分散表現と素性を利用した参考文献書誌情報抽出

浪越　大貴† 　太田　学† 　高須　淳宏†† 　安達　淳††

† 岡山大学大学院自然科学研究科　〒 700–8530 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: †{namikoshi, ohta}@de.cs.okayama-u.ac.jp, ††{takasu, adachi}@nii.ac.jp

あらまし　膨大な文書が格納されている電子図書館の運用には，書誌情報データベースの整備が必須である．特に学術論文の参考文献欄には著者名やタイトルなどの有用な書誌情報が集約されているため，参考文献文字列から書誌情報を自動抽出する研究が行われている．これまでの研究では，Conditional Random Field（CRF）等の機械学習による書誌情報抽出が提案されている．本稿では，この問題に単語の分散表現と素性を入力とするニューラルネットワークモデルを提案し，参考文献文字列からの書誌情報抽出精度を実験により確かめる．

キーワード　書誌情報抽出，ニューラルネットワーク，参考文献文字列，word2vec，素性

1. はじめに

多数の学術論文を蓄積する電子図書館のサービスを利用する際，検索や文書間リンク等の機能は必須であり，これらの機能を利用するには，著者名やタイトルといった書誌情報が必要となる．しかし，これらの書誌情報を人手でデータベースに入力するコストは膨大なため，その作業を可能な限り自動で行う文書解析技術が求められている．

そこで，本稿では，学術論文の参考文献文字列に着目する．学術論文の参考文献欄には，多くの関連文献が記述されており，その著者やタイトルといった書誌情報がある．さらに，参考文献文字列の解析は書誌エンティティの同定や曖昧性解消でも重要である．Pereira ら [1] は，Google Scholar と Microsoft Academic Search から収集した参考文献文字列を対象に，相関ルールを利用して会議名や論文誌名の曖昧性を解消した．Köpcke ら [2] は，DBLP，ACM Digital Library や Google Scholar から得たデータソースを対象に，書誌エンティティ同定の包括的で比較可能な評価手法を提案した．したがって，このような書誌情報抽出を正確かつ可能な限り自動で行う文書解析技術は有用である．これまでに，自然言語処理などの様々な分野で利用されている識別モデルの一つである Conditional Random Field（CRF）を利用して，参考文献文字列から書誌情報を自動抽出する研究が行われている [3, 4].

本稿では，参考文献文字列から書誌情報を抽出するため，word2vec を用いて得られる参考文献文字列の単語の分散表現と，その単語の特徴を示す素性を入力とするニューラルネットワーク（NN）モデルを提案する．また提案したモデルを用いて書誌情報の抽出精度を評価し，その結果について考察する．

本稿の構成は次の通りである．まず，2. 節で学術論文からの書誌情報抽出に関する研究を紹介し，3. 節で本研究で提案する自動書誌情報抽出法にのための NN モデルについて説明する．つづく 4. 節で提案手法の評価実験を示し，その結果について 5. 節で考察する．最後に 6. 節で本稿をまとめる．

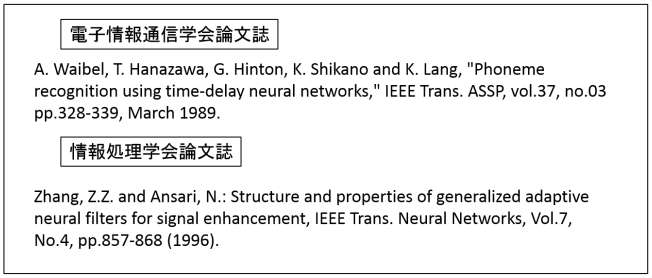

図 1　学術論文誌による参考文献文字列の書式の違い

2. 関連研究

2.1 書誌情報抽出に関する研究

多数の学術論文を格納する電子図書館において，書誌情報の管理は必須であるため，学術論文から書誌情報を自動抽出する研究が行われている．ルールにより参考文献文字列から書誌情報を抽出する場合，例えば図 1 のように著者名，タイトル，発行年などの書式が異なる論文誌ごとに，書誌情報抽出のためのルールを設定する必要がある．増大する学術論文誌をかかえる電子図書館では，このようなルールを定義し，管理していくことは今後ますます困難となることが予想される．

そのため，学習データを用意すれば利用可能な機械学習による書誌情報抽出が多く提案されている．例えば，CRF [5] を用いた書誌情報抽出に関する研究に，Peng ら [6]，Councill ら [7]，Do ら [8]，Cuong ら [9] の研究がある．Peng らは，学術論文のタイトルページと参考文献欄から書誌情報を抽出した．タイトルページからの書誌情報抽出では，英語論文 935 件，参考文献欄からの書誌情報抽出では，英語論文 500 件を対象に，著者名や論文誌名など 13 項目の書誌情報を抽出する実験を行った．タイトルページからの書誌情報抽出の平均 F 値は 0.939，参考文献欄からの書誌情報抽出の平均 F 値は 0.915 であった．一方，Councill らは，CRF に基いて参考文献欄から書誌情報を抽出するオープンソースのツール“ParsCit”を開発した．ParsCit は，空白文字をデリミタとして，英文の参考文献

文字列をトークン列に変換し，そのトークン列に書誌要素ラベルを付与する．Cora データセット [10] を対象に，著者名やタイトルなど 13 項目の書誌情報を抽出する実験を行ったところ，その平均 F 値は 0.950 であった．また，Do らは，CRF を利用して学術論文の著者とその所属機関の組み合わせを発見する情報抽出システム“Enlil”を開発した．ACM Digital Library，ACL Anthology，Cross Disciplinary Corpus を対象に，著者名を抽出し著者と所属機関を照合する実験を行った．著者名の抽出において F 値は ACM Digital Library で 0.946，ACL Anthology で 0.918，Cross Disciplinary Corpus で 0.916 であった．著者と所属機関の照合においては F 値は ACM Digital Library で 0.889，ACL Anthology で 0.836，Cross Disciplinary Corpus で 0.870 であった．近年では，Cuong らが CRF を拡張した higher order semi-Markov CRF（HOCRF）を提案した．Cora，FLUX-XIM，ICONIP，humanities データセットを対象に，Council らと同様に 13 項目の書誌情報を抽出する実験を行ったところ，その全てのデータセットについての平均 F 値は 0.943 であった．

2.2 単語の分散表現

NN では，文字や文字列をベクトルとしてモデルへ入力する必要がある．ベクトル化には，文字や単語を対応する ID に置き換えてベクトル化する方法や，one-hot 表現でベクトル化する方法がある．しかし，one-hot 表現では，ベクトルの各要素に単語が対応するため，次元が語彙数となり，一般に数万次元になる．そのため，低次元の密なベクトルを得る方法として分散表現と呼ばれる数値ベクトルで表現する方法がある．単語の分散表現を獲得する手法として，Mikolov らによって提案された word2vec [11,12] があげられる．word2vec は，文章中の各単語を周辺の単語から予測するというタスクを NN で学習し，中間層の値を出力することによって，文章や単語の語順を考慮した単語の特徴を表すベクトルを獲得する．文章中の単語を予測する方法としては，中心の単語から周辺の単語を予測するもの，周辺の単語から中心の単語を予測するものがある．前者は Skip-gram モデル，後者は Continuous Bag-of-Words（CBOW）モデルと呼ばれる．単語の特徴を表すベクトルの次元数は中間層のノード数と等しく，word2vec では数百程度がよく用いられる．よって，one-hot 表現では次元が語彙数に依存し，一般に数万次元となるが，word2vec では数百程度に次元を削減することができる．

3. NN モデルによる参考文献書誌情報抽出

3.1 参考文献書誌情報抽出

本研究では，参考文献文字列を図 2 のように，まずトークン列に変換し，そのトークン列から著者名などの主要な書誌情報を抽出する．本研究では [4] と同様にトークン列への変換は人手で行い，トークン列への書誌要素ラベル付与は，提案する NN モデルを用いて行う．参考文献文字列のトークン列への自動変換については CRF を利用する方法が [3] で提案されている．参考文献文字列から抽出する書誌情報の一覧と，それに対応する書誌要素ラベルを表 1 にまとめる．表 1 の Other は他の

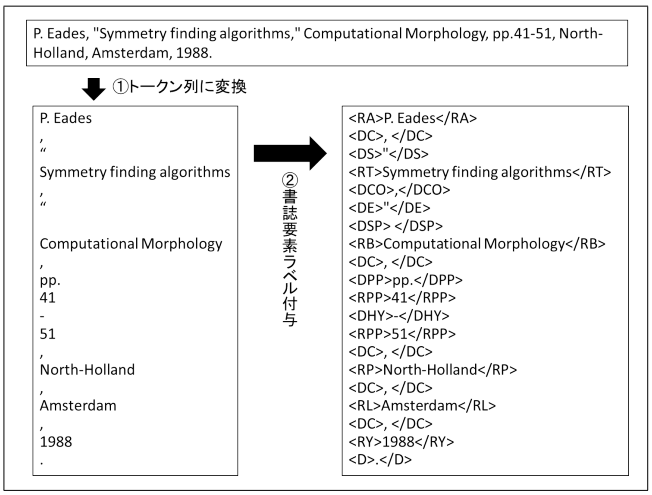

図 2　参考文献書誌情報抽出

表 1　抽出する書誌情報 [4]

書誌要素	書誌要素ラベル
Author	RA
Editor	RE
Translator	RTR
Author Other	RTR
Title	RT
Booktitle	RBT
Journal	RW
Conference	RC
Volume	RV
Number	RN
Page	RPP
Publisher	RP
Day	RD
Month	RM
Year	RY
Location	RL
URL	RURL
Other	ROT

どの書誌要素にも分類されない書誌要素であり，具体的には所属機関などが含まれる．本研究では，変換されたトークン列の各トークンに対し，<RA> や <RT> などの書誌要素ラベル，<DC>（カンマ＋空白）などのデリミタラベルを付与する．なお，図 2 で D から始まるラベルはデリミタラベルを表し，24 種類が定義されている [4].

3.2 提案モデル

本研究で用いる NN モデルの構造を図 3 に示す．図 3 において Input_1，Input_2 は入力層，Embedding は Embedding 層，Convolutional neural network（CNN）は畳み込み層，Bi-LSTM は Bi-directional Long-short-term memory（Bi-LSTM）層，Concatenate はマージ層，Dense は全結合層を表す．

ここで，Input_1 (word) への入力は参考文献文字列の各ワードがそれぞれ ID に変換されて得られるベクトルとする．ここで，ワードとは参考文献文字列をデリミタを用いて分割した各

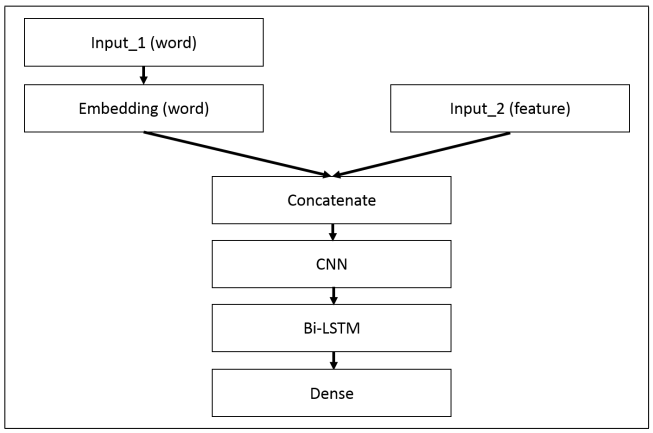

図 3　提案モデル

文字列を表す．また一つ以上のワードが一つのトークンを構成する．また，入力は系列長をそろえる必要があるため，入力ベクトルの先頭を参考文献文字列の最大系列長に合わすように 0 でパディングする．また，Input_2 (feature) への入力は参考文献文字列の各ワードの特徴を示す素性を表すベクトルを入力とするものである．この素性ベクトルについては，3.3 節で説明する．

Embedding 層は Input_1 の出力を受け取り，分散表現への埋め込みを行う層である．本来，Embedding 層ではこの変換の重みを学習によって更新するが，Embedding 層の重みは学習によって更新しない．また，本モデルでは Embedding 層の重みの初期値には，実験で用いる参考文献文字列コーパスの word2vec で事前に学習したモデルにおける分散表現を利用する．

その後の処理は，他の NN モデルと同様に図 3 の構造通りに処理する．word の Embedding 層と feature の入力層の出力を Concatenate でマージし，マージ層の出力を畳み込み層への入力とする．その後，畳み込み層の出力を Bi-directional LSTM 層への入力とする．最後に Bi-directional LSTM 層の出力を全結合層へ入力し，モデルの最終的な出力を得る．

本モデルは，参考文献文字列の各ワードとその素性を入力として各ワードに付与されうるラベルの確率を出力する．その後，参考文献文字列の各トークンを構成するワードの条件付き確率の相加平均を求め，その値に基づいて最大確率のラベルを選び，各トークンに付与するラベルを決定する．例えば，図 2 の“ P. Eades ”というトークンはデリミタで分割され，“ P ”，“ . ”，“ Eades ”がワードの入力となる．よって，ワード毎に付与されうるラベルの確率が得られるので，その確率の相加平均に基づき，“ P. Eades ”というトークンに付与するラベルを決定する．

3.3　素性ベクトル

本稿では，ワードの分散表現とは別に，参考文献文字列の各ワードの素性を NN の入力とする．用いる素性は先行研究である川上らが CRF による書誌情報抽出で用いた素性 [4] を参考にした．ただし，川上らはトークンの特徴を素性としたが，本研究ではそれをワードの特徴に置き換える．また，川上らが使った素性のうち，ワード自身，ワードを小文字にした文字列の素性は使用しない．本稿で用いるワードの素性を表 2 にまとめる．この素性ベクトルは 46 の素性で構成されており生成されるベクトルは 46 次元となる．これらは言語的な素性のみで，ページ内での位置情報などのレイアウトに関する素性はない．素性には，ワードのワード列における出現位置，文字数，ワードの文字種とその割合，ワードの先頭・末尾から四文字目までの文字コード，数字や記号などの特定の文字や特徴的な文字列の有無，各種辞書のエントリの有無などがある．ここで，特徴的な文字列とは，例えば“ Academic ”のことで，この文字列を含むワードは Publisher を表す書誌要素である可能性が高い．また，辞書としては，人名 (注1)，論文誌名 (注2)，会議名 (注3)，出版社名 (注4)，地名 (注5)，月名の辞書と，学会誌名などの分類困難なものをまとめた辞書の 7 種類がある．表 2 の各素性の括弧内の数字はワードの相対位置を表しており，0 が現在のワード，また，$i \in \{-4, -3, -2, -1, 0, 1, 2, 3, 4\}$ である．なお，表 2 で，“ 数 ”はその素性に関する実際の素性数を表す．例えば，<num_word(0)> の場合，“ 大文字のみ ”，“ 先頭のみ大文字 ”，“ 小文字のみ ”の各単語数と総単語数という四つの素性を持つ．

表 2　素性ベクトルを構成する素性

素性	数	内容
<word_ab_pos(0)>	1	ワード列における絶対的な出現位置
<word_re_pos(0)>	1	ワード列における相対的な出現位置
<num_char(0)>	1	ワードの文字数
<num_word(0)>	4	ワード内の単語数
<num_period(0)>	4	ワード内のピリオド数
<f_kanji(0)>	1	ワード内の漢字数の割合
<f_hiragana(0)>	1	ワード内のひらがな数の割合
<f_katakana(0)>	1	ワード内のカタカナ数の割合
<f_alphabet(0)>	1	ワード内の全角アルファベット数の割合
<f_digit(0)>	1	ワード内の全角数字数の割合
<h_alphabet(0)>	1	ワード内の半角アルファベット数の割合
<h_digit(0)>	1	ワード内の半角数字数の割合
<h_symbol(0)>	1	ワード内の記号数の割合
<first_1-4_string(0)>	4	ワードの先頭から四文字目までの文字コード
<last_1-4_string(0)>	4	ワードの末尾から四文字目までの文字コード
<last_char(i)>	1	ワードの最後の文字種
<capital(i)>	1	ワード中の大文字の有無
<digit(i)>	1	ワード中の数字の有無
<symbol(i)>	2	ワード中の記号の有無
<keyword(i)>	2	ワード中の特徴的な文字列の有無
<dictionary(i)>	8	辞書的素性
<word_token(0)>	1	参考文献文字列のワード数
<editor(0)>	1	参考文献文字列中の Editor に関する記述の有無
<URL(0)>	1	参考文献文字列中の URL に関する記述の有無
< y(-1), y(0)>	1	ラベルの遷移

4.　評 価 実 験

4.1　実 験 環 境

提案モデルによる書誌情報の抽出精度を検証するため，評価実験を行う．実験データとして，以下の参考文献文字列コーパスを利用する．

IEICE-E　2000 年の電子情報通信学会英文論文誌に含まれる参考文献文字列 4,497 件

(注1)：http://www.census.gov/genealogy/names/など
(注2)：http://science.thomsonreuters.com など
(注3)：http://www.allconferences.com/など
(注4)：http://www.narosa.com/nbd/PublisherDistributed.asp など
(注5)：http://www.fallingrain.com/world/index.html など

表 3　IEICE-E における書誌情報抽出精度（提案モデル）

		畳み込み層	
	Bi-LSTM 層数	あり	なし
提案モデル	1	0.8929	0.8749
	2	0.9015	0.8987
	3	0.8973	0.8899
CRF [4]		0.9703	

表 4　IEICE-E における書誌情報抽出精度（word モデル）

		畳み込み層	
	Bi-LSTM 層数	あり	なし
word モデル	1	0.8796	0.8841
	2	0.8908	0.8950
	3	0.8890	0.8931

表 5　IEICE-E における書誌情報抽出精度（feature モデル）

		畳み込み層	
	Bi-LSTM 層数	あり	なし
feature モデル	1	0.8810	0.8241
	2	0.8914	0.8727
	3	0.8890	0.8733

また，表 1 に示した，Author，Title 等の 18 種類の書誌要素ラベルは，評価の際には川上らの研究 [4] に倣い，RA，RE，RTR，RAOT を AUTHOR，RT，RBT を TITLE，RW，RC を JOURNAL，RV，RN，RPP を VOLUME，RP を PUBLISHER，RD を DAY，RM を MONTH，RY を YEAR，RL，RURL，ROT を OTHER と再分類し，書誌要素ラベルが同じ分類のものは正解判定において区別しない．また，デリミタラベルの種類の誤りも無視する．評価指標として，参考文献文字列を構成する全てのトークンに正しく書誌要素ラベルを付与できた参考文献文字列数を，全参考文献文字列数で割った書誌情報抽出精度を用いる．また，5 分割交差検定で書誌情報抽出精度は算出する．また，本研究で用いる NN モデルの実装には Keras [13] を用いる．

4.2 実験結果

5 分割交差検定を行うため，IEICE-E の参考文献文字列を五つに分割し，そのうち四つを学習データ，残りの一つをテストデータとする．本稿では，提案した NN モデルの書誌情報抽出精度の比較のため，川上ら [4] の CRF による書誌情報抽出精度をベースラインとし，書誌情報抽出精度を比較する．ベースラインとして用いる CRF の実装には CRF++ 0.58(注6) [14] を用いる．また，実験では図 3 の NN モデルに対し，畳み込み層の有無や，Bi-LSTM 層の数（1 から 3）を変更し，各モデル毎に書誌情報抽出精度を算出する．

ここで，提案した NN モデルのハイパーパラメータについて説明する．Embedding 層の重みの初期値に利用する word2vec のパラメータは minCount を 0 と設定し，残りのパラメータはデフォルトの設定とする．minCount を 0 とした理由は，参考文献文字列は著者名などの固有名詞を多く含んでおり参考文献文字列コーパスに一度しか登場しないワードも存在する．そのため，minCount を 1 以上にするとワードの分散表現が得られないことがあるからである．また，Embedding 層では，出力系列の次元を 100 とした．畳み込み層では，畳み込みフィルターの数を 80，1 フィルターがカバーするワードの数は 5 つで固定した．Bi-LSTM 層では，出力系列の次元を，1 層目は 80 次元，2 層目は 60 次元，3 層目は 40 次元とした．全結合層における活性化関数は sigmoid とした．また，最適化関数は rmsrop，損失関数の損失は平均二乗誤差，学習率は 0.001 とした．また，全ての NN モデルにおいてバッチサイズは 100 とし，学習回数は 100 回に固定した．

(注6)：http://crfpp.googlecode.com/svn/trunk/doc/index.html

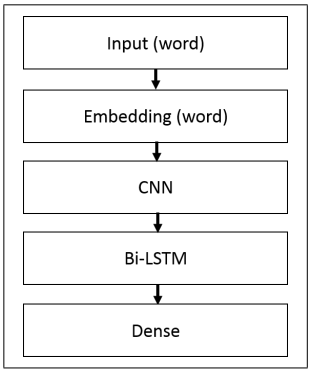

図 4　word モデル

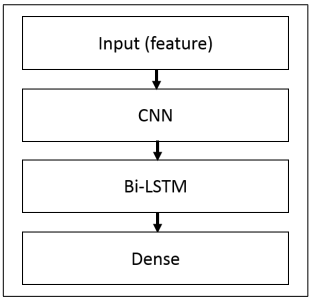

図 5　feature モデル

提案した NN モデルの書誌情報抽出の結果を表 3 に示す．表 3 よりどの NN モデルを用いても書誌情報抽出精度はベースラインである CRF の書誌情報抽出精度を大きく下回った．また，NN モデルでは Bi-LSTM 層の層数は 2 層のときが書誌情報抽出精度が最も高く，3 層になると書誌情報抽出精度が下がることがわかる．また，Bi-LSTM の層数にかかわらず，畳み込み層がある方が畳み込み層がないときに比べ，書誌情報抽出精度が高いことがわかる．

5. 考　　察

5.1 ワードの分散表現と素性を組み合わせた効果

図 3 の NN モデルはワードベクトルと素性ベクトルを入力としている．ここで，二つのベクトルを組み合わせたことによる書誌情報抽出精度の変化を確認するため，入力をワードベクトルのみ，素性ベクトルのみとした場合の書誌情報抽出精度を表 4，表 5 に示す．以後，入力がワードベクトルのみのモデルを word モデル，入力が素性ベクトルのみのモデルを feature

表 6　各書誌要素の抽出精度

	AUTHOR	TITLE	JOURNAL	VOLUME	PUBLISHER	DAY	MONTH	YEAR	OTHER
提案モデル	0.9973	0.9904	0.9819	0.9950	0.9189	0.9358	0.9955	0.9955	0.9262
CRF [4]	0.9978	0.9935	0.9966	0.9905	0.9983	0.9167	0.9977	0.9989	0.9717

表 7　トークン毎のラベル付与状況（提案モデル）

	AUTHOR	TITLE	JOURNAL	VOLUME	PUBLISHER	DAY	MONTH	YEAR	OTHER
AUTHOR	10,600	14	6			1			7
TITLE	2	4,554	20	3	11				6
JOURNAL	1	31	3,807	7	18				
VOLUME		9	16	11,807	3	6		1	18
PUBLISHER	5	4	21		872		3		43
DAY				3					2
MONTH		1			2		1,766		3
YEAR		4		4	1		1	4,440	7
OTHER	3	9	19	9	28		4	3	1,117

モデルと呼ぶ．また，word モデル，feature モデルの構造をそれぞれ図 4，図 5 に示す．NN モデルのハイパーパラメータについては 4.2 節と同様とする．また，4.2 節と同様に 5 分割交差検定で各モデルの書誌情報抽出精度を算出する．

表 4 より，入力がワードベクトルのみの場合，畳み込み層を設けない方が書誌情報抽出精度が高いことがわかる．また，表 5 より，入力が素性ベクトルのみの場合，畳み込み層を設けた方が書誌情報抽出精度が高く，その影響も顕著となっている．あらためて表 3 と比べると，二つのベクトルを Concatenate して入力とすることで，どちらか片方のベクトルを入力とする場合に比べて，書誌情報抽出精度が向上することがわかる．しかし，いずれのモデルでも Bi-LSTM 層数を 3 層にすると，書誌情報抽出精度が低下しているため，モデルを複雑にすると過学習が起きやすいと考えられる．

5.2　提案モデルにおけるエラー解析

4.2 節の実験では，最も書誌情報抽出精度が高かった NN モデルで 4,497 件の参考文献文字列中，443 件にラベル付与誤りが発生した．表 6 に各書誌要素ラベルの抽出精度を示す．表 6 では，4.2 節で最も書誌情報抽出精度が高いモデルを提案モデル，比較対象の CRF を CRF [4] と示す．表 3 より，畳み込み層とその後に Bi-LSTM 層を 2 層設けるモデルが最も書誌情報抽出精度が高いため，このモデルが提案モデルである．表 6 より，提案モデルを CRF と比較すると，“VOLUME”と“DAY”を除く書誌要素の抽出精度で提案モデルは CRF を下回っていることがわかる．特に，“PUBLISHER”，“OTHER”の二つの書誌要素の抽出精度が大きく劣っていることが分かる．また，“DAY”の書誌要素の抽出精度は優っているが，“DAY”は IEICE-E の参考文献文字列に 109 件しか出現しない書誌要素であるため，他の書誌要素に比べて，参考文献文字列全体の抽出精度に与える影響は小さい．

次に，表 7 に提案モデルにおける参考文献文字列のトークン毎の書誌要素ラベル付与の状況をまとめる．表 7 の縦は正解ラベル，横は提案モデルによる付与ラベルを表す．また，空欄は 0 件である．表 7 より，“TITLE”を“JOURNAL”や“PUBLISHER”，“JOURNAL”を“TITLE”，“PUBLISHER”を“JOURNAL”や“OTHER”，“OTHER”を“JOURNAL”や“PUBLISHER”に誤る場合が多いことが分かる．その中でも特に，“PUBLISHER”，“OTHER”のラベル付与の誤りが大きいことが分かる．“PUBLISHER”は IEICE-E の参考文献文字列に 949 回出現する書誌要素であるが，“AUTHOR”や“VOLUME”などの書誌要素に比べて，参考文献文字列に現れにくい．また，“OTHER”も開催地や URL などの参考文献文字列に現れにくい書誌要素である．このように，学習データの件数が少ない書誌要素については NN のパラメータ学習が十分とはいえない．

表 8　事前学習を行った提案モデルによる書誌情報抽出精度

	書誌情報抽出精度
事前学習モデル（全件）	0.9130
事前学習モデル（全件の半分）	0.9102
提案モデル（事前学習なし）	0.9015

5.3　提案モデルの事前学習

5.2 で述べたように，“PUBLISHER”や“OTHER”などの書誌要素が少ないため，NN のパラメータが十分に学習できなかった．そこで，IEICE-E とは異なる雑誌で提案モデルの学習を事前に行い，提案モデルの NN の重みを更新する．その後，更新した NN の重みを初期値として，IEICE-E で提案モデルの学習を行い書誌情報抽出精度を算出する．事前学習を行うためのデータとして，以下の参考文献文字列コーパスを利用する．IEEE-CS　1952 年から 2012 年までの IEEE Trans. Computers に含まれる参考文献文字列の引用回数上位 4,770 件

事前学習するモデルは 4.2 節で最も書誌情報抽出精度が高かったモデルとする．また，ハイパーパラメータは 4.2 節と同様とする．また，事前学習に用いる学習データの量と書誌情報抽出精度の関係を確認するため，IEEE-CS を全件事前学習に使用する場合と全件のうち半分（2,385 件）を事前学習に使用する場合の二つで NN の重みの初期値を求める．このとき，全件を事前学習に使用した場合を，事前学習モデル（全件），半

分を事前学習に使用した場合を事前学習モデル（全件の半分）とする．

事前学習を行った提案モデルによる書誌情報抽出精度を表 8 に示す．表 8 より，書誌情報の抽出対象とは異なる雑誌の参考文献文字列を全件用いて事前学習を行うことで書誌情報抽出精度が約 1.2 ポイント上昇した．また，全件のうち半分を事前学習に用いた場合でも，書誌情報抽出精度が約 0.9 ポイント上昇した．このことから，事前学習で求めた重みを提案モデルの重みの初期値とすることで，対象の雑誌の参考文献文字列における書誌情報抽出精度が向上することがわかる．また，事前学習の学習データを増やすことで，さらなる精度向上が期待できる．

6. ま と め

本稿では，参考文献書誌情報抽出において単語の分散表現とその単語の特徴を示す素性を入力とする NN を提案し，先行研究の CRF による書誌情報抽出精度と比較をした．実験では，電子情報通信学会英文論文誌の参考文献文字列 4,497 件に対し，その参考文献文字列を構成する各ワードに提案した NN モデルを用いて書誌要素ラベルを付与した．

提案モデルについて実験を行った結果，最も書誌情報抽出精度が高かったのは，二つの入力をマージした後に畳み込み層とその後に 2 層の Bi-LSTM 層を設けたモデルであり，その分類精度は 90.15%だった．しかし比較対象とした CRF に書誌情報抽出精度は及ばなかった．そこで，提案モデルにおける参考文献文字列の各書誌要素のエラー解析を行ったところ，“ PUBLISHER ”や“ OTHER ”のような参考文献文字列に出現しにくい要素に推定誤りが多いことを確認した．よって，このような書誌要素について NN のパラメータ学習が不十分だった可能性が高い．そこで本稿では，書誌情報の抽出対象とは異なる雑誌で提案モデルの事前学習を行って NN の重みの初期値を調整した．事前学習によって更新された NN の重みの初期値を用いて実験を行った結果，事前学習を行わない場合に比べ，書誌情報抽出精度が向上することを確認した．また，事前学習に用いる学習データ件数が多い方が書誌情報抽出精度が高かったため，事前学習の学習データを増やすことでさらに書誌情報抽出精度が向上する見込みがある．

今後の課題として，事前学習に用いる NN のハイパーパラメータの調整，事前学習に用いる学習データの量と質の検討などが挙げられる．また，他の参考文献文字列コーパスによる実験，日本語の参考文献文字列への対応が挙げられる．

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] D. A. Pereira, E. E. B. da Silva, and A. A. A. Esmin, “ Disambiguating publication venue titles using association rules, ” in Proc. of JCDL 2014, 2014, pp. 77-85.
[2] Köpcke, A. Thor, and E. Rahm, “Evaluation of entity resolution approaches on real-world match problems, ” PVLDB, vol. 3, no. 1, pp. 484-493, 2010.
[3] M. Ohta, D. Arauchi, A. Takasu, and J. Adachi, “ Empirical Ecaluation of CRF-Based Bibliography Extraction from Reference Strings”, in Proc. of IAPR DAS 2014, 2014, pp. 287-292.
[4] 川上尚慶, 太田学, 高須淳宏, 安達淳, “ 少量学習データによる参考文献書誌情報抽出精度の向上 ”, 情報処理学会論文誌データベース, vol. 8, no. 2, pp. 18-29, 2015.
[5] J. Lafferty, A. McCallum, and F. Pereira, “ Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data ”, in Proc. of ICML 2001, 2001, pp. 282-289.
[6] F. Peng, and A. McCallum, “ Accurate Information Extraction from Research Papers Using Conditional Random Fields ”, in Proc. HLT-NAACL 2004, 2004, pp. 329-336.
[7] I. G. Councill, C. L. Giles, and M. Y. Kan, “ ParsCit: an open-source CRF reference string parsing package, ” in Proc. of LREC 2008, 2008, pp. 661-667.
[8] H. H. N. Do, M. K. Chandrasekaran, P. S. Cho, and M. Y. Kan, “ Extracting and matching authors and affiliations in scholarly documents, ” in Proc. of JCDL 2013, 2013, pp. 219-228.
[9] N. V. Cuong, M. K. Chandrasekaran, M. Y. Kan, and W. S. Lee, “ Scholarly document information extraction using extensible features for efficient higher order semi-CRFs, ” in Proc. of JCDL 2015, 2015, pp. 61-64.
[10] A. McCallum, K. Nigam, J. Rennie, and K. Seymore, “ Automating the Construction of Internet Portals with Machine Learning ”, Information Retrieval, vol. 3, no. 2, pp. 127-163, 2000.
[11] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “ Distributed representations of words and phrases and their compositionality ”, in Proc. of NIPS 2013, 2013, pp. 3111-3119.
[12] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “ Efficient estimation of word representations in vector space ”, arXiv preprint arXiv:1301.3781, 2013, pp. 1-12.
[13] F. Chollet, Keras. https://github.com/fchollet/keras, 2015.
[14] T. Kudo, K. Yamamoto, and Y. Matsumoto, “ Applying Conditional Random Fields to Japanese Morphological Analysis ”, in Proc. of EMNLP 2004, 2004, pp. 230-237.