# ラベル内容のエンコードとラベル間の制約に基づく補助コーパスを用いた固有表現抽出

大井 拓　三輪 誠　佐々木 裕
豊田工業大学
{sd19013,makoto-miwa,yutaka.sasaki}@toyota-ti.ac.jp

## 概要

自然言語処理分野における教師あり深層学習では，高い性能を達成するためには大量のラベル付きデータを含むコーパスが必要である．ラベル付きデータを増やす方法として，補助コーパスとしての既存のラベル付きコーパスの利用が考えられるが，コーパスごとにアノテーションの基準が異なるため，単純に補助コーパスを追加して学習しても性能が向上するとは限らない．本研究では，固有表現抽出を対象に，コーパス情報や説明文を含むラベルの情報を埋め込み表現にエンコードし，異なるコーパスのラベル間の違いを考慮しながら，補助コーパスを対象コーパスに追加して学習を行う新たなスパンベースのモデルを提案する．生物医学分野の 2 つの固有表現抽出コーパスについて，BC5CDR を対象コーパス，NCBI を補助コーパスとして，本提案手法を評価した結果，補助コーパスにより対象コーパスでの性能を向上できることがわかった．

## 1　はじめに

文章の内容を捉えるにはその話題の中心となる固有表現の把握が重要であり，固有表現抽出 (Named Entity Recognition; NER) は重要な基礎タスクとして古くから取り組まれている．

近年，自然言語処理の分野では，正解がラベル付けされた訓練データからなるラベル付きコーパスを用いてモデルを学習する教師あり深層学習を用いた手法 [1, 2] が主流となっている．教師あり深層学習においては，大量の訓練データが必要であるが，その人手でのラベル付けには高いコストがかかる．

訓練データを増やす方法として，補助コーパスとして既存のラベル付けされたコーパスを利用することが考えられる．しかし，NER のコーパスはラベルの種類の違い，アノテーションガイドラインにおけるラベル付け基準の違いによるラベルが付いている用語の違い，というコーパスごとのラベル付けの違いがあるため，そのまま同様のデータとして扱うことが難しい．例えば，生物医学分野のコーパスである BC5CDR (BioCreative V CDR task corpus) [3] と NCBI (NCBI Disease corpus) [4] における病名に対するアノテーションの違いを図 1 に示した．この 2 つのコーパスはともに “pain” がテキスト上に現れており，BC5CDR では病名としてラベル付けされているが，NCBI ではラベル付けされていない．これは一般的な用語に対してはラベル付けを行わない NCBI に対して，BC5CDR では一般的な用語の中でも “pain” や “cancer” などにはラベル付けを行うと基準を定めているためである．

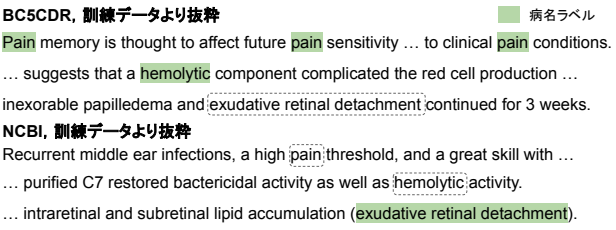


**図 1**　BC5CDR と NCBI の両方に含まれる事例でラベル付けに違いがある例

本研究では，複数コーパスにおけるラベル付けの違いによる影響の緩和により，対象コーパスにそのコーパス以外のコーパス（補助コーパス）を追加して学習することで，NER モデルの性能向上を目指す．そのためにコーパス情報を与えたラベル表現による新たなスパンベースの固有表現抽出モデルを提案する．実験では，対象コーパスに補助コーパスを追加した場合の性能の変化を評価し，提案手法の有効性を検証する．本研究の貢献は次の通りである．

- 補助コーパスを対象コーパスに追加して学習を行うスパンベースの固有表現抽出手法の提案
- 複数コーパスのラベルを区別するラベルの内容を表すためのラベル表現の有効性の確認

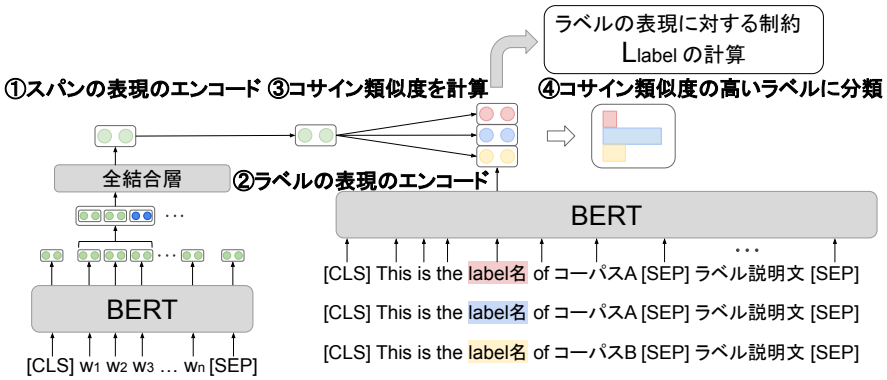


**図 2**　提案モデルの全体像

- 複数コーパスのラベル表現に意味的な近さにより制約をかけることの有効性の確認

# 2　関連研究

## 2.1　スパン表現による固有表現抽出

文中の一定範囲の領域であるスパンを対象としたスパン表現を用いた NER が近年注目を浴びている [2, 5, 6]．Zhong ら [2] のモデルでは，文のエンコードの部分で BERT (Bidirectional Encoder Representations from Transformers) [1] における注意機構による文脈を含んだ表現を用いて，対象スパンの先頭，末尾の単語の埋め込み，長さの埋め込みを連結し，スパンを表現している．このスパンの表現を全結合層に入力し，分類ラベルの次元数に落とし各次元をラベルに割り当てることで分類を行う．Zhong らは単純なモデルでありながら，end-to-end で NER と関係抽出を解くタスクで最高性能を示し，NER でも高い性能を示すスパン表現によるモデルを提案した．

## 2.2　補助コーパスとラベル表現を用いた固有表現抽出

分類の対象となるラベル自体の意味内容も分類を行う上で重要であるという考えから，Ma ら [7] は，トークンベースの固有表現について，BERT によって埋め込み表現ベクトルへ変換した分類ラベルを用いて固有表現抽出を行う手法を提案した．具体的には，それぞれのトークンの BERT による埋め込み表現ベクトルと分類の対象となるすべてのラベル表現のベクトルで内積を計算し，そのトークンを値の高いラベルに分類する．この手法によって，似たラベルを扱っている補助コーパスで BERT をファインチューニングすることで対象コーパスの訓練データを各ラベルの事例数を 1，5，20，50 件に制限した条件で高い性能を実現した．一方で，コーパスごとのラベル付けの違いは考慮しておらず，目的とするコーパスの訓練データの事例数を制限しない場合，性能の向上は見られなかった．

# 3　提案手法

本研究では，補助コーパスを用いた異なるコーパスのラベルとその意味関係を考慮したスパン表現による固有表現抽出モデルを提案する．提案モデルの全体像を図 2 に示す．2.2 節の Ma ら [7] とは異なり，スパン表現のモデルを用いることでトークンレベルではなく，固有表現レベルでのラベルの表現を利用する．まず，異なるコーパスのラベルを区別するために，コーパス名の情報やラベルの説明文をラベルに付加した上で，BERT を用いてラベルを埋め込み表現ベクトルへエンコードする．また，学習中にラベルの表現を調整する制約を加えることで，コーパス間の近いラベルと異なるラベルを考慮したラベルの表現を目指す．

## 3.1　ラベル表現によるスパン分類モデル

2.1 節の Zhong らのモデルと同様に予測対象となる文を事前学習済みの BERT によってエンコードし，その表現から単語を組み合わせたスパンの表現を作成する．BERT の出力の表現ベクトルをスパンの先頭と末尾で結合し，そこにスパンの長さの埋め込み表現を結合したものを全結合ニューラルネットワークによって 768 次元にしたものをスパンの表現とする．$\boldsymbol{x}_1$ から $\boldsymbol{x}_n$ のスパンの表現を以下に示す．

$$\boldsymbol{h}_{span} = \mathrm{Linear}(\mathrm{Concat}(\boldsymbol{x}_1, \boldsymbol{x}_n, \Phi(n))) \qquad (1)$$

ここで，Linear(·) はスパンの次元から 768 次元にする全結合層，Concat(·) はベクトルの結合，$\Phi(n)$ はスパンの長さ $n$ に対する埋め込み表現を表す．

コーパスの違いやラベルの意味内容を含んだ分類を行うためにラベルに対してもエンコードを行う．文のエンコードと同様に事前学習済みのBERT によってラベル表現を得る[1)]．具体的には，複数のコーパスとそのそれぞれの分類ラベルの名前を “[CLS] This is the <ラベル名> of <コーパス名> [SEP] ラベル説明文 [SEP]” のように入力することでラベルにコーパスの情報を加える．ラベル表現によって分類を行うため，負例についても，それに対応する “others” ラベルの表現を作る．ラベル表現には入力の<ラベル名>のトークンに対応する BERT の出力を利用する．

作成したスパンの表現とラベル表現によってスパンの分類を行う．スパンの表現とラベル表現は同じ次元数のベクトルであるため，スパンの表現と各ラベル表現のコサイン類似度を計算し，そのスパンの各ラベルに対するスコアとし，スコアが最も高いラベルに分類する．さらに，正解ラベル $y$ について，NER の損失 $L_{CE}$ を次のように計算する．

$$\hat{y} = \mathrm{Softmax}(\alpha \cos(\boldsymbol{h}_{span}, \boldsymbol{h}_{label}))$$

$$\cos(\boldsymbol{h}_{span}, \boldsymbol{h}_{label}) = \frac{\boldsymbol{h}_{span} \cdot \boldsymbol{h}_{label}}{\|\boldsymbol{h}_{span}\| \|\boldsymbol{h}_{label}\|} \tag{2}$$

$$L_{CE} = -\sum_{\boldsymbol{h}_{span} \in S} \sum_{i} y_i \log \hat{y}_i \tag{3}$$

ここで，$\alpha$ は学習を進めるためのハイパーパラメータである．

### 3.2 ラベル表現に対する制約を考慮した学習

学習では，NER の分類の損失に，ラベル間のコサイン類似度に対する制約としてラベル間の関係を表現した損失を加えて学習を行う．

$$L = L_{CE} + L_{label} \tag{4}$$

ここで，$L_{label}$ は意味的に近いラベル表現が近づき，遠いラベル表現が遠くなるように以下のように定義する．

$$\begin{aligned} \boldsymbol{L}_{label} = &\sum_{\mathbb{H}_{near}} \max\{0, \beta - \cos(\boldsymbol{h}_{label}, \boldsymbol{h}'_{label})\} \\ &+ \sum_{\mathbb{H}_{far}} \max\{0, \cos(\boldsymbol{h}_{label}, \boldsymbol{h}'_{label}) - \beta\} \end{aligned} \tag{5}$$

$\beta$ は，ハイパーパラメータであり，ラベル表現間のコサイン類似度の閾値を表す．また，$(\boldsymbol{h}_{label}, \boldsymbol{h}'_{label}) \in \mathbb{H}_{near}$ は近づけるラベル表現の組み合わせの集合，$(\boldsymbol{h}_{label}, \boldsymbol{h}'_{label}) \in \mathbb{H}_{far}$ は遠ざけるラベル表現の組み合わせの集合である．

学習においては，モデルを対象コーパスに特化させるために，1 エポック中にまず補助コーパスの訓練データによる学習を行い，次に対象コーパスの訓練データによる学習を行う．学習時には異なるコーパスのラベルへの分類を防ぐため，対象としている文が属するコーパスのラベルのみでの分類を行う．

1) 2 つの BERT はパラメータは共有しない．

**表 1** コーパスのラベルごとの事例数

| コーパス | ラベル | 訓練 | テスト |
|---|---|---|---|
| BC5CDR | Disease | 4,182 | 4,424 |
| | Chemical | 5,203 | 5,385 |
| NCBI | Disease | 5,145 | 960 |

**表 2** ラベル表現間のユークリッド距離．括弧内はコーパス，赤字は表現を近づける組み合わせ $\mathbb{H}_{near}$ を，黒字は表現を遠ざける組み合わせ $\mathbb{H}_{far}$ を表す．$-L_{label}$ は 3.2 節のラベル表現による損失を利用しないモデル．Dis は病名，Chem は薬物，O は負例を表す．

| | | Proposed | $-L_{label}$ |
|---|---|---|---|
| O (BC5) | Dis (BC5) | 38.9 | 44.4 |
| O (BC5) | Chem (BC5) | 30.2 | 46.5 |
| O (BC5) | O (NCBI) | 18.3 | 17.8 |
| O (BC5) | Dis (NCBI) | 40.0 | 45.6 |
| Dis (BC5) | Chem (BC5) | 35.8 | 35.7 |
| Dis (BC5) | O (NCBI) | 37.3 | 44.2 |
| Dis (BC5) | Dis (NCBI) | 19.3 | 19.4 |
| Chem (BC5) | O (NCBI) | 22.7 | 45.7 |
| Chem (BC5) | Dis (NCBI) | 31.8 | 35.1 |
| O (NCBI) | Dis (NCBI) | 36.3 | 44.9 |

## 4 実験設定

複数コーパスでの学習における提案手法の有効性を検証するために，BC5CDR [3] を対象コーパスと NCBI [4] を補助コーパスとして評価を行った．各コーパスのラベルとデータの統計を表 1 に示す．

実験では，生物医学文書による事前学習済みである PubMedBERT [8] をエンコーダとして採用する．スパンの表現に用いるエンコーダのパラメータは 1 エポックごとの更新，ラベル表現に用いるエンコーダのパラメータは 1 エポックおきの更新として訓練データでのファインチューニングを行う．

### 4.1 比較モデル

提案手法の有効性を検証するためにモデル構造及び入力を変更した以下の手法を比較する．

**表 3** ラベル表現による分類結果．値は F 値 [%]．最も高いスコアを太字とした．

| | BC5CDR | +NCBI |
|---|---|---|
| Linear | 89.11 | 89.06 |
| Cosine | 89.22 | 89.05 |
| Proposed | 89.12 | **89.25** |

**表 4** ラベルの入力による影響 [%]．Abstract はラベルの入力のラベル説明文を各コーパスの論文の要旨に，Label はラベルのみを入力に変更して学習したモデル．最も高いスコアを太字とした．

| | P | R | F |
|---|---|---|---|
| Proposed | 89.20 | 89.30 | **89.25** |
| Abstract | **89.64** | 88.69 | 89.16 |
| Label | 88.39 | **89.83** | 89.10 |

- **Linear** Zhong ら [2] の手法．エンコーダには提案手法と同様に PubMedBERT を用い，提案手法と条件を合わせるために全結合層を 2 層とし，間の隠れ層は 768 次元とした．学習に 2 つのコーパスの訓練データを用いる場合は，重複するラベルを同じラベルとした．
- **Cosine** ラベルのエンコードを行わずラベル数のベクトルを学習可能なパラメータとして与え，全結合層ではなくスパンの表現とのコサイン類似度を計算し分類する手法．
- **Proposed** 提案手法．ラベル入力のラベル説明文としてコーパス発表時の論文中のアノテーションガイドラインの章を採用した．3.2 節の $L_{label}$ における閾値 $\beta$ は 0.6 とした．制約をかけたラベルの組み合わせについては，表 2 に $\mathbb{H}_{near}$ に含まれる組み合わせを赤字で，$\mathbb{H}_{far}$ に含まれる組み合わせを黒字で示した．

# 5 結果と考察

## 5.1 ベースラインモデルとの比較

まず，ベースラインモデルと提案手法の比較として Linear，Cosine，Proposed の BC5CDR テストデータでの評価結果を表 3 に示す．各モデルを BC5CDR のみで学習した場合と NCBI を補助コーパスとして追加して学習した場合のモデルで評価を行った．

Linear と Cosine では BC5CDR の訓練データのみで学習を行った場合に比べて，NCBI を追加した場合，性能が低くなっている．対して，提案手法である Proposed では NCBI を追加した場合に性能が高い結果となった．このことから，提案手法のラベルの入力とラベルの表現に対する制約が複数のコーパスにおけるラベルの違いに対して有効であるとわかった．

**表 5** ラベル間の制約の損失の影響 [%]．最も高いスコアを太字とした．

| | P | R | F |
|---|---|---|---|
| Proposed | 89.20 | **89.30** | **89.25** |
| $-L_{label}$ | **89.31** | 88.37 | 88.83 |

## 5.2 ラベル情報による影響

次に，ラベル情報の追加による影響を検証するために，ラベルの BERT の入力を変更して評価を行なった結果を表 4 に示す．ラベルの内容を表す情報を追加することで性能が向上していることがわかる．

## 5.3 ラベル表現での損失関数による影響

最後に，提案手法の制約の損失 $L_{label}$ の影響についての評価を表 5 に，各ラベル表現間のユークリッド距離を表 2 に示す．3.2 節で定義した $L_{label}$ によって，特に BC5CDR の Chemical と NCBI の Others のラベルについて，その距離が大きく変化していることがわかる．

# 6 おわりに

本研究では対象コーパスからの固有表現抽出に補助コーパスを追加して学習するスパンベースの固有表現抽出手法を提案した．複数コーパスのラベルの違いによる影響を緩和するために，ラベルの内容に関するコーパス情報を含ませてラベルをエンコードし，ラベル表現間の制約を損失として導入した．BC5CDR を対象コーパス，NCBI を補助コーパスとした評価においては，補助コーパスを用いた提案手法により，対象コーパスでの性能を向上できることを確認した．

今後の課題としては，対象コーパスと補助コーパスを区別せず，コーパスを同時に用いることで両コーパスでの性能向上が可能なモデルを目指す．また，ラベルの入力やラベル表現に対する制約によって性能が変化したことから，ラベル表現の関係の改善によるさらなる性能向上を目指す．さらに，学習に用いるコーパスを増やした場合の影響を検証したい．

## 謝辞

本研究は JSPS 科研費 JP20K11962 の助成を受けたものです.

## 参考文献

[1] 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.

[2] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In **Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 50–61, Online, June 2021. Association for Computational Linguistics.

[3] Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. **Database**, Vol. 2016, , 05 2016. baw068.

[4] Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. Ncbi disease corpus: A resource for disease name recognition and concept normalization. **Journal of Biomedical Informatics**, Vol. 47, pp. 1–10, 2014.

[5] Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing**, pp. 188–197, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[6] Mohammad Golam Sohrab and Makoto Miwa. Deep exhaustive model for nested named entity recognition. In **Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing**, pp. 2843–2849, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.

[7] Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. Label semantics for few shot named entity recognition. In **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 1956–1971, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[8] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. **ACM Trans. Comput. Healthcare**, Vol. 3, No. 1, oct 2021.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, **Advances in Neural Information Processing Systems**, Vol. 32. Curran Associates, Inc., 2019.

[10] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**, pp. 38–45, Online, October 2020. Association for Computational Linguistics.

## A　ハイパーパラメータ

本研究における実験で用いたハイパーパラメータを表 6 に示す．スパンを作成する BERT，ラベルを作成する BERT，それ以外の全結合ニューラルネットワークなどのパラメータで異なる学習率を用いている．

**表 6**　各モデルのハイパーパラメータ

| | Linear | Cosine | Proposed |
|---|---|---|---|
| 学習率（スパン） | 4e-5 | 4e-5 | 4e-5 |
| 学習率（ラベル） | - | - | 1e-5 |
| 学習率（その他） | 6e-5 | 8e-5 | 8e-5 |
| $\alpha$ | - | 20 | 20 |
| $\beta$ | - | - | 0.6 |
| エポック数 | 100 | 100 | 100 |
| batch size | 48 | 48 | 48 |

## B　実験環境

実験のためのプログラミング言語は Python 3.7.11，機械学習ライブラリとして PyTorch [9] のバージョン 1.11.0，事前学習モデルを使用するために Transformers [10] のバージョン 3.0.2 を用いた．また，計算機では CPU に Intel(R) Xeon(R) CPU E5-2698 v4 及び Intel(R) Xeon(R) W-3225，GPU に NVIDIA Tesla V100 DGXS 32GB 及び NVIDIA RTX A6000 を用いた．