# TabNet: 表の意味構造を理解するハイブリッド型深層ニューラルネットワーク

西田　京介†　　貞光　九月†　　東中竜一郎†　　松尾　義博†

† 日本電信電話株式会社 NTT メディアインテリジェンス研究所　〒 239–0847 神奈川県横須賀市光の丘 1-1
E-mail: †{nishida.kyosuke,sadamitsu.kugatsu,higashinaka.ryuichiro,matsuo.yoshihiro}@lab.ntt.co.jp

**あらまし**　本研究では，表の種類を分類するための深層ニューラルネットワークアーキテクチャTabNet を提案する．表の種類は，Web 上に大量に存在する表から知識を獲得するために必須な情報であり，正確な表分類を行うためには表の意味構造を理解することが重要である．我々は「系列の行列」という表のデータ構造に着目してハイブリッド型のアーキテクチャを設計した．まず，再帰型ニューラルネットワークが各セルに記載されたテキスト（単語および HTML タグの系列データ）をエンコードして表を 3 次テンソルに変換し，次に，畳み込み型ニューラルネットワークがセルの集合から形成される意味特徴（例えば，属性が記述された行の存在）を抽出する．様々な構造とトピックを含む大規模な Web 表のデータセットを構築して評価実験を行い，TabNet が表分類に特化した従来手法と他の深層学習アーキテクチャに比べて統計的有意に高い分類精度を実現することを確認した．

**キーワード**　Web，表，表種類分類，深層学習，RNN，CNN

## 1. はじめに

Web 上には膨大な量の表が存在する．Cafarella らによる 2008 年の報告では，Web クロールデータに含まれる数十億ページには 141 億個の表が含まれ，関係知識を抽出可能な表は 1.54 億個にも上った (注1) [4]．近年のビッグデータ研究の進展に伴って，表からの情報抽出に関する研究が改めて注目を集めており，大規模データセットである WDC Web Table Corpus 2015 のリリース [25] や，表の検索 [38]，表セルの検索 [37]，表の拡張 [26]，表に基づく質問応答 [49]，表知識獲得 [11] など様々な研究が行われている．

このような表データに基づく研究開発において最も重要かつ根底をなす技術が**表種類分類技術**である．Crestan らは (主語，述語，目的語) (注2) のセマンティックトリプルで表現される知識が，表の内外でどのように記載されているかに基いて，表の種類のタクソノミおよび機械学習に基づく分類手法を提案した [8], [9]．この定義に基づいた表種類分類に関する研究では，主に機械学習アルゴリズムで利用する特徴量の設計について取り組まれてきた．Eberius らによる state-of-the-art 手法は，表全体に関する特徴と，行・列に関する局所的な特徴を述べ 127 個利用している [12]．しかし，表はセル間の意味的関係に基いて定義されているにも関わらず，従来手法は，セル毎の文字数の分散や，特定の文字・記号やタグが含まれるセルの割合など，表層的な特徴を用いているにすぎない．たとえば，エンティティを列挙する列や，属性を記載する行などは，表の意味構造を表す重要な構成要素であるが，このような様々な意味・形状・大きさを持つセル集合を捉える特徴を手動で設計することは難し

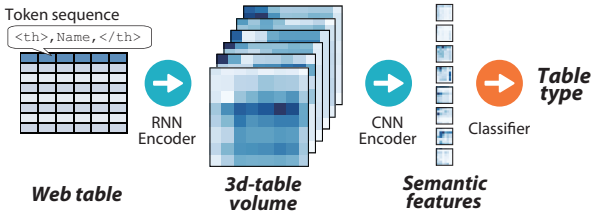


図 1　提案アーキテクチャTabNet の概要．最初に RNN が各セルのテキスト（トークン系列）を符号化し，次に CNN が表種類分類に有用な意味特徴を抽出する．

い．このため，従来手法では表の訓練データ数を増やしても分類精度が頭打ちとなり，大きな精度向上は実現できなかった．

我々は表の意味構造を理解するために，表は**系列の行列**というデータ構造を持つこと，すなわち，単語や HTML タグの系列であるテキストが記載されたセルが行列状に配置されているデータであることに着目した．近年では深層ニューラルネットワーク（Deep Neural Network; DNN）による系列および行列データからの意味表現抽出が大きな成功を収めており，再帰型ニューラルネットワーク（Recurrent Neural Network; RNN）は機械翻訳 [2], [45] を含む系列データのモデリングにおいて，畳み込み型ニューラルネットワーク（Convolutional Neural Network; CNN）は画像（ピクセルの行列データ）のクラス分類 [16] など様々なコンピュータビジョンタスクにおいて，それぞれ従来の機械学習手法を大きく凌駕する精度を実現している．

**本研究の貢献：** 本研究は，DNN の表種類分類への適用について世界で初めて検討し，以下の貢献を果たした．

- RNN と CNN の結合により表の意味構造を理解可能な新アーキテクチャTabNet を提案した（図 1）．
- TabNet は，表分類に特化した state-of-the-art 手法 [12]

(注1)：大部分の表（<table>タグ）は画面レイアウト目的で利用される．
(注2)：(エンティティ，属性，値) のモデルと可換である．

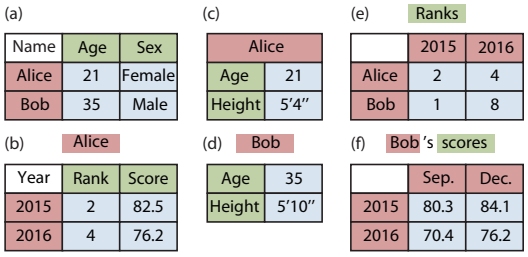


図 2　知識表の例．(a,b) 関係表，(c,d) エンティティ表，(e,f) 行列表．赤，緑，および青色のセルは，それぞれエンティティ，属性，値が記されているセルを示す．

と，他の DNN アーキテクチャ [48] に対して統計的有意かつ大きく優れた分類性能を実現した．

本論文の構成を以下に示す．まず，2 章にて本研究で取り組む問題の定義を示し，3 章にて提案する深層ニューラルネットワークアーキテクチャTabNet を説明する．次に，4 章にて評価実験の結果を示し，5 章にて本研究の貢献および新規性について議論した後，6 章に結論を示す．なお，本研究のさらなる議論については [32] も併せて参照されたい．

## 2. 問 題 定 義

本章では，はじめに本研究で分類の対象とする表について定義する．次に，本研究で深層ニューラルネットワークにより理解を目指す表の意味構造について説明する．

### 2.1 表 の 種 類

本研究で扱う表について定義を行う．

**定義 1.** **表**は，HTML の`<table>`タグを用いて記載された $N$ 行・$M$ 列の順序付きセル集合である．

**定義 2.** **セル** $c_{ij}$ は，表における $i$ 番目の行と $j$ 番目の列の交点である．ここで，$1 \leqq i \leqq N$ かつ $1 \leqq j \leqq M$ を満たす．

**定義 3.** **知識表**は，周辺のテキストと併せて（エンティティ，属性，値）のセマンティックトリプルを含む表である．

知識表の種類は，表内および表周辺に含まれる知識に基いて分類される [9]．WDC Web Table Corpus 2015 では，知識表を関係表，エンティティ表，行列表の 3 種類に分類しており [25]，関係表およびエンティティ表は垂直方向と水平方向の 2 種類がある．本研究でも同じタクソノミを利用して表種類の分類を行う．

#### 2.1.1 知　識　表

以下に，関係表，エンティティ表，行列表のそれぞれについて説明する．図 2 に例を示す．

**定義 4.** **関係表**は，エンティティあるいはエンティティの主要な観点のリストについて，それぞれ 1 つ以上の属性の値を挙げる．

関係表には図 2(a) のように完全なトリプルを含むものと，図 2(b) のようにエンティティを含まないものがある．後者は，表の外に出現するエンティティの主要な観点をリスト化したものであり，エンティティと観点の組み合わせが具体化（reification）されたエンティティを形成する．図 2(a,b) は垂直方向の関係表であり，水平方向の関係表ではエンティティが行方向にリスト化される．

**定義 5.** **エンティティ表**は，単一のエンティティについて，1 つ以上の属性と値のペアを記載する．

エンティティ表もまた，図 2(c) のように完全なトリプルを含むものと，図 2(d) のようにエンティティを含まないものがある．エンティティ表は関係表に比べて，単一のエンティティについての知識を記載する点が異なる．図 2(c,d) は水平方向のエンティティ表であり，垂直方向のエンティエィ表は 1 列の中で属性と値のペアを記載する．

**定義 6.** **行列表**は，エンティティの各組み合わせに対応する値がすべて同じ属性を持つ．

行列表は図 2(e,f) に示すように完全なトリプルを含まず，属性は一般的に表の外に記載される．属性の値は，行，列，および表の外に記載された 2 つ以上のエンティティの組み合わせに対応する．

#### 2.1.2 その他の表

Crestan らは他に 3 種類の知識表および 2 種類のレイアウト表を定義している [9]．**列挙表**は，同一の関係（例えば，”is-a” 関係）を持つ値を列挙し，**カレンダー表** は日付に関する情報を整理した表である．**フォーム表** はエンティティ表に類似しているもので，属性の値に対応するセルが（ユーザが記入あるいは選択するために）空欄である．

また，知識表以外のレイアウトを目的とした表は 2 つに大別される．**ナビゲーション表**は他ページへのハイパーリンクを整理するために利用され，**フォーマット表**は視覚的に情報を整理する目的で利用される．Web における表の大部分はフォーマット表である．

本研究では，これらの知識表およびレイアウト表をすべて「その他」の表として扱う．以上の定義に基づき，本研究では表種類分類問題を以下のように定義した．

**問題 1.** 本研究で取り組む**表種類分類問題**では，与えられた表を，垂直方向の関係表（Vertical Relational; VR），水平方向の関係表（Horizontal Relational; HR），垂直方向のエンティティ表（Vertical Entity; VE），水平方向のエンティティ表（Horizontal Entity; HE），行列表（Matrix; M），その他の表（Other; O）の 6 種類に分類する．

### 2.2 表の意味構造

本節では，知識表の意味構造を形成する主要な構成要素について説明する．併せて，従来研究 [9], [12] が用いる特徴量のうち，意味構造に関連するものを示す．

#### 2.2.1 エンティティ行および列

垂直（水平）方向の関係表は，1 つ以上のエンティティをリストにした列（行）を持つ．行列表はそれらの両方を持ち，エンティティ表はどちらも持たないため，エンティティ行・列の

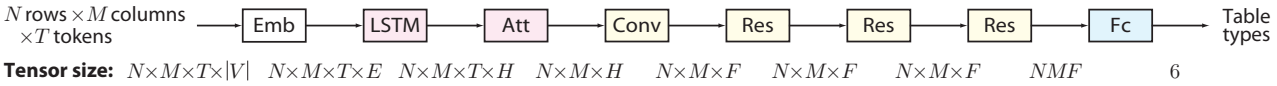


図 3　TabNet の全体アーキテクチャ．埋め込み層（Embedding）層が各トークンの one-hot ベクトル（$|V|$ 次元）を $E$ 次元の連続値ベクトルに変換する．RNN が，LSTM とアテンション（Attention）機構を利用して，各セルを $H$ 次元のベクトルに変換する．CNN が $F$ 個のフィルタを持つ畳み込み（Convlutional）層と，Residual units を用いた畳み込みブロックの積み重ねにより，表の意味構造を捉えた特徴を抽出する．全結合（Fully-Connected）層とソフトマックス関数により，6 クラスの予測確率を推定する．

存在は重要な特徴となる．従来手法では，各行・列の文字列の種類数に着目することで，エンティティ（すなわち，関係データベースにおける主キー）行・列の発見を試みる．

**2.2.2**　属性行および列

垂直（水平）方向の関係表およびエンティティ表は，1 つ以上の属性を記載した行（列）を持つ．行列表は属性が記載されたセルを表内に持たない．従来手法では，各列においてコロン記号がセル内の文字列に含まれる割合を，水平方向のエンティティ表に出現しやすい特徴として利用している．

**2.2.3**　同一属性を持つ値セルのブロック

ここで，同一属性を持つ値セルのグループを sibling ブロックと呼ぶ．垂直（水平）方向の関係表は，複数個の $n\times1$ $(1\times m)$ サイズの sibling ブロックを各列（行）に沿って持つ．行列表は 1 つの大きな sibling ブロックを形成する．エンティティ表は sibling ブロックをほとんど持たない．従来研究では，セルの文字列長の分散や，セル文字列の種類（数字，アルファベットなど）の一致を sibling ブロックの検出に利用している．

## 3.　ネットワークアーキテクチャ

本章では，RNN と CNN の結合により表の意味構造を理解可能なネットワーク TabNet について説明する．本章ではアーキテクチャの一般的な議論を行い，具体的なパラメータ値や層数，学習方法などについては 4.4 節に示す．

### 3.1　概　　要

図 3 に TabNet のアーキテクチャの概要を示す．TabNet への入力は，各セルが $T$ 個のトークン（語彙サイズ $|V|$）を持つ $N$ 行 $M$ 列の固定サイズの表とし，$N\times M\times T$ よりも大きい（小さい）表データはそれぞれクロッピング（パディング）を行う．入力が与えられると，埋め込み層によりセル内の各トークンを $E$ 次元のベクトルに変換する．次に，Long Short-Term Memory（LSTM）とアテンション機構を備えた RNN により，各セル（トークン系列）を $H$ 次元のベクトルにエンコードして，各セルの意味表現を獲得する．結果として RNN の出力は $N\times M\times H$ の 3 次テンソルとなり，このテンソルから CNN を用いて表の意味構造を表す特徴量の抽出を行う．CNN では，$F$ 個のフィルタを持つ畳み込み層の積み重ねにショートカット接続を加えた residual units を利用する．最後に，CNN の出力を 1 次元ベクトルに変換して分類層に与え，全結合層およびソフトマックス関数により 6 クラスの表種類の予測確率を出力する．

### 3.2　埋め込み層

#### 3.2.1　トークナイズ

TabNet では，単語，HTML タグ，行番号，列番号をトークンとして扱う．HTML タグの属性は，`rowspan` と `colspan` 以外は無視する．結合されたセルについては，図 4 に示すようにセルの行・列番号およびタグ名にて結合元のセルと識別を可能にする．`<thead>`, `<tbody>`, `<tr>`, `<colgroup>`, `<col>`, `<caption>`タグの内容については本研究では利用していない．全ての単語とタグについては小文字への統一および NFKC 正規化を行った．

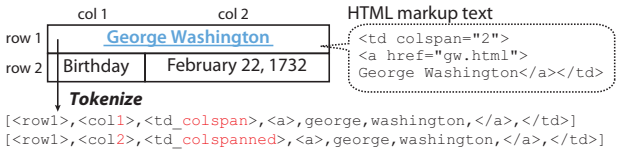


図 4　HTML マークアップテキストのトークナイズ例．

#### 3.2.2　トークン埋め込み

各セルは固定長のトークン系列（クロッピングあるいはパディングを実施したもの），すなわち，$T$ 個の one-hot ベクトル $(x_1, x_2, \ldots, x_T)$ を持つ．ここで，語彙集合 $V$ における $v$ 番目のトークンに対応する one-hot ベクトルは，$v$ 番目の要素のみ 1，他の要素は 0 で表される．埋め込み層は，各 one-hot ベクトルを，重み行列 $W_e \in \mathbb{R}^{E\times|V|}$ を用いて $E$ 次元の連続値ベクトルに射影する：

$$e_t = W_e x_t, \tag{1}$$

ここで，$|V|$ は語彙集合のサイズを表す．

### 3.3　再帰型ニューラルネットワーク（RNN）

埋め込み層の適用後，各セルは密な連続値ベクトルの系列 $(e_1, e_2, \ldots, e_T)$ を持つ．TabNet はこの系列を逆順に処理する LSTM とアテンション機構を，各セルの意味表現を獲得するために利用する．

#### 3.3.1　Long Short-Term Memory（LSTM）

LSTM は系列データの長期的な依存関係を学習することのできる RNN の特別な一種であり [13], [20]，隠れ状態 $h_t$ の系列を以下のように定義する：

$$\begin{pmatrix} f_t \\ i_t \\ o_t \\ g_t \end{pmatrix} = W_h h_{t-1} + W_x e_t + b, \tag{2}$$

$$c_t = \sigma(f_t) \odot c_{t-1} + \sigma(i_t) \odot \tanh(g_t), \tag{3}$$

$$h_t = \sigma(o_t) \odot \tanh(c_t), \tag{4}$$

ここで，$W_h \in \mathbb{R}^{4H\times H}$，$W_x \in \mathbb{R}^{4H\times E}$，$b \in \mathbb{R}^{4H}$ がモデルパラ

メータである．$\sigma$ はシグモイド関数，$\odot$ はアダマール積を表す．

**3.3.2** アテンション機構

TabNet では各セルのトークン系列のうち重要なものがセルの意味表現の形成に強く寄与するように，[48] と同じアテンション機構を用いる．

$$u_t = \tanh(W_a h_t + b_a), \tag{5}$$

$$\alpha_t = \frac{\exp(u_t^\intercal u_a)}{\sum_t \exp(u_t^\intercal u_a)}, \tag{6}$$

$$s = \sum_t \alpha_t h_t, \tag{7}$$

ここで，$W_a \in \mathbb{R}^{C\times H}$，$b_a \in \mathbb{R}^C$，$u_a \in \mathbb{R}^C$ はモデルパラメータである．$C$ はコンテキストベクトル $u_a$ の長さを表す．

### 3.4 Convolutional Neural Network

RNN の適用後，入力された表は $N\times M\times H$ テンソルに変換される．これは画像の 高さ×幅×深さ と同じデータ構造であるため，一般的な方法で CNN が適用できる．CNN の畳み込み層は，高さ・幅方向に小さな受容野（ただし，深さ方向は全て考慮する）を持つ複数個のフィルタから構成され，各フィルタはフィルタのエントリと入力の内積を，入力の高さ・幅方向に渡って計算し，2 次元の活性化マップを生成する．

TabNet では最初に $F$ 個の $3\times 3$ フィルタをストライド幅 1 で適用する畳み込み層を適用し，その後に次項で説明する畳み込みブロックを複数個適用する．なお，プーリングについては実施しない．

**3.4.1** residual units を利用した畳み込みブロック

Deep residual networks (ResNets) は多数の residual units の積み重ね（100 層以上）により画像分類タスクにおいて高い精度を実現した [16], [18]．層間のショートカット接続を持つことが特徴で，次の一般形で表される：

$$x_{l+1} = f\left(h(x_l) + \mathcal{F}(x_l, \mathcal{W}_l)\right), \tag{8}$$

ここで，$x_l$ と $x_{l+1}$ は $l$ 番目のユニットの入出力であり，$\mathcal{W}_l$ は $l$ 番目のユニットに関する重みとなる．TabNet では，2 層の畳み込み層（$F$ 個の $3\times 3$ フィルタ，ストライド幅 1）を $\mathcal{F}$ として利用した．関数 $f$ は ReLU 関数 [31] とし，関数 $h$ は全てのユニットにおいて恒等写像 $h(x_l) = x_l$ とした．このショートカット接続により層数が増加した際の過学習のリスクが軽減される．

**3.4.2** Batch normalization

Batch normalization（BN）は，ミニバッチ内における各層の各入力特徴の分布を $\mathcal{N}(0,1)$ に正規化することでニューラルネットワークの学習を高速化する技術である [21]．TabNet ではオリジナルの ResNet [16] と同様に，各畳み込みフィルタの適用直後と ReLU 活性化関数の適用直前で BN を適用した．

### 3.5 分 類 層

CNN の出力は $N\times M\times F$ テンソルとなる．これを $NMF$ 次元のベクトルへ変換した後に，複数個の全結合層を適用して，最終的に 6 次元まで次元縮退を行って表種類の分類を行う．活性化関数には ReLU を利用し，中間層においては ReLU の直前に BN を適用した．最終層の出力を softmax 関数に与えることで各表種類の予測確率を得る．

表 1　利用したデータセットに含まれる表の個数．

| table type | train | test | total |
|---|---|---|---|
| Vertical Relational (VR) | 11,397 | 328 | 11,725 |
| Horizontal Relational (HR) | 255 | 38 | 293 |
| Vertical Entity (VE) | 390 | 75 | 465 |
| Horizontal Entity (HE) | 15,618 | 879 | 16,497 |
| Matrix (M) | 662 | 18 | 680 |
| Other (O) | 32,356 | 2,229 | 34,585 |
| total | 60,678 | 3,567 | 64,245 |

### 3.6 ネットワークの学習

TabNet は，誤差逆伝播法で求められた交差エントロピー誤差の勾配に基いて，確率的勾配降下（stochastic gradient descent; SGD）により end-to-end 学習が可能である．これは，一般的なニューラルネットワークのフレームワークにより実装できる．なお，埋め込み層の埋め込み行列 $W_e$ については，表の訓練データセットとは別に，Word2Vec [28], [29] あるいは GloVe [33] を用いて大規模なテキストコーパスから学習させることが有効である．

## 4. 評 価 実 験

### 4.1 データセット

評価実験では，April 2016 Common Crawl の一部より収集した 272,888 個の表から，表の出現数が上位 500 ドメインの Web サイト（最上位は wikipedia.org）に含まれ，2 行 2 列以上かつ内部に表を持たない 64,245 個の表に絞り込んだデータセットを構築して利用した．正解データについては，1 人の熟練者により表の種類を 2.1 節の定義に従って 6 種類に分類して付与した．

ここで，1 つの Web サイトは類似した表を多く含むため，データセットをランダムに分割した場合，完全に未知のデータとは言えない表がテストデータに含まれてしまう．そこで，本実験では上位 300 サイトに含まれる 60,678 個の表を訓練データとして，残りをテストデータとして分割した．表 1 に，各クラスのデータの個数を示す．

### 4.2 評 価 指 標

訓練およびテストデータの各クラスのデータ数は不均衡である．特に，垂直方向の関係表，水平方向のエンティティ表，その他の表が，他のクラスに比べて多い．そこで，先行研究 [12] と同様に重み付きマクロ平均 $F_1$ 値（各クラスについて $F_1$ 値を計算し，データ数によって重み付けした平均）を評価指標として利用した．

### 4.3 ベースライン

表種類分類に特化した従来手法，および，文書分類に用いられるアーキテクチャのうち表分類に利用可能なものをベースラインとして合計 5 つ利用した．

**4.3.1** 表種類分類に特化した従来手法

表種類分類に関する研究では，これまで主に表分類に有用な

特徴量の設計について取り組まれてきた．本実験では，代表的な研究により提案された特徴を用いて Random Forests で学習を行った．

- Cafarella08 [5]

Cafarella らが設計した表全体に対する 7 つの特徴．行数，列数，セル内文字列長の分散など．

- Crestan11 [9]

Crestan らが設計した 107 個の特徴．表全体に対する特徴に加えて，表の最初の 2 行・2 列および最終の行・列について，行・列単位の素性が含まれる．構造的特徴に加えて，`<th>`タグの割合など HTML レベルの特徴や，コロンが含まれるセルの割合などセル内文字列の特徴が利用される．

- Eberius15 [12]

Eberius らによる Crestan11 を拡張して得られた 127 個の特徴．Dresden Web Table Corpus の構築に利用された．

#### 4.3.2 ニューラルネットワーク

深層ニューラルネットワークが表種類分類に適用された研究はこれまでに存在しない．そこで，文書分類に用いる最新のニューラルネットワークアーキテクチャのうち，表種類分類に適した構造をベースラインとして利用した．

- Hierarchical Attention Network（HAN）[48]

HAN は階層構造を持つ文書（単語の系列である文，文の系列である文書）を分類するために設計されたアーキテクチャであり，アテンション機構を備えた 2 層の RNN で構成される．ここで，表も文書と同様に階層構造を持つ（トークンの系列であるセル，セルの系列である行，行の系列である表）ことから，我々は 3 層の HAN を構築してベースラインとした．なお，HAN が用いる RNN ブロックは，TabNet の RNN ブロックと同一の物を用いた．

- Bidirectional HAN

上記した HAN では表を行の系列として扱ったが，もう一つの階層構造として，列の系列として表を見ることができる．そこで，行方向と列方向の 2 つの HAN を構築し，これらの出力を連結して分類層に与えて表種類分類を行ったものを Bidirectional HAN として利用した．

### 4.4 モデル構成

本節では提案アーキテクチャである TabNet の具体的なモデル構成および学習方法について示す．

#### 4.4.1 事前学習時

トークン埋め込みのために事前学習を実施した．トークンのうち単語については，Word2Vec（skip-gram モデルおよび negative sampling）を学習して埋め込み行列 $W_e$ のパラメータとして利用した [28], [29]．Word2Vec の学習コーパスには Wikipedia 記事を用いた．なお，これらの記事中の表はテストデータには含まれない．事前学習に出現しない単語については未知語（`UNK` トークン）として扱った．他の HTML タグと行・列番号のトークンについては，埋め込み行列のパラメータをランダムに初期化した．埋め込み次元数 $E$ は 100 に設定した．

#### 4.4.2 学 習 時

予備実験により，表の左上部分を用いれば分類には十分であることを確認したため，全ての表は 8 行 8 列に固定した．大きな表については左上の角からクロッピングを行い，小さな表については，右下部分に空セルをパディングした．各セルは 50 個のトークン系列とし，同様に先頭からクロッピングあるいは末尾に `PAD` トークンをパディングした．この `PAD` トークンは RNN のエンコーディング時に影響を与えないように実装した．なお，トークン系列の固定長化は，セル中に含まれる非常に長いテキストに対する処理速度低下を防ぐ目的で導入した．

アーキテクチャの実装にはニューラルネットワークの汎用ライブラリである Chainer [41] を用いた．CNN のパラメータについては [16] と同様に初期化した．LSTM については，forget bias は [22] と同様に各要素を 1 で初期化，hidden-to-hidden 重みについては orthogonal 初期化 [35]，input-to-hidden 重みについては一様分布からのサンプリングを行う Xavier 初期化 [14] を実施した．パラメータ最適化は Momentum SGD をモーメント値 0.9，ミニバッチサイズ 50 で実施した．学習エポックの数は 5 回とし，学習率はそれぞれ 0.1，0.1，0.1，0.01，0.001 とスケジューリングした．その他， LSTM の隠れベクトル長 $H = 100$，アテンション機構で用いるコンテキストベクトル長 $C = 100$，CNN の畳み込みフィルタ数 $F = 32$ と設定した．residual units を用いた畳み込みブロックの個数は 3 とし，全結合層の数は 2（ニューロン数は 100，6）とした．結果として，TabNet の重みパラメータを持つ層数は 12 となった．これらのモデル構成・パラメータは，訓練データから選択した 60 サイトの表をバリデーションデータとして利用してチューニングした．

表 2 テストデータに対する重み付きマクロ平均 $F_1$ 値．ニューラルネットワークの結果は 5 回の試行の平均 ± 標準誤差を示す．

| method | weighted macro $F_1$ |
|---|---|
| Cafarella08 | 0.6926 |
| Crestan11 | 0.8114 |
| Eberius15 | 0.8165 |
| HAN | 0.8409 ± 0.0056 |
| Bidirectional HAN | 0.8562 ± 0.0045 |
| TabNet | **0.8842** ± 0.0070 |
| Ensemble of 5 HANs | 0.8471 |
| Ensemble of 5 Bidirectional HANs | 0.8652 |
| Ensemble of 5 TabNets | **0.9105** |

### 4.5 評 価 結 果

本節では，以下のリサーチクエスチョン（RQ）を明らかにするために行った評価実験の結果について示す．

**RQ 1. TabNet は表種類分類に特化した手法と他の DNN アーキテクチャよりも良い精度を実現できるか？**

表 2 は全訓練データで学習したモデルのテストデータに対する重み付きマクロ平均 $F_1$ 値を示す．提案手法 TabNet は 5 回の異なる初期化による試行で $F_1$ 値が平均 88.42% となり，単一モデルとしては最も良い分類精度を実現した．TabNet は，表種類分類に特化した state-of-the-art 手法である Eberius15 に比べ 6.77% の精度向上を示し，他の DNN アーキテクチャである Bidirectional HAN に比べて有意な精度向上が確認された

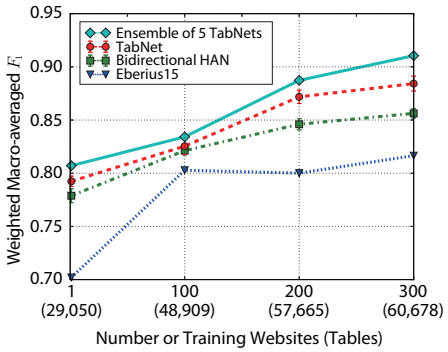


図 5　訓練 Web サイト数による分類精度への影響．TabNet と Bidirectional HAN のエラーバーは 5 試行の標準誤差を示す．

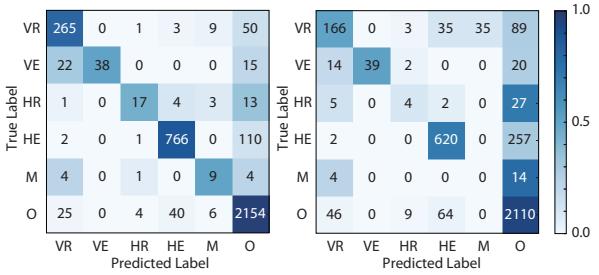


図 6　混同行列．セルの色については行ごとに正規化した値を示す．左：TabNet（5 モデルのアンサンブル），右：Eberius15．

($t$ 検定，$p < .05$)．また，表 2 に示す様に，TabNet の 5 モデルのアンサンブル（多数決，同数の場合はランダムに出力を選ぶ）について評価したところ，単一モデルに対して 2.63% の精度向上が確認された．

**RQ 2. DNN は表の訓練データ数が増えると共に分類精度を向上できるか？**

これまで主に取り組まれてきた手動設計した特徴量を Random Forests などの機械学習アルゴリズムにより学習する方法では，表の訓練データ数が増えても大きな精度向上に結びつくことは無かった．そこで，訓練 Web サイト数を 1，100，200，300（表数はそれぞれ 29,050，48,909，57,665，60,678 個）と変化させた場合の分類精度について評価した．図 5 に示す通り，Eberius15 については訓練データ数を増加しても分類精度の向上が頭打ちとなるが，DNN を使用する TabNet と Bidirectional HAN は訓練データ数の増加により分類精度が改善された．これらの結果は，手動設計した特徴量に比べ DNN が表の意味構造を本質的に捉えやすく，データ数が増えることにより様々な表の構造およびトピックを学習できることを示唆する．

**RQ 3. TabNet の分類エラーは妥当な結果となっているか？**

図 6 に TabNet と Eberius15 の混同行列を示す． Eberius15 はデータ数が少ない行列表（M）と水平方向の関係表（HR）の大部分を見落とし，さらに垂直方向の関係表（VR）と 水平方向のエンティティ表（HE）を間違うなど妥当ではないエラーが多いことが分かる．その一方で，TabNet については垂直方向の関係表（VR）とエンティティ表（VE）間のエラーなど比較的妥当なものが多かった．

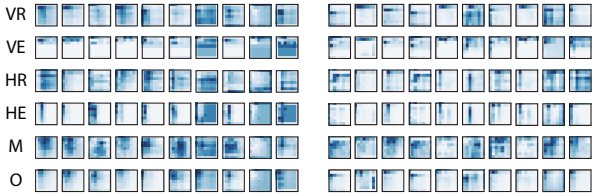


図 7　最初（左）と最終（右）の畳み込み層フィルタ（32 個中のそれぞれ 10 個）の活性化マップの可視化．表種類毎にテストデータに対する活性状況を平均化して作成した．縦方向に並んだ 6 つのマップが同じフィルタに対応する．

**RQ 4. TabNet は表の意味構造を捉える特徴を抽出できるか？**

図 7 は CNN の最初と最終の畳み込みフィルタの活性化マップの平均を可視化したものを示す．興味深いことに，属性行・列を持つ関係表とエンティティ表では属性行・列に沿った線状の活性が見られ，それらを持たない行列表では同様の活性が見られなかった．その一方で，行列表ではマップ上で広く分布した活性が見られた．これは，同一属性を持つ値セルのグループ（sibling ブロック）を捉えた特徴である．このような，サイズや形状が様々に異なるセルのブロックについては，HAN などの階層 RNN では捉えにくい特徴であり，TabNet が用いる CNN が特徴抽出に有効に働いていると考える．

## 5. 議　　論

### 5.1 表データを対象とした研究への本研究の貢献

本研究で取り組んだ表種類分類技術は，表データが持つ良質な知識を活用するために重要な技術である．例えば，Google の Knowledge Vault は垂直方向の関係表を知識ベースを獲得するための情報源の一つとして利用している [11]．Microsoft では水平方向のエンティティ表の情報を利用して検索エンジンにおいて直接質問に回答する機能を実現している [49]．その他にも，Wang らは垂直方向の関係表を基にエンティティのシードから concept（エンティティ集合）の拡張 [44]，Dalvi らは垂直方向の関係表からクラス・インスタンスペアの抽出 [10]，He らはクエリログと垂直方向の関係表から類似属性の獲得 [19] をそれぞれ行っている．我々の研究はこれらに対して良質な表のコーパスを準備する段階で大きな貢献が可能である．

また，我々の研究はこれまでの表へのアノテーション [27], [42] やエンティティリンキング [3], [30] とは異なり，既存の知識ベースに依存しない．これは，既存の知識ベースがカバーしない新規やロングテールな情報について知識獲得をする際に利点となる．

### 5.2 ハイブリッド型 DNN アーキテクチャの新規性

RNN と CNN を結合したアーキテクチャについては近年研究が進んでいる．代表的な研究としては，画像キャプション生成 [47] が挙げられるが，他にも画像セグメンテーション [6], [43]，音声認識 [34]，文書分類 [39], [46] など様々なタスクで用いられている．これらのアーキテクチャは，最初に CNN，その後に RNN を利用しており，我々のアーキテクチャとは適用順が異なる．TabNet は，最初に RNN がセルの意味情報をエンコード

し，次に CNN がセル間の意味関係を学習する点で新規性が高く，表の意味構造を捉える特徴を獲得できる．また，表以外にも，データ構造が系列の行列（系列のリスト含む）となるデータに対して適用することが可能なアーキテクチャである．

### 5.3 未利用の深層学習技術

本節では，予備検討の結果利用しなかった技術および最近報告された技術の導入について議論を行う．まず，パラメータ最適化については，RmsProp [15], [40] や Adam [24] などの適応学習率のアルゴリズムも用いて検証したが，学習率のスケジューリングを手動で設定した Momentum SGD が最も良い結果が得られた．次に，正規化については，dropout [36]，Batch Normalization [21]，Layer Normalization [1] について実験的に検討したが，Batch Normalization が最も良い結果が得られた．また，本研究では入力となる表を固定サイズとした. [17], [23] などで提案されたプーリング技術を用いることで可変長の入力サイズを処理することが可能だが，表では画像に比べてセル数が少ないためプーリングによる分類精度の悪化が大きく採用できなかった．最近，画像セグメンテーションなどで注目を集めている Dilated convolution [7], [50] についても表のサイズが小さいため，TabNet においてはプーリングと同様の問題があると考える．未実験の技術のうち特に有望なものとしては，機械翻訳で導入された多層 LSTM への Residual connections の導入 [45] があり，今後評価を行いたい．

## 6. おわりに

本研究では表の種類を分類するための深層ニューラルネットワークアーキテクチャTabNet を提案した．TabNet は表に含まれる（エンティティ，属性，値）の意味構造に基いて，6 種類に表を分類する．TabNet は表の構造を反映して設計されたアーキテクチャであり，RNN が各セルのトークン系列をエンコードし，CNN がセルの意味構造（属性を記載する行の存在など）を特徴として抽出することで，表種類の分類を行う．

本研究で我々は，Web クロールデータから 64,245 個の様々な構造・トピックを含む表のコーパスを構築して評価実験を行った．異なる初期化により 5 つの TabNet モデルのアンサンブルを構築したところ，3,567 個の未知の表データに対して，重み付きマクロ平均 $F_1$ 値において 91.05% の分類精度を達成した．この結果は，表種類分類に特化した state-of-the-art 手法 [12] と，他の深層ニューラルネットワークアーキテクチャ [48] に対してそれぞれ 9.40%，5.43% の精度向上を実現した．また，TabNet のエラー分析を行ったところ，他手法に比べてエラーの内容が妥当であることを確認した．そして，CNN の活性化マップの可視化により，TabNet が用いる CNN によって表種類に有用な表の意味構造が抽出されていることを確認できた．

今後は，TabNet で分類を行って構築した高品質な表コーパスを用いて，知識ベース構築や知識検索を改善することが目標である．また，提案アーキテクチャの応用可能性について，TabNet を他のタスクに適用して評価したい．

## 文　献


[1] L. J. Ba, R. Kiros, and G. E. Hinton. Layer normalization. eprint arXiv:1607.06450, 2016.

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015.

[3] C. S. Bhagavatula, T. Noraset, and D. Downey. TabEL: Entity linking in web tables. In *ISWC*, pages 425–441, 2015.

[4] M. J. Cafarella, A. Y. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Webtables: exploring the power of tables on the web. *PVLDB*, 1(1):538–549, 2008.

[5] M. J. Cafarella, A. Y. Halevy, Y. Zhang, D. Z. Wang, and E. Wu. Uncovering the relational web. In *WebDB*, 2008.

[6] L. Chen, J. T. Barron, G. Papandreou, K. Murphy, and A. L. Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. eprint arXiv:1511.03328, 2015.

[7] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[8] E. Crestan and P. Pantel. A fine-grained taxonomy of tables on the web. In *CIKM*, pages 1405–1408, 2010.

[9] E. Crestan and P. Pantel. Web-scale table census and classification. In *WSDM*, pages 545–554, 2011.

[10] B. B. Dalvi, W. W. Cohen, and J. Callan. Websets: extracting sets of entities from the web using unsupervised information extraction. In *WSDM*, pages 243–252, 2012.

[11] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610, 2014.

[12] J. Eberius, K. Braunschweig, M. Hentsch, M. Thiele, A. Ahmadov, and W. Lehner. Building the dresden web table corpus: A classification approach. In *BDC*, pages 41–50, 2015.

[13] F. A. Gers, J. Schmidhuber, and F. A. Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

[14] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, pages 249–256, 2010.

[15] A. Graves. Generating sequences with recurrent neural networks. eprint arXiv:1308.0850, 2013.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. eprint arXiv:0706.1234, 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(9):1904–1916, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. eprint arXiv:1603.05027, 2016.

[19] Y. He, K. Chakrabarti, T. Cheng, and T. Tylenda. Automatic discovery of attribute synonyms using query logs and table corpora. In *WWW*, pages 1429–1439, 2016.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[21] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015.

[22] R. Józefowicz, W. Zaremba, and I. Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, pages 2342–2350, 2015.

[23] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *ACL (1)*, pages 655–665, 2014.

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[25] O. Lehmberg, D. Ritze, R. Meusel, and C. Bizer. A large public corpus of web tables containing time and context metadata. In *WWW*, pages 75–76, 2016.

[26] O. Lehmberg, D. Ritze, P. Ristoski, R. Meusel, H. Paulheim, and C. Bizer. The mannheim search join engine. *J. Web Sem.*, 35:159–166, 2015.

[27] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *PVLDB*, 3(1):1338–1347, 2010.

[28] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. eprint arXiv:1301.3781, 2013.

[29] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[30] V. Mulwad, T. Finin, and A. Joshi. Semantic message passing for generating linked data from tables. In *ISWC*, pages 363–378, 2013.

[31] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814, 2010.

[32] K. Nishida, K. Sadamitsu, R. Higashinaka, and Y. Matsuo. Understanding the semantic structures of tables with a hybrid deep neural network architecture. In *AAAI*, pages 168–174, 2017.

[33] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[34] T. N. Sainath, O. Vinyals, A. W. Senior, and H. Sak. Convolutional, long short-term memory, fully connected deep neural networks. In *ICASSP*, pages 4580–4584, 2015.

[35] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. eprint arXiv:1312.6120, 2013.

[36] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[37] H. Sun, H. Ma, X. dong He, W. Yih, Y. Su, and X. Yan. Table cell search for question answering. In *WWW*, pages 771–782, 2016.

[38] N. T. Tam, N. Q. V. Hung, M. Weidlich, and K. Aberer. Result selection and summarization for web table search. In *ICDE*, pages 231–242, 2015.

[39] D. Tang, B. Qin, and T. Liu. Document modeling with gated recurrent neural network for sentiment classification. In *EMNLP*, pages 1422–1432, 2015.

[40] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[41] S. Tokui, K. Oono, S. Hido, and J. Clayton. Chainer: a next-generation open source framework for deep learning. In *LearningSys Workshop at NIPS*, 2015.

[42] P. Venetis, A. Y. Halevy, J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering semantics of tables on the web. *PVLDB*, 4(9):528–538, 2011.

[43] F. Visin, K. Kastner, A. C. Courville, Y. Bengio, M. Matteucci, and K. Cho. Reseg: A recurrent neural network for object segmentation. eprint arXiv:1511.07053, 2015.

[44] C. Wang, K. Chakrabarti, Y. He, K. Ganjam, Z. Chen, and P. A. Bernstein. Concept expansion using web tables. In *WWW*, pages 1198–1208, 2015.

[45] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. eprint arXiv:1609.08144, 2016.

[46] Y. Xiao and K. Cho. Efficient character-level document classification by combining convolution and recurrent layers. eprint arXiv:1602.00367, 2016.

[47] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[48] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy. Hierarchical attention networks for document classification. In *NAACL HLT*, pages 1480–1489, 2016.

[49] X. Yin, W. Tan, and C. Liu. FACTO: a fact lookup engine based on web tables. In *WWW*, pages 507–516, 2011.

[50] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. eprint arXiv:1511.07122, 2015.