Vol.2021-NL-249 No.4
2021/7/27

複数の補助教師データセットを用いた固有表現抽出の学習手法

市川　智也[1,a)]　渡邊　大貴[2,b)]　田村　晃裕[1,c)]　岩倉　友哉[2,d)]　馬　春鵬[2,e)]　加藤　恒夫[1,f)]

概要：固有表現抽出（Named Entity Recognition; NER）は，テキストからの知識獲得に用いる要素技術の一つであり，化学物質や医療の知識抽出に用いられている．NER の精度改善のため，対象タスクの教師データとは別の教師データを補助教師データとして用いるマルチタスク学習である補助学習が提案されている．従来の補助学習では補助教師データとして 1 種類の教師データしか用いていない．そこで，本研究では，複数種類の教師データを補助教師データとして活用する NER の学習手法を提案する．具体的には，補助教師データ毎の補助学習を順次行うことで，対象タスクのモデルを補助教師データの種類の数だけファインチューニングする方法と，全種類の教師データを一つの補助学習で用いる方法の 2 種類の学習手法を提案する．評価実験では，化学物質名抽出タスクにおいて，7 種類の化学/科学技術分野の補助教師データを用いて提案手法の評価を行った．その結果，提案手法は従来手法よりも精度が高く，複数の補助教師データを用いることで精度が向上することを確認した．

Learning Method for Named Entity Recognition with Multiple Auxiliary Training Data

1. はじめに

固有表現抽出（Named Entity Recognition; NER）は，文中から固有表現（Named Entity; NE）や専門用語を抽出する自然言語処理技術の一つであり，様々な場面で用いられている．たとえば，新材料や新薬の開発，材料を用いた製品開発には化学物質に関する知識が必要不可欠であり，NER は，論文や特許で日々報告される化学物質間の相互関係や化学物質の物性値といった情報を構造化し蓄積するための要素技術の一つとして用いられている．

NER に関する研究は古くから盛んに行われている．近年では，ニューラルネットワーク（Neural Network; NN）による手法が主流となっており，再帰的ニューラルネットワーク（Recurrent Neural Network; RNN）と条件付確率場（Conditional Random Fields; CRF）を組み合わせた BiLSTM-CRF モデルによる手法 [1] や Transformer[2] による手法が，NER において高い精度を実現している．また，近年では，対象タスクの教師データとは別の教師データも用いるマルチタスク学習により，複数の NER データセットから特徴量を同時に学習することでモデルの精度が改善することが報告されている [3], [4], [5], [6], [7].

特に，バイオ分野の NER（BioNER）においては，複数のタスクを同時に学習するマルチタスク学習と比較し，対象タスク以外のタスクを補助タスクとして用いるマルチタスク学習である補助学習を行うことで目的タスクにおいて高い精度を示すことが報告されている [3].

本研究では，先行研究の補助学習が 1 種類の補助教師データしか用いなかったのに対し，複数の NER データセットを補助教師データとして用いる手法を提案する．具体的には，補助教師データ毎の補助学習を順次行うことで，対象タスクのモデルを補助教師データの種類の数だけファインチューニングする方法と，全種類の教師データを一つの補助学習で用いる方法の 2 種類の学習手法を提案する．BioCreative IV's CHEMDNER タスクで評価した結果，7

[1] 同志社大学
Kyotanabe, Kyoto 610–0394, Japan
[2] 富士通株式会社
Minato, Tokyo 105–7123, Japan
[a)] ctwg0109@mail4.doshisha.ac.jp
[b)] watanabe-taiki@fujitsu.com
[c)] aktamura@mail.doshisha.ac.jp
[d)] iwakura.tomoya@fujitsu.com
[e)] ma.chunpeng@fujitsu.com
[f)] tsukato@mail.doshisha.ac.jp

ⓒ 2021 Information Processing Society of Japan

種類の補助教師データを用いることで，提案手法は従来手法に比べて，F 値が向上することを確認した．また，提案手法において複数の補助教師データを利用する際，対象タスクの性能改善に寄与する度合いの小さい補助教師データから順に用いることで精度が向上することを確認した．

2. 従来手法

本節では，対象タスクの教師データ以外の教師データも用いる従来のマルチタスク学習手法を説明する．2.1 節では，本研究で用いる NER モデルを概説する．そして，その NER モデルを複数のタスクで同時に学習するマルチタスク学習に拡張したモデルについて述べる．このマルチタスク学習では，対象タスクとその他のタスクは同等に扱われる．2.2 節では，対象タスクではないタスクの教師データを補助教師データとして用いる補助学習を説明する．

2.1 マルチタスクモデル

本研究では，Huang ら [1] により提案された BiLSTM-CRF モデルを NER モデルとして使用する．BiLSTM-CRF モデルは，双方向 LSTM と CRF を用いた系列ラベリングモデルである．

BiLSTM-CRF モデルは，まず，双方向 LSTM により，入力文中の各単語の中間表現を算出する．入力文を $\mathbf{w} = w_1, w_2, \cdots, w_N$，各単語 w_i を埋め込み層でベクトル化した結果を $\mathbf{x} = \mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_N}$ とすると，単語 w_i の中間表現 $\mathbf{e_i}$ を以下のように算出する．

$$\overrightarrow{\mathbf{h_i}} = LSTM^{(f)}(\mathbf{x_i}, \overrightarrow{\mathbf{h_{i-1}}}) \tag{1}$$

$$\overleftarrow{\mathbf{h_i}} = LSTM^{(b)}(\mathbf{x_i}, \overleftarrow{\mathbf{h_{i+1}}}) \tag{2}$$

$$\mathbf{h_i} = [\overrightarrow{\mathbf{h_i}}; \overleftarrow{\mathbf{h_i}}] \tag{3}$$

$$\mathbf{e_i} = \mathbf{W_e}\mathbf{h_i} \tag{4}$$

ここで，$\rightarrow$ と $\leftarrow$ は，それぞれ，順方向と逆方向を表し，$LSTM^{(f)}$ と $LSTM^{(b)}$ は，それぞれ，順方向と逆方向の LSTM を表す．また，; はベクトルの結合を表す．$\mathbf{W_e} \in R^{k \times d}$ は重み行列であり，d は隠れ状態ベクトル $\mathbf{h_i}$ の次元数，k は識別対象のラベルの数である．

その後，双方向 LSTM により算出された $\mathbf{e_i}$ を CRF に入力し，ラベル系列を求める．まず，双方向 LSTM の出力系列 $\mathbf{e} = (\mathbf{e_1}, \mathbf{e_2}, \cdots, \mathbf{e_N})$ をスコア行列に変換した $\mathbf{P} = (\mathbf{e_1}, \mathbf{e_2}, \cdots \mathbf{e_N})^T$ と，遷移スコア行列 $\mathbf{A}$ を用いて，ラベル系列 $\mathbf{y} = (y_1, y_2, \cdots, y_N)$ に対するスコア関数を次のように定義する．

$$\mathrm{s}(\mathbf{e}, \mathbf{y}) = \sum_{i=0}^{N} A_{y_i, y_{i+1}} + \sum_{i=1}^{N} P_{i, y_i} \tag{5}$$

ここで，$A_{i,j}$ は i 番目のラベルから j 番目のラベルに遷移するスコアを表す．このスコア関数を用いてラベル系列 $\mathbf{y}$ の出力確率を次のように sotmax 関数により計算する．

$$p(\mathbf{y}|\mathbf{e}) = \frac{e^{\mathrm{s}(\mathbf{e}, \mathbf{y})}}{\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_w}} e^{\mathrm{s}(\mathbf{e}, \tilde{\mathbf{y}})}} \tag{6}$$

ここで，$\mathbf{Y_w}$ は入力文 $\mathbf{w}$ に対するすべての可能なラベル系列である．そして，次式のようにスコアを最大化する $\mathbf{y}$ を求めることで出力ラベル系列 $\mathbf{y}^*$ を獲得する．

$$\mathbf{y}^* = \arg\max_{\tilde{\mathbf{y}} \in \mathbf{Y_w}} \mathrm{s}(\mathbf{e}, \tilde{\mathbf{y}}) \tag{7}$$

このように，BiLSTM-CRF モデルは，ラベリング問題を各単語に対して独立にモデル化するのではなく，系列全体で同時にモデル化する．

学習時には正解ラベル系列を用いて次式を最大化するパラメータを求める．

$$\log(p(\mathbf{y}|\mathbf{e})) = \mathrm{s}(\mathbf{e}, \mathbf{y}) - \log(\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_w}} e^{\mathrm{s}(\mathbf{e}, \tilde{\mathbf{y}})}) \tag{8}$$

図 1 にマルチタスク学習用に拡張した BiLSTM-CRF モデルを示す．単語埋め込み層と BiLSTM 層は全ての教師データで共有し，共通の重みを用いる．一方で，CRF 層はデータセットごとに用意し，CRF 層の重みは共有しない．各データセットの CRF 層における損失を L_i $(i = 1, 2, \cdots M)$ とすると，このマルチタスク学習の目的関数は次式のように定義される．

$$Loss = \frac{1}{M} \sum_{i=1}^{M} L_i \tag{9}$$

ただし，M は教師データセットの種類数である．

このマルチタスクモデルの学習では，対象タスクの教師データとそれ以外のタスクの教師データを同等に扱うことで，全てのタスクで共通の一つのモデルを学習する．推論時には，学習したモデルにおいて目的のタスクに該当する CRF 層を用いて NER を行う．

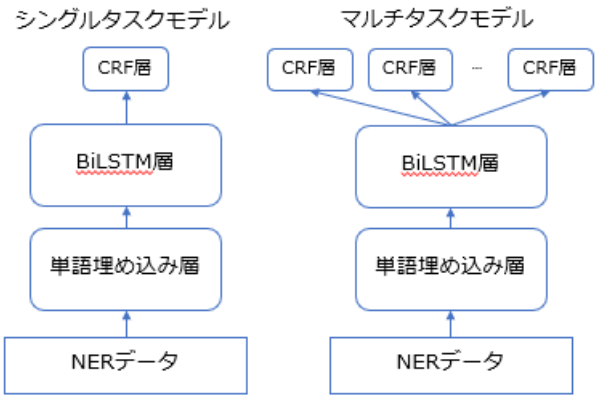

図 1 マルチタスクモデル

2.2 補助モデル

Wang ら [3] は，対象タスクの教師データ（メイン教師データ）とそれ以外の教師データ（補助教師データ）を区

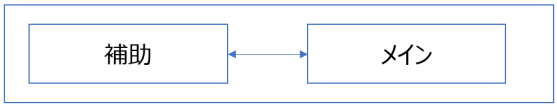

図 2 補助モデル

Algorithm 1: 補助モデルのアルゴリズム

Data: メイン教師データ D_{main}，補助教師データ D_{aux}

1 **begin**
2 　**for** $i = 1, 2, \cdots, Epoch$ **do**
3 　　**for** $j = 1, 2, \cdots, Iteration$ **do**
4 　　　$Batch_{main} = extract(D_{main}, BatchSize)$
5 　　　$Batch_{aux} = extract(D_{aux}, BatchSize)$
6 　　　$train(Model, Batch_{aux})$
7 　　　$train(Model, Batch_{main})$
8 　　$is\_converge(Model)$

別するマルチタスク学習である補助学習を行うことで，対象タスクに対する NER の性能改善を行った．この補助モデルの概要を**図 2** に示す．補助モデルでは，メイン教師データから作成したメインバッチと補助教師データから作成した補助バッチを用いて学習を行う．イテレーション毎に，補助バッチでモデルのパラメータを更新し，その後でメインバッチでモデルのパラメータを更新する．このメイン教師データと補助教師データの交互の学習を，メイン教師データに対する損失が収束するまで繰り返す．

補助学習のアルゴリズムをアルゴリズム 1 に示す．アルゴリズム 1 において，添え字は対象タスク（$main$）と補助タスク（aux）を表す．$Epoch$，$Iteration$ は，それぞれ，メイン教師データに対するエポック数とイテレーション数であり，$BatchSize$ はバッチサイズを表す．各エポックのイテレーション回数は，メイン教師データの総数をバッチサイズで割った値である（$Iteration = |D_{main}|/BatchSize$）．4,5 行目の $extract$ は教師データセットからバッチサイズの数だけデータを抽出することでバッチを作成する関数であり，6,7 行目の $train$ はバッチデータに基づき NER モデル $Model$ のパラメータを更新する関数である．また，8 行目の $is\_converge$ は対象タスクに対する損失に基づき学習の終了判定を行う関数である．

3. 提案手法：複数の補助教師データを用いる補助学習

2.2 節で説明した従来の補助学習では，補助教師データとして 1 種類の教師データしか用いていない．本節では，複数種類の教師データを補助教師データとして活用する手法を提案する．3.1 節と 3.2 節では複数の補助教師データを一つの補助学習で用いる手法を提案し，3.3 節では補助教師データの種類の数だけ補助学習を順次行い，メインモデルをファインチューニングする手法を提案する．

Algorithm 2: 補助結合モデルのアルゴリズム

Data: メイン教師データ D_{main}，M 種類の補助教師データ $D^{(1)}_{aux}, D^{(2)}_{aux}, \cdots, D^{(M)}_{aux}$

1 **begin**
2 　$D_{aux} = [D^{(1)}_{aux}; D^{(2)}_{aux}; \cdots; D^{(M)}_{aux}]$
3 　**for** $i = 1, 2, \cdots, Epoch$ **do**
4 　　**for** $j = 1, 2, \cdots, Iteration$ **do**
5 　　　$Batch_{main} = extract(D_{main}, BatchSize)$
6 　　　$Batch_{aux} = extract(D_{aux}, BatchSize)$
7 　　　$train(Model, Batch_{aux})$
8 　　　$train(Model, Batch_{main})$
9 　　$is\_converge(Model)$

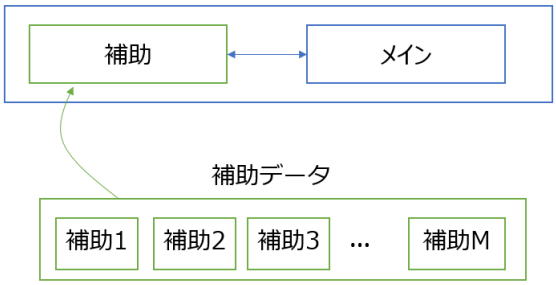

図 3 補助結合モデル

3.1 補助結合モデル

補助結合モデルは，複数の補助教師データを結合したデータセットを一つの補助教師データとみなし，2.2 節で説明した補助学習を行うモデルである．補助結合モデルの概要を**図 3**，アルゴリズムをアルゴリズム 2 に示す．補助結合モデルは，2.2 節の補助モデルと同様，メイン教師データから作成したメインバッチと補助教師データから作成した補助バッチを用意する．そして，メイン教師データに対する損失が収束するまで，補助バッチを用いた学習とメインバッチを用いた学習を交互に繰り返す．補助モデルとの違いは，複数の補助教師データを結合したデータセットからデータを抽出することで補助バッチを作成する点である．一つの補助バッチには複数種類の補助教師データが混在し得る．

3.2 補助反復モデル

補助反復モデルは，エポック毎に補助教師データとして使用する教師データの種類を変える補助学習である．補助反復モデルの概要を**図 4**，アルゴリズムをアルゴリズム 3 に示す．補助反復モデルにおいても，メイン教師データから作成したメインバッチを用いた学習と補助教師データから作成した補助バッチを用いた学習を，メイン教師データに対する損失が収束するまで交互に繰り返す．補助結合モ

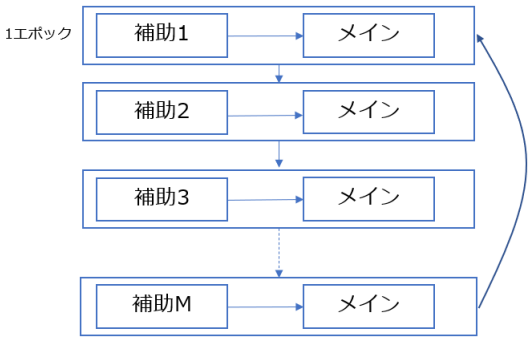

図 4 補助反復モデル

Algorithm 3: 補助反復モデルのアルゴリズム

Data: メイン教師データ D_{main}，M 種類の補助教師データ $D_{aux}^{(1)}, D_{aux}^{(2)}, \cdots, D_{aux}^{(M)}$

begin
  for $i = 1, 2, \cdots, Epoch$ **do**
    for $k = 1, 2, \cdots, M$ **do**
      for $j = 1, 2, \cdots, Iteration$ **do**
        $Batch_{main} = extract(D_{main}, BatchSize)$
        $Batch_{aux} = extract(D_{aux}^{(k)}, BatchSize)$
        $train(Model, Batch_{aux})$
        $train(Model, Batch_{main})$
    $is\_converge(Model)$

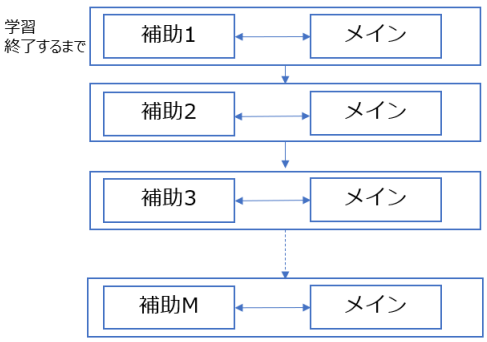

図 5 補助スタックモデル

デルとの違いは，補助バッチは特定の補助教師データセットから作成し，補助バッチの作成元とする補助教師データセットはエポック単位で切り替える点である．

3.3 補助スタックモデル

補助スタックモデルは，補助教師データ毎の補助学習を順次行うことで，補助教師データの種類の数だけメインモデルのファインチューニングを行う方法である．補助スタックモデルの概要を**図 5**，アルゴリズムをアルゴリズム 4 示す．補助スタックモデルでは，特定の補助教師データを用いた補助学習を行う．そして，メイン教師データに対する損失が収束したときに，新しい補助教師データに切り替え，新しい補助教師データを用いて，メインモデルのファインチューニングを行う．補助結合モデルと補助反復モデルでは，メインモデルの学習は一度だけ（収束は一度だけ）だが，補助スタックモデルでは，補助教師データの種類の数だけモデルを学習する．

Algorithm 4: 補助スタックモデルのアルゴリズム

Data: メイン教師データ D_{main}，M 種類の補助教師データ $D_{aux}^{(1)}, D_{aux}^{(2)}, \cdots, D_{aux}^{(M)}$

begin
  for $k = 1, 2, \cdots, M$ **do**
    for $i = 1, 2, \cdots, Epoch$ **do**
      for $j = 1, 2, \cdots, Iteration$ **do**
        $Batch_{main} = extract(D_{main}, BatchSize)$
        $Batch_{aux} = extract(D_{aux}^{(k)}, BatchSize)$
        $train(Model, Batch_{aux})$
        $train(Model, Batch_{main})$
      $is\_converge(Model)$

表 1 NER データセット

データセット名	固有表現の種類	アノテーション数
NCBI Disease[8]	Disease	6,881
BC5CDR[9]	Disease	19,665
BC5CDR[9]	Drug/Chem	15,411
CHEMDNER[10]	Drug/Chem	79,842
BC2GM[11]	Gene/Protein	20,703
JNLPBA[12]	Gene/Protein	35,460
LINNAEUS[13]	Species	4,077
s800[14]	Species	3,708

4. 実験

4.1 実験設定

評価実験では，BioCreative IV の CHEMDNER データセット [10] を使用して各モデルの評価を行う．CHEMDNER データセット以外の教師データとして**表 1** に示す 7 種類の NER データセットを使用する．本実験で評価するモデルは，2 節及び 3 節で説明したモデルである．1 種類の補助教師データを使用する従来の補助学習モデルでは，7 種類の NER データセットをそれぞれ補助教師データとして用いたモデルの中で，開発データで最も性能が良かったモデルを評価データで評価した．従来の複数の教師データを用いるマルチタスク学習及び複数の補助教師データを用いる提案の補助学習では，8 種類全てのデータセットを用いる．補助反復モデル及び補助スタックモデルでは，メインモデルの学習終了時点に近い補助教師データほど影響が大きく，これら提案手法の性能は，補助教師データの使用順に影響を受けると考えられる．そこで，CHEMDNER タスクの性能改善に寄与する度合いの小さい補助教師データから順に補助教師データとして用いる．具体的には，CHEMDNER

表 2 実験結果

モデル	F-score (%)
シングルモデル	92.25
マルチタスクモデル	92.20
補助モデル	92.29
補助結合モデル	92.36
補助反復モデル（ソート）	92.39
補助スタックモデル（ソート）	92.35

以外の 7 種類の NER データセットを，各データセット単体で補助教師データとした補助モデルの開発データに対する性能の昇順で，アルゴリズム 3 及び 4 の $D^{(1)}_{aux}, D^{(2)}_{aux}, \cdots$ として用いた．ただし，補助スタックモデルの最後の補助教師データは CHEMDNER と同じ種類の固有表現である BC5CDR-chem とした．補助教師データの入力順に関する考察は 4.3 節で行う．

各 NER モデルはオープンフレームワーク FLAIR[15] を拡張して実装した．単語埋め込みは FLAIR で提供されている Contextual String Embeddings（CSE）[16] と FastText[17] を使用した．CSE と FastText の各モデルはともに，医学文献コーパス Pubmed abstracts より学習したモデルを使用した．BiLSTM 層の次元数は 256 とした．オプティマイザーは SGD を使用し，スケジューリングにより学習率を調整した．具体的には，エポックごとの損失が 4 回連続して，これまで損失の最低値より小さくならなかったときに，学習率を 2 分の 1 倍にした．そして，学習率が 1e-4 以下になったときに学習を終了した．評価時は，学習を終えたときのモデルを使用した．ハイパーパラメータのチューニングでは，学習率の初期値として 0.1，0.05 の 2 通り，バッチサイズとして 16，32 の 2 通りを試した．これらの学習率とバッチサイズを組み合わせた 4 つのモデルを開発データで評価し，一番性能が良いハイパーパラメータの組み合わせを選択した．モデルを評価する際は，教師データと開発データを合わせたデータからモデルを学習し，テストデータに対する性能を比較，評価した．評価指標は以下の F 値を用いた．

$$F\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

4.2 実験結果

実験結果を**表 2** に示す．表 2 の「シングルモデル」は CHEMDNER データセットのみを用いたモデルである．

表 2 より，マルチタスクモデルはシングルモデルに比べて 0.05 ポイント F 値が低い．これより，対象タスク以外の教師データを複数種類利用しても必ずしも NER の性能が改善するとは限らないのが分かる．一方で，各補助モデルはシングルモデルよりも性能が高い．このことから，対象タスクに着目した補助学習の方が全てのタスクの教師データを同等に扱うマルチタスク学習よりも性能改善に寄与すると考えられる．

また，複数の補助教師データセットを用いた補助学習モデルは，いずれも，補助教師データを 1 種類しか用いない従来の補助学習モデルよりも性能が高い．具体的には，補助結合モデル，補助反復モデル（ソート），補助スタックモデル（ソート）は，補助モデルに比べて，F 値がそれぞれ 0.07 ポイント，0.1 ポイント，0.06 ポイント高い．これより，教師データセットが複数ある場合，提案手法のように補助学習において複数の補助教師データを用いることで NER 性能が改善できることが分かり，提案手法の有効性を実験的に確認できる．特に，提案モデルの中で補助反復モデルが最も高い性能を達成し，有効であった．補助反復モデルではメインモデルの学習は一度だけですむのに対して，補助スタックモデルでは用いる補助教師データの種類数だけファインチューニングが必要であるため，学習時間は補助スタックモデルの方が長い．今回の実験設定では，そのような長い学習時間を伴う補助スタックモデルをあえて選択する必要がないことが分かった．

4.3 補助教師データの入力順に関する考察

補助スタックモデルと補助反復モデルの性能は，補助教師データの使用順に影響を受けると考えられる．表 2 の実験では，4.1 節で述べた通り，各データセット単体を補助教師データとした補助モデルの開発データに対する性能に基づいて，補助スタックモデル及び補助反復モデルで用いる補助教師データの使用順を定めた．**表 3** に各データセット単体を補助教師データとして用いた補助モデルの性能及びハイパーパラメータ毎の補助スタックモデル（ソート）と補助反復モデル（ソート）の性能を示す．ただし，学習時は教師データのみを用いて，開発データに対する性能を評価している．表 3 より，補助反復モデル（ソート）では，バッチサイズは 32，学習率は 0.1 とし，補助教師データは「JNLPBA → BC2GM → BC5CDR-disease → s800 → BC5CDR-chem → linnaeus → NCBI-disease」の順で用いた．また，補助スタックモデル（ソート）では，バッチサイズは 16，学習率は 0.05 とし，補助教師データは「s800 → JNLPBA → BC5CDR-disease → BC2GM → NCBI-disease → linnaeus → BC5CDR-chem」の順で用いた．

この補助教師データの使用順の有効性を確認するために，「BC2GM → BC5CDR-disease → JNLPBA → linnaeus → NCBI-disease → s800 → BC5CDR-chem」の順で補助教師データを用いた補助反復モデルと補助スタックモデルの性能を評価し，補助反復モデル（ソート）と補助スタックモデル（ソート）と比較する．この順は，最後の補助教師データは CHEMDNER と同じ種類の固有表現である BC5CDR-chem とし，その他の補助教師データは，固有表現の種類が同じ補助教師データが連続しないようにランダ

表 3 補助モデル，補助反復モデル（ソート），補助スタックモデル（ソート）の開発データにおける性能（F-score(%)）

バッチサイズ		16		32	
学習率		0.05	0.1	0.05	0.1
補助モデル	BC2GM	89.95	89.92	89.88	89.91
	BC5CDR-chem	90.05	90.16	90.00	90.03
	BC5CDR-disease	89.95	90.08	89.87	89.93
	JNLPBA	89.90	89.93	89.88	89.89
	linnaeus	90.15	90.02	89.90	90.04
	NCBI-disease	89.98	90.03	89.69	90.08
	s800	89.88	90.01	89.89	89.94
補助反復モデル（ソート）		90.04	90.11	90.03	**90.14**
補助スタックモデル（ソート）		**90.24**	90.07	89.75	90.14

表 4 補助教師データの入力順の違いによる性能差

モデル	F-score (%)
補助反復モデル（ソートなし）	92.18
補助反復モデル（ソート）	92.39
補助スタックモデル（ソートなし）	92.32
補助スタックモデル（ソート）	92.35

ムに並び替えたものである．結果を**表 4** に示す．表 4 より，補助反復モデル（ソート）は補助反復モデルに比べ，F 値が 0.21 ポイント上回っている．また，補助スタックモデル（ソート）は補助スタックモデルに比べて，F 値が 0.03 ポイント上回っている．この結果より，メイン対象タスクの性能改善に寄与する度合いの小さい順に補助教師データを並び替えて学習することで，提案手法の精度が改善することを確認した．

5. おわりに

本研究では，7 種類の化学/科学技術分野の NER データセットを補助教師データとして活用する NER の補助学習手法を提案した．学習方法として，(i) 複数の補助教師データを結合したデータセットを一つの補助教師データとしみなして補助学習を行う補助結合モデル，(ii) エポック毎に補助教師データとして使用する教師データの種類を変える補助学習を行う補助反復モデル，(iii) 補助教師データ毎の補助学習を順次行い補助教師データの種類の数だけメインモデルのファインチューニングを行う補助スタックモデルを提案した．BioCreative IV の CHEMDNER データセットを用いた評価実験を通じて，全ての提案手法は，補助教師データとして 1 種類の教師データしか用いない従来の補助学習よりも高い性能を実現でき，提案手法の中で補助反復モデルが最も性能が高いことを確認した．そして，補助反復モデルや補助スタックモデルでは，対象タスクの性能改善に寄与する度合いの昇順で補助教師データを用いることで性能が改善できることを確認した．

今後は，他の NER データセットでも提案手法が有効かどうかを確認したい．また，補助教師データとして 7 種類より多くの補助教師データを用いた実験を行いたい．

参考文献

[1] Huang, Z., Xu, W. and Yu, K.: Bidirectional LSTM-CRF models for sequence tagging, *arXiv preprint arXiv:1508.01991* (2015).
[2] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H. and Kang, J.: BioBERT: a pre-trained biomedical language representation model for biomedical text mining, *Bioinformatics*, Vol. 36, No. 4, pp. 1234–1240 (online), DOI: 10.1093/bioinformatics/btz682 (2019).
[3] Wang, X., Lyu, J., Dong, L. and Xu, K.: Multitask learning for biomedical named entity recognition with cross-sharing structure, *BMC bioinformatics*, Vol. 20, No. 1, pp. 1–13 (2019).
[4] Crichton, G., Pyysalo, S., Chiu, B. and Korhonen, A.: A neural network multi-task learning approach to biomedical named entity recognition, *BMC bioinformatics*, Vol. 18, No. 1, pp. 1–14 (2017).
[5] Khan, M. R., Ziyadi, M. and AbdelHady, M.: Mt-bioner: Multi-task learning for biomedical named entity recognition using deep bidirectional transformers, *arXiv preprint arXiv:2001.08904* (2020).
[6] Mehmood, T., Gerevini, A. E., Lavelli, A. and Serina, I.: Combining multi-task learning with transfer learning for biomedical named entity recognition, *Procedia Computer Science*, Vol. 176, pp. 848–857 (2020).
[7] Wang, X., Zhang, Y., Ren, X., Zhang, Y., Zitnik, M., Shang, J., Langlotz, C. and Han, J.: Cross-type biomedical named entity recognition with deep multi-task learning, *Bioinformatics*, Vol. 35, No. 10, pp. 1745–1752 (2019).
[8] Doğan, R. I., Leaman, R. and Lu, Z.: NCBI disease corpus: a resource for disease name recognition and concept normalization, *Journal of biomedical informatics*, Vol. 47, pp. 1–10 (2014).
[9] Li, J., Sun, Y., Johnson, R. J., Sciaky, D., Wei, C.-H., Leaman, R., Davis, A. P., Mattingly, C. J., Wiegers, T. C. and Lu, Z.: BioCreative V CDR task corpus: a resource for chemical disease relation extraction, *Database*, Vol. 2016 (2016).
[10] Krallinger, M., Rabal, O., Leitner, F., Vazquez, M., Salgado, D., Lu, Z., Leaman, R., Lu, Y., Ji, D., Lowe, D. M. et al.: The CHEMDNER corpus of chemicals and drugs and its annotation principles, *Journal of cheminformatics*, Vol. 7, No. 1, pp. 1–17 (2015).
[11] Smith, L., Tanabe, L. K., nee Ando, R. J., Kuo, C.-

ⓒ 2021 Information Processing Society of Japan

J., Chung, I.-F., Hsu, C.-N., Lin, Y.-S., Klinger, R., Friedrich, C. M., Ganchev, K. et al.: Overview of BioCreative II gene mention recognition, *Genome biology*, Vol. 9, No. 2, pp. 1–19 (2008).

[12] Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y. and Collier, N.: Introduction to the bio-entity recognition task at JNLPBA, *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, Citeseer, pp. 70–75 (2004).

[13] Gerner, M., Nenadic, G. and Bergman, C. M.: LINNAEUS: a species name identification system for biomedical literature, *BMC bioinformatics*, Vol. 11, No. 1, pp. 1–17 (2010).

[14] Pafilis, E., Frankild, S. P., Fanini, L., Faulwetter, S., Pavloudi, C., Vasileiadou, A., Arvanitidis, C. and Jensen, L. J.: The SPECIES and ORGANISMS resources for fast and accurate identification of taxonomic names in text, *PloS one*, Vol. 8, No. 6, p. e65390 (2013).

[15] Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S. and Vollgraf, R.: FLAIR: An easy-to-use framework for state-of-the-art NLP, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pp. 54–59 (2019).

[16] Akbik, A., Blythe, D. and Vollgraf, R.: Contextual string embeddings for sequence labeling, *Proceedings of the 27th international conference on computational linguistics*, pp. 1638–1649 (2018).

[17] Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T.: Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146 (2017).

ⓒ 2021 Information Processing Society of Japan