Transformer と ⟨CLS⟩ ベクトルを用いた Span-based 固有表現抽出手法

宮崎 太郎 [1]　Simon Clippingdale[2]　後藤 淳 [1]
[1]NHK 放送技術研究所　[2]NHK 財団
{miyazaki.t-jw, goto.j-fw}@nhk.or.jp
simon.c-fe@nhk-fdn.or.jp

概要

固有表現抽出は自然言語処理の基盤技術のひとつであり，自然言語処理を利用したシステムで広く応用されている．従来では CRF (Conditional Random Field) を用いた系列ラベリングを用いる手法が一般的であったが，近年では複数単語のまとまりを入力して，入力のまとまりが固有表現であるか判定する Span-based 手法も用いられる．本稿では，Span-based 手法による固有表現抽出手法について述べる．複数単語からのベクトルのまとめ上げに BERT などの事前学習モデルで用いられる特別な Token ⟨CLS⟩ の考え方を用いることで，従来一般的に用いられていた LSTM (Long Short-term Memory) や Max Pooling を上回る性能が得られることを確認した．

1　はじめに

固有表現抽出 (Named Entity Recognition: NER) は文章中の人名，地名，組織名などの固有表現を抽出，分類するもので，自然言語処理の基盤技術のひとつである．情報抽出 [1] や センシティブな情報の匿名化 [2] など，自然言語を扱う様々なシステム内で広く応用されている．

固有表現は日々新しいものが生まれる．それらの新語に対応した NER を実現するには，単語の表層に頼らない分類手法を用いる必要がある．ニューラルネットワークを用いた NER では，単語をベクトル表現した上で，入力の各単語が固有表現またはその一部を構成する単語かどうかを判定する系列ラベリング [3, 4] を用いるのが一般的である（図 1-(a)）．しかし，系列ラベリングでは入れ子構造の固有表現に対応することができない．そこで，近年では入れ子構造の固有表現に対応した手法も多く提案されている [5, 6, 7]．本稿では，入れ子構造に対応した手法の一つである Span-based 手法を用いた固有表現抽出手法について述べる．

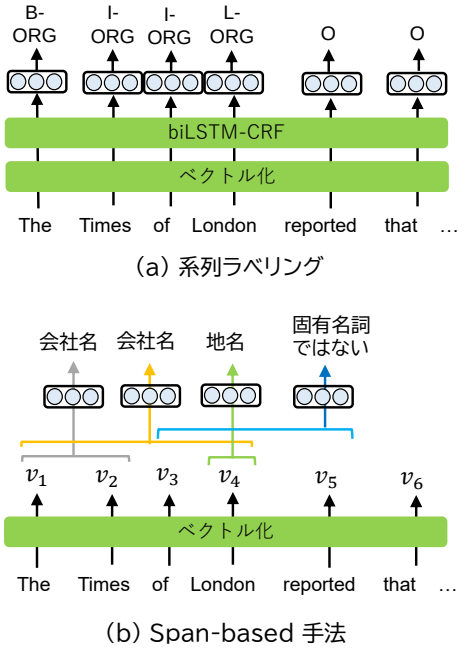

図 1　固有表現抽出手法．系列ラベリングでは単語ごとに固有表現の開始（B-）や途中（I-），最後（L-）や固有表現ではない（O）などのラベルを付与していく．Span-based 手法では，単語のまとまり（Span）を表すベクトルを計算し，Span 全体が固有表現であるかどうか判定する．

Span-based 手法の概略を図 1-(b) に示す．判定の対象とする単語列から単一のベクトルを作成し，固有表現であるか，また固有表現である場合はその種類を判定する．判定対象の単語列に含まれる Token 数は可変であることから，この長さが異なるベクトルを単一のベクトルに変換する必要がある．従来では LSTM (Long Short-term Memory) や Max pooling が用いられることが多い．しかし LSTM は最初や最後に入力される Token のベクトルからの影響を強く受けやすく，また Max pooling では 2 つの共通する単語を多く持つ単語列，例えば「NHK 放送技術研究

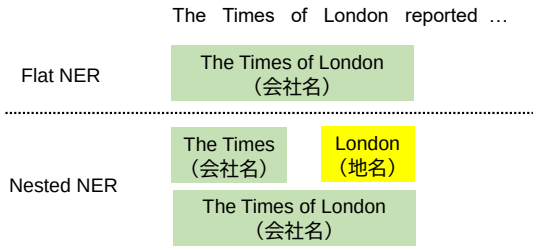

図 2 Flat NER と Nested NER．Flat NER では 1 つの単語は 1 つの固有表現にのみ属するが，Nested NER では複数の固有表現に属する場合がある．

所」と「NHK 放送技術研究所の」の間で出力ベクトルが似通ってしまうことが問題となる．

そこで我々は，これらの問題を解決するために，ベクトルのまとめ上げ部分に Transformer を用いた．Transformer は一般的に入力の行列と同じ大きさのベクトルを出力するが，ここに，BERT[8] 等で用いられる ⟨CLS⟩ を用いてまとめ上げる手法を提案する．

2 関連研究

NER のタスクは，入れ子構造の固有表現を考慮せずに 1 つの単語は最大で 1 つの固有表現に含まれるとして扱う Flat NER と，入れ子構造を考慮した Nested NER の 2 つに大きく分類できる（図 2）．Flat NER は古くから研究されているタスクであり [9]，一方の Nested NER は比較的新しく提案されたタスクである [5]．本稿では Flat NER を扱う．

Flat NER

Flat NER では，多くの手法は CRF (Conditional Random Field) を用いた系列ラベリングを用いる [3, 4]．CRF の前段の双方向 LSTM (Long Short-Term Memory) により，前後の文脈を加味した単語ベクトルを作成し，それぞれの単語が固有表現またはその一部であるか判定する．Akbik et al. は文字ベースの言語モデルを使用し，文字ごとのベクトルを双方向 RNN (Recurrent Neural Network) に入力することで，学習データに多くは出現しない低頻度語の分類精度を向上する手法を提案した [10]．

近年では外部文脈を用いた性能向上の手法も多く提案されている．Yamada et al. は対象とする文の前後の文の情報を活用することで性能が向上することを示した [11]．また，Wang et al. は前後の文ではなく，NER の対象とするテキストをクエリとした検索により獲得した外部の文章を，対象とするテキストと合わせてモデルに入力する手法を提案し，有効性を報告した [12]．

Nested NER

Finkel et al. [5] により Nested NER のタスクが提唱されて以来，多くの手法が提案されている．Lin et al. は，固有表現の特徴的な単語を見つける Anchor と，Anchor を含む固有表現の範囲を決定する Region の 2 つのネットワークからなる Anchor-Region Network を提案した [7]．Tan et al. は，固有表現の範囲と固有表現の種類を推定するそれぞれのネットワークを組み合わせて学習する手法を提案した [13]．Wan et al. は，GCN (Graph Convolutional Network) を用い，学習データ中で対象の文と類似した特徴を持つ文の情報を NER 対象対象の文の情報と合わせてネットワークに入力する手法を提案した [14]．このような系列ラベリングを用いずに固有表現の範囲と種類を推定していく手法は Span-based 手法と呼ばれ，Nested NER の一つの主流となっている．

Span-based ではない Nested NER の手法も提案されている．Ju et al. は系列ラベリングを多段構造にして用いることで，入れ子構造の固有表現を抽出する手法を提案した [6]．Katiyar et al. は単語ごとの固有表現らしさのスコアを用いたハイパーグラフを用いた手法を提案した [15]．

3 提案手法

本稿では，Span-based の NER 手法を用いる．Span-based の固有表現抽出手法の一般的な手法について述べた後に，提案する ⟨CLS⟩ ベクトルを用いた提案手法について述べる．

3.1 Span-based 固有表現抽出

Span-based の NER 手法では，固有表現の範囲を表す Span と，その固有表現の種類を同時に判定することで固有表現を抽出する．Span 推定と固有表現の種類判定を別のネットワークを用いて行う手法 [7, 13] と，入力分から連続する単語のあらゆるパターンを作成してモデルに入力し，入力 Span が固有表現であるかの判定と固有表現である場合はその種類の判定を一つのネットワークで行う手法 [14] が提案されているが，本稿では後者を用いる．

Span-based 手法による NER では，まず入力文の各単語を事前学習モデルなどを用いてベクトル化する．次に，入力文のうちの連続する複数単語からなる Span を全パターン作成し，その Span に含まれる単語のベクトルをまとめて Span 全体を表すベクトルに変換し，そのベクトルを FFNN (Feed-forward

neural network) などに入力し，Span が固有表現であるか，また固有表現である場合にはその種類を判定する．ベクトルのまとめ上げには Max Pooling や LSTM などが用いられることが多い．

3.2 ⟨CLS⟩ と Transformer を用いたベクトルまとめ上げ

提案手法では，Max Pooling や LSTM の代わりに，BERT[8] で用いられる ⟨CLS⟩ と同様の考え方を用いて 1 次元のベクトルに変換する．提案手法の概要を図 3 に示す．BERT における ⟨CLS⟩ は特別な Token で，⟨CLS⟩ に対応する出力ベクトルは文全体を分類する際に用いられる．BERT 内の Transformer ネットワークにより ⟨CLS⟩ に続く入力文の全単語のベクトルが考慮された，文全体を表すベクトルとなる．この考え方を応用し，提案手法では事前学習モデルから出力された Token ごとのベクトルのうち，分類対象の Span に含まれるものすべてと，⟨CLS⟩ をベクトル空間に埋め込んだものを Transformer に入力する．Transformer の入力ベクトルは $v_{\text{in}} \in \mathbb{R}^{(|t|+1)\times s}$ となる．ここで，$|t|$ は分類対象の Span に含まれる単語の Token 数を，s はベクトルの次元数を表す．v_{in} を Transformer encoder に入力し，その出力から ⟨CLS⟩ に対応するベクトル v_{span} を取り出し，このベクトルを Span 全体を表すベクトルとして利用する．学習の過程で，⟨CLS⟩ を特徴空間に埋め込んだベクトルは，Transformer 内の Multi-head self attention で「固有表現の抽出に役立つベクトル要素」に高い重みが与えられる query となることが期待できる．これにより，Transformer encoder の出力 v_{span} は Span 内の各 Token を表すベクトルを，固有表現抽出に役立つように重みつき和を要素としたベクトルが出力される．

3.3 固有表現の抽出

学習したモデルを用いて固有表現を抽出する方法について述べる．今回対象としている Flat-NER では，入力の各単語が最大でも 1 つの固有表現にのみ属するものとして扱う．一方で，提案手法では各単語が複数の固有表現に属することが可能となっているため，出力されたスコアが最大となる固有表現を出力し，Flat-NER の手法として評価を行う．

学習したモデルを用い，入力文から作成した全 Span に対し，固有表現の種類，または固有表現ではない場合のスコアを出力する．「固有表現ではない」スコアが最大の場合にはこの Span を棄却し，それ以外の場合にはスコアが最大となる固有表現の種類とともにスコアを出力し，これをスコアの降順に並べる．スコアが最大のものから順に，Span に属する単語に固有表現のラベルを割り当てていく．この際に，既により高いスコアの Span によりラベル付けされた単語が 1 つでも含まれている場合には，その Span は棄却する．すべての Span について割り当てが完了した時点で，ラベルが付けられていない単語は固有表現ではないとラベルをつけ，これを最終的な固有表現抽出結果として出力する．

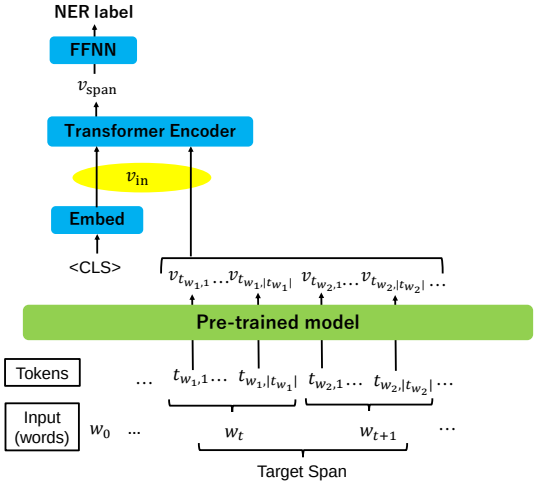

図 3 提案手法の概要．

表 1 データセットの諸元

データ	学習	開発	評価	Class 数	データ種類
WNUT-16	2,394	1,000	3,849	10	SNS
WNUT-17	3,394	1,009	1,287	6	SNS
CoNLL 2003	14,987	3,466	3,684	4	ニュース

4 評価実験

4.1 使用データ

評価実験には，WNUT-16[16]，WNUT-17[17]，CoNLL 2003[18] の 3 つのデータセットを用いた．それぞれのデータ量等を表 1 に示す．

CoNLL 2003 はニュース原稿を基にしたデータであるため，未知の固有表現が少ない一方，WNUT-16 と WNUT-17 は SNS のデータであるため未知の固有表現が多い．特に，WNUT-17 では評価データに出現する固有表現はすべて学習データに出現しない未知のものになるように設計されている．

表 2　実験結果

	WNUT-16	WNUT-17	CoNLL 2003
Transformer	60.05	**60.93**	93.15
LSTM	59.79	60.39	93.11
Max-pooling	**60.09**	60.81	92.99
Avg-pooling	58.31	56.83	93.03
SoTA	59.50	60.45	**94.60**

4.2　実験設定

提案手法の実装には PyTorch と Transformers を用い, RAdam[19] によりモデルを学習した．事前学習モデルとして，XLM-RoBERTa-large[20] を使用した．学習率は事前学習モデル部の追加学習に 1.0×10^{-5}，それ以外の部分を 2.0×10^{-4} とした．学習時のミニバッチサイズを 50，学習エポック数を 50 とし，開発データの micro-F1 が最大となるモデルを最終的なモデルとして評価に用いた．学習はそれぞれのモデルについてランダムシードを変えて 3 回行い，その中央値を報告する．

提案手法で用いる Transformer encoder は，ベースライン手法の Max Pooling などと条件を合わせるために 1 層とした．また，FFNN は 3 層とした．FFNN の各層の間では Layer Normalization と Relu を，また学習時には Dropout を用いた．Dropout 率は 0.3 とした．

4.3　ベースライン手法

ベースライン手法として，ベクトルのまとめ上げ部分に Transformer を用いずに別の手法を用いたものを用意し，性能を比較した．

LSTM　対象単語列の各 Token のベクトルを双方向 LSTM に入力し，すべて入力後の隠れ層のベクトルを FFNN に入力して判定する．このベースライン手法では，LSTM を 1 層とした．

Max pooling　対象単語列の各 Token のベクトルを Max pooling により単一のベクトルに変換し，FFNN に入力して判定する．

Average pooling　対象単語列の各 Token のベクトルを Average pooling により単一のベクトルに変換し，FFNN に入力して判定する．

4.4　実験結果

実験結果と，2023 年 12 月時点の SoTA の性能を表 2 に示す．SoTA はすべて異なる論文で報告されたもので，WNUT-16 が Hu et al.[21], WNUT-17 が Wang et al.[12], CoNLL 2003 が Wang et al.[22] により報告されたものである．

提案手法は，WNUT-16 の Max pooling を除くベースライン手法より高い性能を得られた．WNUT-16 と WNUT-17 では現在の State-of-the-art の手法をも上回る性能となった．これにより，提案手法の有効性を確認することができた．

5　考察

提案手法では，Transformer encoder を用いて分類対象となる単語列の全 Token のベクトルをまとめ上げて 1 次元ベクトルに変換した．この計算過程で，Transformer 内部の Multi-head attention により，各 Token のベクトルの重み付け和に似た性質のベクトルが ⟨CLS⟩ の出力に足されている．これが単純に入力ベクトルの最大値を取る Max Pooling や平均値を取る Average Pooling よりもより精度良い足し合わせとなり，性能が向上したものと考えられる．また，LSTM などの RNN (Recurrent Neural Network) モデルは，入力を順番に処理していくため，初期の入力の情報が忘却されやすく，最後の入力に近いものほど出力への影響が大きいことが知られている．提案手法では入力の Span 内での語順を考慮しない代わりに，LSTM のような語順による影響の大きさの違いが生じないことが性能の向上に寄与したものと考えられる．

今回は Max/Average pooling と条件を合わせるために，提案手法や LSTM でも 1 層のネットワークで実験したが，提案手法では Pooling 等と異なり層を増やすことでき，より複雑なタスクなどでも応用がしやすいことも利点の一つでもあると考えられる．

6　おわりに

本稿では，Span-based 手法による固有表現抽出手法について述べた．複数 Token からなる分類対象の単語列のベクトルを，BERT 等で用いられる ⟨CLS⟩ と同様の考え方により 1 次元に変換し，その変換したベクトルを用いて固有表現の種類を分類する手法を提案した．評価実験において，ベースライン手法と比較して良好な性能を得られることを確認した．特に WNUT-17 データセットにおいては，現在の State-of-the-art を上回る性能を得ることができた．

今回は Transformer や LSTM について，1 層のモデルを用いた．課題として，特に CoNLL 2003 のようにデータサイズが大きい場合が挙げられ，今後，層数の性能への影響を実験により確認していく．

参考文献

[1] Aman Kumar and Binil Starly. “fabner” : information extraction from manufacturing process science domain literature using named entity recognition. **Journal of Intelligent Manufacturing**, Vol. 33, No. 8, pp. 2393–2407, 2022.

[2] Ondřej Sotolář, Jaromír Plhák, and David Šmahel. Towards personal data anonymization for social messaging. In **International Conference on Text, Speech, and Dialogue**, pp. 281–292. Springer, 2021.

[3] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. **arXiv preprint arXiv:1508.01991**, 2015.

[4] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In **Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 260–270, San Diego, California, June 2016. Association for Computational Linguistics.

[5] Jenny Rose Finkel and Christopher D. Manning. Nested named entity recognition. In **Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing**, pp. 141–150, Singapore, August 2009. Association for Computational Linguistics.

[6] Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. A neural layered model for nested named entity recognition. In **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 1446–1459, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[7] Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5182–5192, Florence, Italy, July 2019. Association for Computational Linguistics.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In **Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[9] Casimir Borkowski and Thomas J. Watson. An experimental system for automatic recognition of personal titles and personal names in newspaper texts. In **COLING 1967 Volume 1: Conference Internationale Sur Le Traitement Automatique Des Langues**, 1967.

[10] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In **Proceedings of the 27th International Conference on Computational Linguistics**, pp. 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[11] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. Luke: Deep contextualized entity representations with entity-aware self-attention. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 6442–6454, 2020.

[12] Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Improving Named Entity Recognition by External Context Retrieving and Cooperative Learning. In **the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)**. Association for Computational Linguistics, August 2021.

[13] Chuanqi Tan, Wei Qiu, Mosha Chen, Rui Wang, and Fei Huang. Boundary enhanced neural span classification for nested named entity recognition. In **Proceedings of the AAAI conference on artificial intelligence**, Vol. 34, pp. 9016–9023, 2020.

[14] Juncheng Wan, Dongyu Ru, Weinan Zhang, and Yong Yu. Nested named entity recognition with span-level graphs. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 892–903, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[15] Arzoo Katiyar and Claire Cardie. Nested named entity recognition revisited. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 861–871, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[16] Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. Results of the WNUT16 named entity recognition shared task. In Bo Han, Alan Ritter, Leon Derczynski, Wei Xu, and Tim Baldwin, editors, **Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)**, pp. 138–144, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[17] Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In **Proceedings of the 3rd Workshop on Noisy User-generated Text**, pp. 140–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[18] 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 at HLT-NAACL 2003**, pp. 142–147, 2003.

[19] Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In **Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)**, April 2020.

[20] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. **CoRR**, Vol. abs/1911.02116, , 2019.

[21] Jinpeng Hu, Yaling Shen, Yang Liu, Xiang Wan, and Tsung-Hui Chang. Hero-gang neural model for named entity recognition. In **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 1924–1936, Seattle, United States, July 2022. Association for Computational Linguistics.

[22] Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. Automated concatenation of embeddings for structured prediction. In **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)**, pp. 2643–2660, Online, August 2021. Association for Computational Linguistics.