこで，表中の数値が含まれない行や列をグラフ化の際にまとめてそれぞれ凡例，ラベルとすることにより

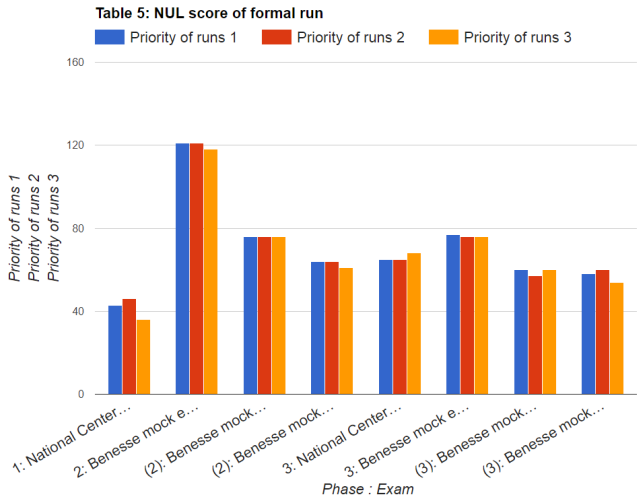

図 12: 複数の行または列にわたる凡例とラベルをまとめたグラフ

それぞれ 1 行，1 列のヘッダ行，ヘッダ列とする．

図 11 で凡例とラベルをまとめてグラフを自動生成した結果を図 12 に示す．

この例では複数行にわたるヘッダ行である “Priority of runs” と “1” は 1 つの凡例として “Priority of runs 1” のようにまとめる．また，複数列にわたるヘッダ列である “Phase” の列と “Exam” の列を “Phase: Exam” のようにまとめる．

4.4 スケール調整

本稿のグラフ化では評価指標を凡例とするが，評価指標ごとに値の範囲が大きく異なる場合が存在する．例えば，図 8 では，“F-measure” は値の範囲が 0〜0.6 であるが，“time[min]” はその範囲が 0.1〜250 である．そのため凡例ごとにグラフのスケールを考慮する必要がある．

本稿では，スケールの違いに対応するために各凡例の最大値が属す範囲を 10 のべき乗で分けてスケールを作成した．すなわち，図 8 では，“F-measure” の最大値は 10^{-1}〜10^{0} の範囲であるのに対し，“Time[min]” の最大値である “240.20” は 10^{2}〜10^{3} の範囲に属するためそれに従ってそれぞれのスケールを定める．

5. 表の集約

NTCIR(注7) の論文集では 1 つのタスクに対して複数の論文がある．その実験結果は同一なデータセット，評価指標であることが多く，論文間で結果を比較することができる．しかし，複数の論文中に示された結果を集約するには労力が伴う．そこで，本稿では各論文中に示された実験結果の表を集約し，1 つのグラフにまとめる方法を提案する．

5.1 表の集約手法

本稿では，各論文の表を列ごとに分割することで実験結果を評価指標ごとにまとめる．しかし，論文中の表において各列に評価指標ごとの結果がまとめられている，つまりヘッダ行に評価指標が書かれているとは限らない．そこでヘッダ列に評価指標が書かれている場合，表を転置する．

また，表中の評価指標の表記は筆者によって異なる場合があるので，その表記を読み替えるために辞書を作成した．辞書を図 13 に示す．各行のコロンの左が読み替え先であり，それより右にある語はそれに読み替える．集約の際にこの辞書を使用し，評価指標の置換を行う．

```
#of ques:#of Item,Question,# of Items
#of correct:# of Correct,Correct,# of Correct Answers
#of incorrect:#of Incorrect,# of Incorrect,incorrect
#of N/A:Unanswered
Correct rate:Accuracy,Accuracy Rate,Correct Answer Rate,Accurate Rate
Total score:Score
F-measure(max):F-measure
F-measure(spec.):F-measure (%)
```

図 13: 表記読み替えのための辞書

Table 1:Scores for official metrics

Transcrip type	Segmetation type	uMAP	pwMAP	fMAP
BASELINE		0.0670	0.0520	0.0536
manual	tt	0.0859	0.0429	0.0500
manual	C99	0.0713	0.0209	0.0168
ASR	tt	0.0490	0.0329	0.0308
ASR	C99	0.0469	0.0166	0.0123
ASR_new	tt	0.0312	0.0141	0.0174
ASR_new	C99	0.0316	0.0138	0.0120

Tbale 4: Passage retrieval result(RYSDT)

uMAP	0.0751
pwMAP	0.0725
fMAP	0.0650

図 14: 集約する表 (Table1 [17]，Table4 [18])

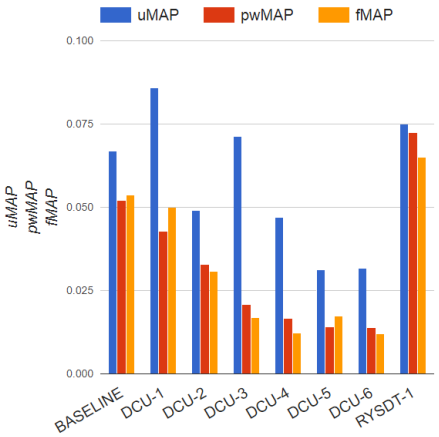

図 15: 2 つの表を集約したグラフ

5.2 集約した表のグラフ化

図 14，図 15 にそれぞれ集約した表とそのグラフ化結果を示す．図 15 のラベルは，NTCIR9 の Spoken Doc タスクの Overview の論文 [19] におけるオーガナイザの表記に合わせている．DCU が図 14 の左側の表であり，3 行目から順に図 15 のラベル “DCU-1” から “DCU-6” に対応している．RYSDT が右側の表を表しており，この表は評価指標である “uMAP”，“pwMAP”，“fMAP” がヘッダ列に書かれているため，転置の上で集約が行われ，グラフでは “RYSDT-1” のラベルにそのデータが示されている．

6. 評価実験

4. 節では，解析した表の構造情報に基づくグラフの自動生成について述べた．5. 節では，NTCIR において同一タスクに提出された複数の論文中の表の集約について述べた．ここでは，グラフの自動生成と表の集約について行った評価実験について

(注7)：http://research.nii.ac.jp/ntcir/index-en.html

表 1: グラフの自動生成結果

グラフ化対象の表の数	グラフ化に成功した表の数	可視化率
102(27)	97(18)	0.95(0.67)

表 2: 表の集約実験結果

overview の表のセルの数	集約した表のセルの数	再現率	適合率
299	340	0.92	0.81

述べる．それぞれに評価指標を定め，グラフの自動生成実験では平井らの手法と比較する．また，表の集約については NTCIR のタスクオーガナイザの執筆した Overview の論文中の表と比較することにより評価する．

まず，6.1 部でグラフの自動生成実験について，つづいて 6.2 部で表の集約について実験結果を示す．

6.1 グラフの自動生成実験

ここでは，4. 節で提案した方法でどの程度の表がグラフ化できるかを実験により示す．

NTCIR9 論文集 (注5) の Spoken Doc タスクの論文から 29 件，NTCIR12 論文集 (注8) の QA Lab-2 タスクの論文から 32 件，SpokenQuery & Doc-2 タスクの論文から 17 件，そして Journal of Machine Learning Research(注9) の Vol.18 より 57 件の計 135 件の実験情報に関する表を実験で使用する．そのうちグラフ化対象となったのは 102 件であった．グラフ化の指標として以下のように可視化率 [10] を用いる．ここでの可視化率とはグラフ化のことである．

$$\text{可視化率} = \frac{\text{グラフ化に成功した表の数}}{\text{グラフ化対象の表の数}} \quad (3)$$

グラフの生成に成功しているかどうかの判断は著者が行い，グラフ上の値と表の値の対応，凡例，およびラベルのテキスト，グラフのタイトルが適切に HTML 化されれば成功とした．グラフ化の結果を表 1 に示す．表中の括弧内の数字は先行研究である平井らの手法 [10] によるものである．

表 1 を見て分かるように平井らの手法と比較して，可視化率が 28 ポイント向上した．

また，グラフ生成に失敗した表にはセルの分割に失敗したものや，文字列の位置が問題となるものなどがあった．詳しくは 7. 節で考察する．

6.2 表の集約実験

5. 節で述べた表の集約について実験を行った．

表の集約の対象としたのは，NTCIR の同一タスクの複数の論文中の表である．具体的には，NITCIR9 論文集の Spoken Doc タスク，NTCIR12 の QA-Lab タスク，Spoken Query & Doc-2 タスクに提出された論文について表を集約し，タスクオーガナイザが執筆した Overview の論文に示された実験結果の表と比較した．

NTCIR9 論文集の Spoken Doc タスクの Overview の論文には 4 件，NTCIR12 の QA-Lab タスクのそれには 6 件，Spoken Query & Doc-2 タスクのそれには 4 件の表が存在した．表だけでなく，グラフとしてもまとめられていた結果も存在したが，本研究では比較に表のみを用いた．

評価指標として，以下のように再現率と適合率を定めた．

$$\text{再現率} = \frac{\text{正しく集約できたセルの数}}{\text{Overview 中の表のセルの数}} \quad (4)$$

$$\text{適合率} = \frac{\text{正しく集約できたセルの数}}{\text{集約した表のセルの数}} \quad (5)$$

集約実験の結果を表 2 に示す．

表 2 に示した Overview 中の表のセルの数は，集約する表にその値が存在したセルのみの数である．集約した表のセルの数は Overview 中の表のセルの数である 299 個よりも多いが，これは集約の際に Overview 中の表に含まれないセルも一緒に集約しているからである．

7. 考　　察

7.1 グラフ化できなかった表

6.1 節の表の自動生成実験においてグラフの生成に失敗した表について，その原因を考察する．失敗の理由は次の 3 種類に分類される．

- セルの生成の失敗
- 表の行と列の予測の失敗
- セル中の文字列の重心による罫線の修正の失敗

まず，セルの生成の失敗は，表の異なるセルに書かれた文字列間の間隔が小さく，異なるセルが同じセルと判断されるために発生した．反対に 1 つのセルであるのに文字列間の空白の大きさにより別のセルに分割され，適切にグラフ化できないものもあった．この解決には，セル中の文字列に対するトークンの結合や分割の規則をさらに検討することが必要である．

次に表の行と列の予測の失敗は，予測後の表にすべてのセルが埋まった行や列が存在しないことにより引き起こされた．本稿では，最初に罫線をセルごとに予測し列，行の順に文字列がどのセルに存在するかを決定する．そのため，すべてのセルが埋まっている行と列が存在しない場合，行数や列数が本来の数よりも小さくなる．表の行の予測に失敗する例を図 16 に示す．

この表では，垂直方向の罫線の予測の際に “Values” と “corpus1” は別の列と判定され，“Values” の列の 2 行目には “max” が位置していると予測される．さらに “corpus1” は 1 行目と判定される．そのため，“Values” と “corpus1” は共に 1 行目と判定され，行数が 1 つ減る．結果として，適切に罫線を予測することができない．

最後に，3.2.3 項で述べた罫線の修正の失敗である．その例となる表の一部を図 17 に示す．この図では，“Method1” や “Method2” は複数行からなるセルであるが，どちらも垂直方向の範囲にある “1” から “Baseline” の文字列の重心に位置するわけではなく，“2” のセルと同じ高さに位置している．これは，例えば “Method1” は 2〜5 行目からなるセルであるということを示すために，2〜5 行目を結合し，中央揃えにするのではなく，2〜5 行目の中心付近と思われる 2 行目に “Method1” を

(注5) : http://research.nii.ac.jp/ntcir/ntcir-9/index-ja.html
(注8) : http://research.nii.ac.jp/ntcir/ntcir-12/index-ja.html
(注9) : http://www.jmlr.org/

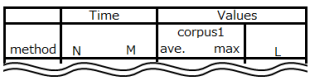

図 16: 行の予測に失敗する表の例

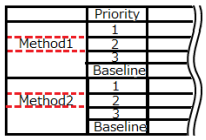

図 17: 表の罫線の修正に失敗する表の例

corpus1				
Method	Score	Pre.	Recall	F-mes.
Method1	26	0.6	0.70	0.65
Method2	35	0.55	0.66	0.6
corpus2				
Method	Score	Pre.	Recall	F-mes.
Method1	12	0.52	0.72	0.6
Method2	34	0.49	0.68	0.57

図 18: 複数の表がまとめられている表

書き，2〜5 行目の間の罫線を表示しないことによって 2〜5 行目からなるセルであることを表したためであると考えられる．

7.2 グラフ化しなかった表

4.1 節で挙げたグラフ化対象としない表のうち複数の表を含む表のグラフ化について考察する．

複数の表を含む表の例を図 18 に示す．この表からグラフを生成することは可能と考えられる．しかし，表とは列の名前を表すヘッダ行から始まり，行がその下に続くものである．一方，図 18 の表は “corpus1” と “corpus2” をそれぞれの表のキャプションとする 2 つの表へと分割することができる．今後，このような表の分割方法についても検討したい．

8. おわりに

本稿では，学術論文の表からグラフを自動生成するために，表の構造解析とグラフの生成手法を提案した．論文 PDF を XML ファイルに変換し，XML 中の文字列トークンの座標と重心を用いることによりグラフの構造を予測した．予測した表の構造を利用して Google Chart API を用い，グラフを生成し Google Chrome で表示した．実験では 102 件の表のグラフ生成を試み，その 95 %のグラフ化に成功した．

今後の課題として，セルの結合，分割を工夫してより複雑な表の構造解析を可能にすることが挙げられる．また，グラフの色や種類を工夫することによって，よりグラフを視覚的に理解しやすくするための方法も検討したい．表の集約においては，評価指標が同じであれば，1 つの表に複数のタスクの実験結果がまとめられている表も存在する　．その場合でも，実験ごとの結果をまとめられるように，表を実験ごとに分割，集約する方法も検討したい．

謝　　辞

本研究の一部は，国立情報学研究所公募型共同研究の援助による．ここに記して深謝する．

文　　献

[1] G. Kamola, M. Spytkowski, M. Paradowski, U. Markowska-Kaczmar : Image-based logical document structure recognition, Pattern Analysis & Applications, Vol. 18, Issue 3, pp. 651-665, 2015.

[2] Y. Wanga, I. T. Phillipsb, R. M. Haralickc : Table structure understanding and its performance evaluation, Pattern Recognition, Vol. 37, Issue 7, pp. 1479-1497, 2004.

[3] F. F. Babatunde, B. A. Ojokoh, S. A. Oluwadare : Automatic table recognition and extraction from heterogeneous documents, Journal of Computer and Communications, Vol. 13, no. 12, pp. 100-110, 2015.

[4] T. G. Kieninger : Table structure recognition based on robust block segmentation, Proceedings of the SPIE, Vol. 3305, pp. 22-32, 1998.

[5] T. G. Kieninger: Applying the T-Recs table recognition system to the business letter domain, Proceedings of the 6th International Conference on Document Analysis and Recognition, pp. 518-522, 2001.

[6] B. Yildiz, K. Kaiser, S. Miksch : pdf2table: A method to extract table information from PDF files, Proceedings of the 2nd Indian International Conference on Artificial Intelligence, pp. 1773-1785, 2005.

[7] A. Shigarov, A. Mikhailov, A. Altaev : Configurable table structure recognition in untagged, Proceedings of the 2016 ACM Symposium on Document Engineering, pp. 119-122, 2016.

[8] 村田真樹，Stijn De Saeger，橋本力，風間淳一，山田一郎，黒田航，馬青，相澤彰子，鳥澤健太郎：論文データからの重要情報の抽出と可視化，第 23 回人工知能学会全国大会，3F2-NFC3-9，2009.

[9] 平井久貴，新妻弘崇，太田学，高須淳宏：学術論文からの実験情報抽出とその可視化，情報処理学会研究報告，Vol. 2015-DBS-162，No. 21，pp. 1-7，2015.

[10] 平井久貴： 学術論文からの実験情報抽出とその可視化に関する研究，岡山大学大学院自然科学研究科電子情報システム工学専攻修士論文，2016.

[11] J. Lafferty，A. McCallum，F. Pereira：Conditional Random Fields:Probabilistic Models for Segmenting and Labeling Sequence Data，Proceedings of 18th International Conference on Machine Learning，pp. 282-289，2001.

[12] Y. Yamashita，T. Matsunaga，K. Cho： YLAB@RU at spoken term detection task in NTCIR-9，Proceedings of NTCIR-9 Workshop Meeting，pp. 287-290，2011.

[13] D. N. Racca，G. J. F. Jones： DCU at the NTCIR-12 SpokenQuery&Doc-2 Task，Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies，pp. 180-185，2016.

[14] N. Sawada，H. Nishizaki： evaluation of DNN-based phoneme estimation approach on the NTCIR-12 Spoken Query&Doc-2 SQ-STD subtask，Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies，pp. 211-216，2016.

[15] K. Katsurada，K. Katsuura，Y. Iribe，T. Nitta： Utilization of suffix array for quick STD and its evaluation on the NTCIR-9 SpokenDoc Task，Proceedings of NTCIR-9 Workshop Meeting，pp. 271-274，2011.

[16] M. Kobayashi，H. Miyashita，A. Ishii，C. Hoshino： NUL system at QA Lab-2 task，Proceedings of the 12th NTCIR Conference on Evaluation of Information Access Technologies，pp. 413-420，2016.

[17] M. Eskevich，G. J. F. Jones： DCU at the NTCIR-9 spokenDoc passage retrieval task，Proceedings of NTCIR-9 Workshop Meeting，pp. 257-260，2011.

[18] S. Tsuge，H. Ohashi，N. Kitaoka，K. Takeda，K. Kita： Spoken document retrieval method combining query expansion with continuous syllable recognition for NTCIR-SpokenDoc，Proceedings of NTCIR-9 Workshop Meeting，pp. 249-256，2011.

[19] T. Akiba，H. Nishizaki，K. Aikawa，T. Kawahara，T. Matsui： Overview of the IR for Spoken Documents Task in NTCIR-9 Workshop，Proceedings of NTCIR-9 Workshop Meeting，pp. 223-235，2011.