日本語の食べ物・飲み物表現の抽出における文字 CNN の効果

新堂 安孝 † 友利 涼 ‡ 兼村 厚範 † 宮尾 祐介 † 森 信介 ‡‡

† 株式会社デジタルガレージ DG Lab
‡ 京都大学 情報学研究科
‡‡ 京都大学 学術情報メディアセンター

1 はじめに

1.1 文字 CNN を使った自然言語処理技術の現状

CNN (Convolutional Neural Network [1]) は画像処理においてよく用いられる手法である [2] が，自然言語処理 [3] でも CNN を用いた手法が高い性能を示すことが知られている．例えば，固有表現抽出および文書分類で文字 CNN を用いた研究成果 [4, 5, 6, 7] がある．

文字 CNN を用いたこれらの研究では評価対象として，CoNLL-2003*1[8] や AG News*2 など表音文字の言語のコーパス，もしくは表語文字を発音情報に置換して表音文字に準ずる内容にしたコーパスを用いている．このため，日本語などの表語文字を多く含む言語に対し，文字 CNN が自然言語処理の性能向上にどのように寄与するか，まだ分かっていない．

1.2 本研究の概要

本研究では，食べログ*3 のレビュー・コーパスから食べ物・飲み物表現を固有表現として抽出するタスクを設定し，日本語の固有表現抽出で文字 CNN がどういった振る舞いをするのかを実験・分析した．この作業を通して以下が判明した．

(1) 文字 CNN を用いることで，固有表現抽出モデルの学習データに含まれる単語の先頭文字列や末尾文字列を参考に，同データ外の単語を適切に扱えるようになり再現率の向上が実現する．ただし食べログ・コーパスにおいては，文字 CNN のパラメータを大きくする必要があった．
(2) 文字 CNN と事前学習ありの単語ベクトルを組み合わせると，固有表現抽出の対象データの中に事前学習モデルに含まれない表現が多い場合には，事前学習ありの単語ベクトル単独より再現率が高くなる．

2 関連研究

文字 CNN を用いた既存研究には，固有表現抽出に関する研究 [5] や文書分類に関する研究 [6] がある．これらはいずれも $\mathbb{R}^m$ の文字ベクトルを並べて $\mathbb{R}^{mn}$ のデータを作った ($m, n \in \mathbb{N}$) 上で CNN を適用している．前者は，文字 CNN の入出力を単語単位で扱い，同出力を単語ベクトルと結合する形で，BiLSTM (Bidirectional Long-Short Term Memory [9, 10]) の入力とし，その後段に最終的な推定処理のために CRF (Conditional Random Field [11]) を用いている．

表 1 GloVe モデル

モデル名	学習データ	学習データの規模
J6	食べログ	6×10^6 単語
J8	食べログ	6×10^8 単語
E9	Wikipedia (en) + Gigaword	6×10^9 単語

表 2 学習・開発・試験データの規模 (単語数)

用途	学習	開発	試験
食べログ	3.8×10^5	1.7×10^5	1.6×10^5
CoNLL-2003	2.0×10^5	0.5×10^5	0.5×10^5

3 実験に用いたアルゴリズムとデータ

3.1 アルゴリズム

本研究では，固有表現抽出アルゴリズムとして先行研究である BiLSTM-CNNs-CRF [5] を採用し，その実装の 1 つである NCRF++*4[12] を用いた．また，基本的な設定項目については，同先行研究と同じ値 (e.g. 単語ベクトルは 100 次元) を用いた．

3.2 データ

本研究では，食べ物・飲み物表現にタグを付与した食べログ・コーパス [13] を用いた．また，比較対象として CoNLL-2003 の英語コーパスも用いた．それぞれのコーパスに対し，事前学習した単語ベクトルの GloVe [14] モデルを表 1 の要領で用意し，必要に応じて併用した．E9 は，先行研究 [5] で用いられたもので，GloVe の公式サイト*5 で配布されている．食べログ・コーパスに対する単語分割処理には，KyTea*6[15] を用いた．また，後述の予備実験・本実験に際し，各コーパスはそれぞれ表 2 の要領で学習・開発・試験データに分割した．

開発・試験データそれぞれに含まれる各固有表現が，

連絡先: 新堂 安孝 <shindo@dglab.com>

*1 https://www.clips.uantwerpen.be/conll2003/ner/
*2 http://www.di.unipi.it/~gulli/AG_corpus_of_news_articles.html
*3 https://tabelog.com/
*4 Ver.0.1: https://github.com/jiesutd/NCRFpp
*5 https://nlp.stanford.edu/projects/glove/
*6 Ver.4.7: http://www.phontron.com/kytea/

表 3　食べログ・コーパスの固有表現の内訳

用途	開発データ		試験データ	
学習リソース	リソース内	外	リソース内	外
学習データ	6657	1085	6340	928
学習データ + J6	7216	526	6831	437
学習データ + J8	7685	57	7210	58

表 4　CoNLL-2003 の固有表現の内訳

用途	開発データ		試験データ	
学習リソース	リソース内	外	リソース内	外
学習データ	4514	1428	3597	2049
学習データ + E9	4514	1428	3599	2047

表 5　予備実験結果 (食べログ，開発データ，F 値)

D \ F	15	30	60	120	240	480	960
10	0.597	0.644	0.680	0.684	0.688		
20	0.652	0.732	0.747	0.755	0.760		
40	0.713	0.764	0.770	0.787	0.792		
80	0.752	0.775	0.790	0.804	0.803	0.809	0.729
160	0.765	0.786	0.802	0.811	0.820	0.818	0.820
320			0.809	0.816	0.821	0.826	0.821
640			0.676	0.822	0.824	0.819	0.822
1280			0.671	0.819	0.821	0.824	0.812

表 6　予備実験結果 (CoNLL-2003，開発データ，F 値)

D \ F	15	30	60	120	240	480	960
10	0.666	0.688	0.698	0.693	0.696		
20	0.694	0.739	0.774	0.791	0.810		
40	0.718	0.773	0.807	0.836	0.854		
80	0.745	0.800	0.835	0.861	0.873	0.880	0.886
160	0.735	0.815	0.845	0.871	0.886	0.891	0.898
320			0.856	0.876	0.891	0.901	0.902
640			0.861	0.882	0.898	0.903	0.906
1280			0.866	0.886	0.896	0.900	0.905

BiLSTM-CNNs-CRF のモデル学習に用いたリソース内の単語だけで表現できるか，それとも同リソース外の単語を含むかについて，計測した結果を表 3, 4 に記す．なお，表 4 において E9 の有無で変化が少ないことは，以下の不整合が原因と考えられる．

- CoNLL-2003 のタスク定義が人名・組織名・地名を中心に抽出するものであることから，その学習・開発・試験データ内の固有表現のほぼ全てが大文字を含む．
- E9 は uncase 処理を経たデータから学習されている．

4　予備実験

本実験に先立ち，文字 CNN のパラメータを検討すべく予備実験を行った．NCRF++ に対し単語ベクトルを使用せずに，文字 CNN の 2 つのパラメータ，文字ベクトルの次元数と CNN のフィルタ数 (詳細は [16] を参照のこと．以降，それぞれ D, F と表記する) を調整し，開発データの推定結果の F 値を基準に評価した．食べログ・コーパスと CoNLL-2003，それぞれの結果を表 5, 6 に示す．

これら各枠の結果は，開発データに対して D と F の各組み合わせで得られる F 値の最大値に準じる．食べログ・コーパスと CoNLL-2003 のいずれでも，表の左上から右下に進むにしたがって F 値が向上する傾向が見られる．本実験では，両コーパスで F 値がほぼ飽和した (D, F) = (640, 240) のパラメータを，先行研究 [5] に準じるパラメータ (D, F) = (10, 30) とともに用いる．

5　本実験

5.1　実験概要

本実験では，食べログ・コーパスと CoNLL-2003 を用い，以下 2 種類の実験を行った．

(A) 予備実験と同様，D と F の各組み合わせに対し，開発データでどこまで高い性能が得られるか確認．

(B) 上記 (A) で得られた調整結果を用い，試験データで通常の手順の実験を実施．

なお，文字 CNN については「未使用」「(D, F) = (10, 30)」「(D, F) = (640, 240)」の 3 つの選択肢 (節 4 を参照のこと) を，単語ベクトルについては「未使用」「事前学習なし」「事前学習あり (GloVe モデルは別途指定)」の 3 つの選択肢を，それぞれ用いた．

表 7, 9 に (A) の結果を，表 8, 10 に (B) の結果を，それぞれ記す．以降の節 5.2, 5.3, 5.4 にて，文字 CNN の効果に注目する形で実験結果の詳細について議論する．

5.2　実験結果分析 (1)

文字 CNN 単独の性質を確認すべく以下を比較した．

(a) 文字 CNN「未使用」×単語ベクトル「事前学習なし」

(b) 文字 CNN「(640, 240)」×単語ベクトル「未使用」

食べログ・コーパスにおいては，(a) が抽出できなかった学習データ外の単語を含む固有表現を (b) ではよく抽出していた．このケースでは該当単語が学習データ内の固有表現である単語の先頭文字列や末尾文字列を含むことが多かった．以下に同ケースの固有表現の例を示す (太字が学習データ外の単語，かっこ内が太字単語と先頭文字列や末尾文字列が共通する学習データ内にある固有表現である単語)．

- 河豚␣の␣**ひれ酒**　(梅酒，濁り酒，冷酒)
- **時鮭**␣の␣自家␣製␣筋子　(塩鮭，秋鮭，鮭)
- **きのこご飯**　(きのこ，ご飯，炒飯)
- **焼きしゃぶ**　(焼きそば，焼肉，鯛しゃぶ)
- カレー␣**温麺**　(温玉，湯麺，素麺)
- カレー␣**つけ麺**　(つけもの，冷麺，涼麺)
- **ショコラノワール**　(ショコラ，シロノワール)
- **フォレノワール**　(シロノワール，ピノノワール)
- 熱␣い␣**ダシ汁**　(しじみ汁，冷や汁，出汁)
- **牡蠣鍋**　(牡蠣，つみれ鍋，熊鍋)

学習データ内の単語と先頭・末尾文字列が等しい学習データ外の単語を適切に抽出するこの性質により，(a) より (b) の再現率が高くなっているものと思われる．

CoNLL-2003 においても，(a) が抽出できなかった学習データ外の単語を含む固有表現を (b) ではよく抽出しており，該

表 7　本実験結果 (食ベログ，開発データ)

単語ベクトル	未使用			事前学習なし			事前学習あり w/J6			事前学習あり w/J8		
文字 CNN (D, F)	F 値	精度	再現率	F 値	精度	再現率	F 値	精度	再現率	F 値	精度	再現率
未使用				0.807	0.840	0.779	0.835	0.850	0.820	0.858	0.859	0.857
(10, 30)	0.644	0.772	0.552	0.807	0.841	0.776	0.840	0.858	0.823	0.864	0.871	0.856
(640, 240)	0.824	0.831	0.817	0.827	0.845	0.809	0.842	0.855	0.828	0.858	0.865	0.851

表 8　本実験結果 (食ベログ，試験データ)

単語ベクトル	未使用			事前学習なし			事前学習あり w/J6			事前学習あり w/J8		
文字 CNN (D, F)	F 値	精度	再現率	F 値	精度	再現率	F 値	精度	再現率	F 値	精度	再現率
未使用				0.821	0.849	0.795	0.844	0.854	0.834	0.870	0.869	0.870
(10, 30)	0.663	0.784	0.575	0.819	0.846	0.793	0.851	0.866	0.837	0.871	0.878	0.864
(640, 240)	0.828	0.833	0.822	0.837	0.854	0.822	0.849	0.859	0.840	0.866	0.871	0.862

表 9　本実験結果 (CoNLL-2003，開発データ)

単語ベクトル	未使用			事前学習なし			事前学習あり w/E9		
文字 CNN (D, F)	F 値	精度	再現率	F 値	精度	再現率	F 値	精度	再現率
未使用				0.873	0.925	0.826	0.938	0.948	0.929
(10, 30)	0.688	0.720	0.660	0.892	0.906	0.879	0.946	0.947	0.946
(640, 240)	0.898	0.900	0.895	0.905	0.909	0.901	0.949	0.949	0.950

表 10　本実験結果 (CoNLL-2003，試験データ)

単語ベクトル	未使用			事前学習なし			事前学習あり w/E9		
文字 CNN (D, F)	F 値	精度	再現率	F 値	精度	再現率	F 値	精度	再現率
未使用				0.805	0.872	0.748	0.895	0.908	0.883
(10, 30)	0.690	0.714	0.668	0.827	0.839	0.816	0.914	0.915	0.913
(640, 240)	0.841	0.840	0.841	0.847	0.849	0.844	0.912	0.910	0.913

当単語のほぼ全てで先頭が大文字だった．この結果は人名・組織名・地名を中心に抽出する CoNLL-2003 のタスクの定義から自然なものである．先頭が大文字の単語 (i.e. 学習データに含まれる固有表現の先頭の単語と先頭文字列が重複する単語) を適切に抽出するこの性質により，(a) より (b) の再現率が高くなっているものと思われる．

5.3　実験結果分析 (2)

文字 CNN と事前学習なし単語ベクトルの併用時の性質を確認すべく以下を比較した．

(a) 文字 CNN「未使用」×単語ベクトル「事前学習なし」
(b) 文字 CNN「(10, 30)」×単語ベクトル「事前学習なし」
(c) 文字 CNN「(640, 240)」×単語ベクトル「事前学習なし」

食ベログ・コーパスにおいては，(a) と (b) では特別な違いは見られなかったが，(b) と (c) では後者が再現率が高く節 5.2 と類似の傾向が見られた．CoNLL-2003 においては，(a) と (b) では後者が再現率が高く節 5.2 と類似の傾向が見られ，また，(b) と (c) では学習データ内に単語のある・なしに関係なく後者が組織名を中心によりよく抽出していた．

CoNLL-2003 では，単純に単語先頭が大文字であるかどうかが固有表現抽出の重要要素であると考えられることから，文字 CNN のパラメータが小さくても十分な効果が出ているものと推測できる．一方で食ベログ・コーパスでは，単語中の先頭・末尾の文字列が固有表現抽出に重要な要素であると考えられることから，文字の種類の多さも含めて CoNLL-2003 に比べて複雑であり，文字 CNN のパラメータを大きくする必要があるものと推測できる．

5.4　実験結果分析 (3)

文字 CNN と事前学習済み単語ベクトルの併用時の性質を確認すべく以下を比較した．

(a) 文字 CNN「未使用」×単語ベクトル「事前学習あり」
(b) 文字 CNN「(640, 240)」×単語ベクトル「事前学習あり」

食ベログ・コーパスにおいては，単語ベクトルに J6 を用いた場合，(a) の時点で学習データや J6 に含まれない単語を含む固有表現 (表 3 を参照のこと) をよく抽出しているが，総数はわずかであるものの同様な固有表現で (a) より (b) が再現率が高く，節 5.2 と類似の傾向が見られた．

CoNLL-2003 においては，単語ベクトルに E9 を用いた場合，(a) の時点で学習データや E9 に含まれない単語を含む固

有表現 (特に人名．表 4 を参照のこと) をよく抽出しているが，総数は多くないものの同様な固有表現 (特に組織名) で (a) より (b) が再現率が高く，やはり節 5.2 と類似の傾向が見られた．

文字 CNN の効果が出なかった食べログ・コーパスの J8 を用いた結果と表 3 および上記の結果を考慮すると，単語ベクトルの事前学習がタスクで必要な表現を十分にカバーしている場合，文字 CNN の貢献は限定的であることが推測できる．同時に文字 CNN の学習が十分に進まなくなった可能性も考えられる．

6 まとめ・今後の展望

本研究では，食べログのレビュー・コーパスから食べ物・飲み物表現を固有表現として抽出するタスクを設定し，CoNLL-2003 (英語) の固有表現抽出と比較することで，日本語の固有表現抽出で文字 CNN がどういった振る舞いをするのかを実験・分析した．この作業を通して以下が判明した．

(1) 固有表現抽出モデルの学習に用いた言語リソース (単語ベクトルの事前学習モデルを含む) に含まれない単語 w に対し，文字 CNN を用いた同モデルは，食べログ・コーパスでは学習データに含まれる固有表現である単語の先頭・末尾文字列との重複がある場合，CoNLL-2003 では先頭文字が大文字である場合，単語 w が固有表現を構成すると判断する傾向がある．結果として再現率向上が実現する．ただし食べログ・コーパスでは，文字 CNN で大きなパラメータを与えないとこの傾向が見られない．

(2) 文字 CNN と事前学習ありの単語ベクトルを組み合わせると，抽出対象のデータに事前学習でカバーしきれない表現が多い場合には，事前学習ありの単語ベクトル単独より再現率が高くなるが，そうでない場合，文字 CNN の効果は限られたものになる．

今後，本研究で得られた知見を生かし，言語モデルを用いた単語の特徴量 [17] との併用を前提とした固有表現抽出に向く文字ベースの同特徴量の開発を進める予定である．

謝辞

本研究において各種サポートをしてくださった同僚およびカカクコム社の方々に感謝いたします．また，食べログ・コーパスのアノテート作業を担当してくださった方々に感謝いたします．

参考文献

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 2009.

[4] Jason P.C. Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 357–370, 2016.

[5] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1064–1074, 2016.

[6] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Proceedings of the 29th Conference on Neural Information Processing Systems*, pp. 649–657, 2015.

[7] Alexis Conneau, Holger Schwenk, Yann LeCun, and Loïc Barrault. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Vol. 1, pp. 1107–1116, 2017.

[8] Erik F. Tjong, Kim Sang, and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning*, pp. 142–147, 2003.

[9] Sepp Hochreiter and Jurgen Schmidhuder. Long short-term memory. *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

[10] Mike Schuster and Kuldip K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp. 2673–2681, 1997.

[11] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pp. 282–289, 2001.

[12] Jie Yang, Shuailong Liang, and Yue Zhang. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3879–3889, 2018.

[13] 新堂安孝, 友利涼, 富田紘平, 兼村厚範, 森信介. レストラン・レビューにおける食べ物・飲み物表現の抽出. 電子情報通信学会技術研究報告 NLC2018-26, pp. 97–102, 2018.

[14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1532–1543, 2014.

[15] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, pp. 529–533, 2011.

[16] Cícero dos Santos and Victor Guimarães. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entities Workshop*, pp. 25–33, 2015.

[17] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 2227–2237, 2018.