化合物の同義語辞書を用いた固有表現抽出

渡邊 大貴 [1,3]，　田村 晃裕 [1,3]，　二宮 崇 [1,3]，　牧野 拓哉 [2,3]，　岩倉 友哉 [2,3]
[1] 愛媛大学 大学院理工学研究科 電子情報工学専攻, [2] 株式会社富士通研究所
[3] 理研 AIP-富士通連携センター
{t_watanabe@ai, tamura@, ninomiya@}cs.ehime-u.ac.jp
{ makino.takuya, iwakura.tomoya}@jp.fujitsu.com

1 はじめに

固有表現抽出 (NER) は，情報抽出や，エンティティリンキングといったアプリケーションに用いられる自然言語処理の重要な基礎技術の一つである．NER は，事前に固有表現 (NE) と定義した専門用語を文中から抽出する技術である．近年では，ニューラルネットワーク (NN) に条件付確率場 (CRF) を組み合わせたモデルが高い精度を実現している [1, 2]．また，大規模なラベルなしコーパスから事前学習したニューラル言語モデルを用いた手法が CoNLL 2003 shared task データセットなどにおいて，最高精度の手法となっている [3, 4]．

化学分野のデータ解析において，化学分野テキストからの NE 抽出は重要な役割を果たす．現在，実験用データの整理や，特許分析などに用いられる，化合物に関するデータベースは人手で作成されている．しかし，日々大量に報告される化学技術文書を人手で解析することは，非常に困難な状況にある．この問題を解決するために，NER を用いて，自動的に化合物名を抽出する技術が研究されている [5]．しかし，化合物には多様な表記があることが，化学文書の NER を難しくしている．例えば，フェニルアラニンは，L-β-phenylalanine，(S)-2-Benzylglycine や Phenylalanine などの異表記で表現される場合がある．これらの異表記を同一視しないと，同一化合物に対する統計量が分散してしまい，特に低頻度な化合物に対する NER の精度が低くなってしまう可能性がある．しかし，従来のモデルでは，化合物特有の微妙な表記ゆれや，部分構造に基づく別称を考慮する機能を有していない．

そこで，本研究では，化合物名の抽出と化合物の言い換えを Multi-task 学習することで，表現の同一性を学習し化合物名抽出の性能改善を行う手法を提案する．化合物の言い換えには機械翻訳で標準的に用いられるアテンションに基づくニューラル機械翻訳 (ANMT) モデル [6, 7] を使用する．BioCreative IV の CHEMDNER タスクにおける評価実験を通じて，提案手法は従来の BiLSTM-CRF モデル [1] や文字ベースの言語モデルを用いた BiLSTM-CRF[3] より化合物の抽出精度が高いことを示す．また，提案手法は，BioCreative IV の CHEMD-NER タスクで最高精度である，ドキュメントレベルの情報を利用する NN に基づくモデル (Att-ChemdNER)[5] よりも F-score が 1.28 ポイント高いことを示す．

2 NN を用いた NER

本節では，提案手法のベースラインモデルとなる NN を用いた NER の手法について説明する．

2.1 BiLSTM-CRF モデル

BiLSTM-CRF モデル [1] は，LSTM と CRF を用いて NER を行うモデルである．このモデルは，まず，入力文 $\mathbf{w} = w_1, w_2, \cdots, w_N$ が与えられたとき，i 番目の入力単語 w_i を単語埋め込み層により単語埋め込みベクトル $\mathbf{h}_i^{word}$ に変換する．また，i 番目の入力単語を構成する文字列 $\mathbf{w_i} = w_{i,1}, w_{i,2}, \cdots, w_{i,M}$ に対して，j 番目の文字 $w_{i,j}$ を文字埋め込み層により，$\mathbf{c}_{i,j}$ に変換する．その後，次式により定義される双方向の LSTM を用いて，単語 $\mathbf{w_i}$ の文字ベースの隠れ状態を算出する．

$$\overrightarrow{\mathbf{h}_{i,j}^{char}} = LSTM^{(f,char)}(\mathbf{c}_{i,j}, \overrightarrow{\mathbf{h}_{i,j-1}^{char}}), \quad (1)$$

$$\overleftarrow{\mathbf{h}_{i,j}^{char}} = LSTM^{(b,char)}(\mathbf{c}_{i,j}, \overleftarrow{\mathbf{h}_{i,j+1}^{char}}). \quad (2)$$

ここで，$\rightarrow$ と $\leftarrow$ はそれぞれ順方向と逆方向を表し，$LSTM^{(f,char)}$，$LSTM^{(b,char)}$ は順方向と逆方向の LSTM を表す．単語 $\mathbf{w_i}$ に対する文字埋め込みベクトルは，$\mathbf{h}_i^{char} = [\overrightarrow{\mathbf{h}_{i,M}^{char}}; \overleftarrow{\mathbf{h}_{i,0}^{char}}]$ により獲得する．ここで ; はベクトルの結合を表す．単語 w_i に対するベクトル表現 $\mathbf{x}_i$ は，

単語埋め込みベクトル $\mathbf{h}_i^{word}$ と文字埋め込みベクトル $\mathbf{h}_i^{char}$ を用いて次式により算出される．

$$\mathbf{x}_i = [\mathbf{h}_i^{word}; \mathbf{h}_i^{char}]. \tag{3}$$

上記により得られた入力文に対するベクトル表現 $\mathbf{X} = \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ と，双方向の LSTM を用いて，単語 w_i の中間表現を次式により算出する．

$$\overrightarrow{\mathbf{h}_i} = LSTM^{(f)}(\mathbf{x}_i, \overrightarrow{\mathbf{h}_{i-1}}), \tag{4}$$

$$\overleftarrow{\mathbf{h}_i} = LSTM^{(b)}(\mathbf{x}_i, \overleftarrow{\mathbf{h}_{i+1}}), \tag{5}$$

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}], \tag{6}$$

$$\mathbf{e}_i = W_e \mathbf{h}_i. \tag{7}$$

ここで $\mathbf{h}_i$ の次元数は d であり，$W_e \in \mathcal{R}^{k \times d}$ は重みベクトルを表し，k は識別対象のタグの数を表す．

Bi-LSTM CRF モデルでは，タグ付けを各単語に対して独立にモデリングするのではなく，双方向 LSTM の出力系列をスコア行列に変換した $\mathbf{P} = (\mathbf{e}_1 \mathbf{e}_2 \cdots \mathbf{e}_N)^T$ に基づき，CRF を用いて同時にモデリングする．ここで $\mathbf{P} \in \mathcal{R}^{N \times k}$ である．スコア行列 $\mathbf{P}$ の i 行 j 列目の要素を $P_{i,j}$ とすると，$P_{i,j}$ は，i 番目の単語に対する j 番目のタグのスコアである．具体的には，出力タグ系列 $\mathbf{y} = y_1, y_2, \cdots, y_N$ に対するスコアを次式で定義する．

$$s(\mathbf{w}, \mathbf{y}) = \sum_{i=0}^{N} A_{y_i, y_{i+1}} + \sum_{i=1}^{N} P_{i, y_i}. \tag{8}$$

ここで，$\mathbf{A}$ は遷移スコアの行列であり，$A_{i,j}$ は i 番目のタグから j 番目のタグに遷移するスコアを表現する．出力タグ系列 $\mathbf{y}$ の確率は，次式の通り，入力系列 $\mathbf{w}$ に対するすべての可能なタグ系列 $\mathbf{Y_w}$ 上の softmax 関数により計算される．

$$p(\mathbf{y}|\mathbf{w}) = \frac{e^{s(\mathbf{w}, \mathbf{y})}}{\sum_{\tilde{\mathbf{y}} \in \mathbf{Y}_w} e^{s(\mathbf{w}, \tilde{\mathbf{y}})}}. \tag{9}$$

学習時は，正しいタグ系列を用いて次式を最大化する．

$$\log(p(\mathbf{y}|\mathbf{w})) = s(\mathbf{w}, \mathbf{y}) - \log(\sum_{\tilde{\mathbf{y}} \in \mathbf{Y_w}} e^{s(\mathbf{w}, \tilde{\mathbf{y}})}). \tag{10}$$

デコード時は，次式で算出されるスコアを最大化することで，出力タグ系列が得られる．

$$y^* = \arg\max_{\tilde{\mathbf{y}} \in \mathbf{Y_w}} s(\mathbf{w}, \tilde{\mathbf{y}}). \tag{11}$$

2.2 ニューラル言語モデルを用いた NER

本節では，CoNLL2003 shared task で最高精度である，文字レベルの言語モデルを用いた NER(BiLSTM-CRF + Contextual String Embeddings)[3] について説明する．以降，このモデルを「BiLSTM-CRF + CSE」と記す．初めに文字レベルの言語モデルの学習方法について述べ，その後，文字レベルの言語モデルと BiLSTM-CRF を組み合わせる方法について述べる．

文字レベルの言語モデルは，文字系列 $(\mathbf{c}_0, \mathbf{c}_2, \cdots, \mathbf{c}_T) =: \mathbf{c}_{0:T}$ に対する分布 $P(\mathbf{c}_{0:T})$ を推定することを目的とし，与えられた過去の文字列から次の文字を推定するモデル $P(\mathbf{c}_t|\mathbf{c}_0, \cdots, \mathbf{c}_{t-1})$ を学習する．文全体の結合確率は，先行する文字系列を条件とする文字の分布の予測の積で計算する．

$$P(\mathbf{c}_{0:T}) = \prod_{t=0}^{T} p(\mathbf{c}_t|\mathbf{c}_{0:t-1}). \tag{12}$$

条件付確率 $p(\mathbf{c}_t|\mathbf{c}_{0:t-1})$ は順方向の $LSTM^{(f,CharLM)}$ の出力 $\overrightarrow{\mathbf{h}_t}$ を用いて次式の通り近似する．

$$P(\mathbf{c_t}|\mathbf{c}_{0:t-1}) \approx \prod_{i=0}^{t} P(\mathbf{c}_i|\overrightarrow{\mathbf{h}_i}), \tag{13}$$

$$\overrightarrow{\mathbf{h}_i} = LSTM^{(f,CharLM)}(\mathbf{c}_{i-1}, \overrightarrow{\mathbf{h}_{i-1}}). \tag{14}$$

また，逆方向のモデルも同様に，逆方向の $LSTM^{(b,CharLM)}$ を用いて定義される．

$$P(\mathbf{c_t}|\mathbf{c}_{t+1:T}) \approx \prod_{i=t+1}^{T} P(\mathbf{c}_i|\overleftarrow{\mathbf{h}_i}), \tag{15}$$

$$\overleftarrow{\mathbf{h}_i} = LSTM^{(b,CharLM)}(\mathbf{c}_{i+1}, \overleftarrow{\mathbf{h}_{i+1}}). \tag{16}$$

次に，文字レベルの言語モデルを BiLSTM-CRF に組み込む方法について説明する．BiLSTM-CRF+CSE モデルでは，単語 w_i に対するベクトル表現 $\mathbf{x}_i$ として，式 (3) ではなく，単語埋め込みベクトル $\mathbf{h}_i^{word}$，文字埋め込みベクトル $\mathbf{h}_i^{char}$ に，文字レベルの言語モデルによる単語 w_i の単語畳み込みベクトル $\mathbf{h}_i^{CharLM}$ を加えた，次式の通り定義されたベクトルを用いる．

$$\mathbf{x}_i = [\mathbf{h}_i^{CharLM}; \mathbf{h}_i^{word}; \mathbf{h}_i^{char}]. \tag{17}$$

$\mathbf{h}_i^{CharLM}$ は，順方向の文字レベル言語モデルで文頭から単語 w_i の最後の文字まで畳み込んだベクトル $\overrightarrow{\mathbf{h}_{w_{i+1}-1}}$ と，逆方向の文字レベル言語モデルで文末から単語 w_i の最初の文字まで畳み込んだベクトル $\overleftarrow{\mathbf{h}_{t_i-1}}$ の結合で算出する．

$$\mathbf{h}_i^{CharLM} := [\overrightarrow{\mathbf{h}_{t_{i+1}-1}}; \overleftarrow{\mathbf{h}_{t_i-1}}]. \tag{18}$$

入力文に対するベクトル表現が得られた後は，BiLSTM-CRF と同様の処理を行いタグ系列 $\mathbf{y}$ を予測する．

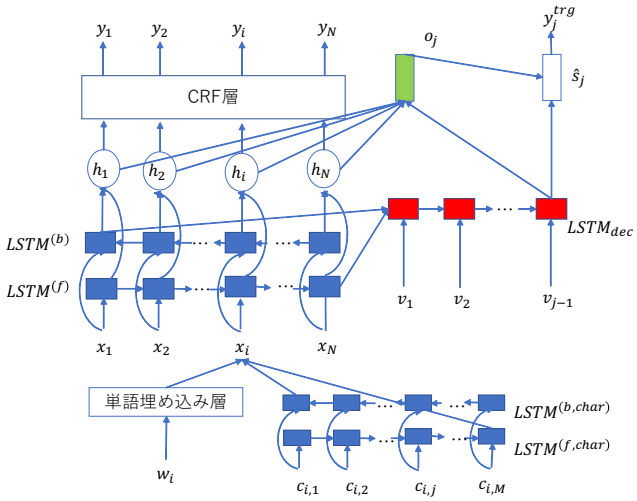

図 1: BiLSTM-CRF と化合物名の言い換えモデルとの Multi-task 学習を行うモデルの全体像

3 提案手法

本節では，まず，提案モデルの構造を述べ，その後，Multi-task 学習について述べる．

3.1 言い換えモデル

本研究では，化合物名の言い換えを行うモデルとして，機械翻訳で標準的に用いられる ANMT モデル [6, 7] を使用する．提案手法の全体像を図 1 に示す．ANMT モデルは，原言語文を固定長のベクトルに変換するエンコーダ用の RNN と，変換した固定長のベクトルから目的言語文を生成するデコーダ用の RNN を用いて翻訳を行うモデルである．

ANMT モデルのエンコーダとして式 (4)，式 (5) で定義される双方向の LSTM を使用する．また，入力文に対する単語埋め込みベクトルは，文字ベースの言語モデルを使用しない場合は式 (3)，使用する場合は式 (17) を使用する．式 (4) により生成される $\overrightarrow{\mathbf{h}}_N$ および，式 (5) により生成される $\overleftarrow{\mathbf{h}}_0$ を結合したベクトル $\mathbf{s}_0 = [\overrightarrow{\mathbf{h}}_N; \overleftarrow{\mathbf{h}}_0]$ をデコーダの LSTM の初期状態とする．

デコーダは，エンコーダによって与えられる入力文の情報に基づき，言い換えの系列 $\mathbf{y}^{trg} = y_1^{trg}, y_2^{trg}, \cdots, y_T^{trg}$ を出力する．まず，j 番目の LSTM デコーダの隠れ状態 $\mathbf{s}_j$ を，次式により算出する．

$$\mathbf{s}_j = LSTM_{dec}([\mathbf{v}_{j-1}; \hat{\mathbf{s}}_{j-1}], \mathbf{s}_{j-1}). \tag{19}$$

ここで，$\mathbf{v}_{j-1}$ は $j-1$ 番目の出力トークンである y_{j-1}^{trg} の単語埋め込みベクトル，$\hat{\mathbf{s}}_{j-1}$ はアテンションベクトルを表す．その後，j 番目のアテンションベクトル $\hat{\mathbf{s}}_j$ を，コンテキストベクトル $\mathbf{o}_j$ を使用して次式の通り算出する．

$$\hat{\mathbf{s}}_j = \tanh(W_e[\mathbf{s}_j : \mathbf{o}_j]). \tag{20}$$

ここで，W_e は重み行列，tanh はハイパボリックタンジェント関数である．コンテキストベクトル $\mathbf{o}_j$ は，式 (6) で表される隠れ状態の加重平均であり，次式により算出される．

$$\mathbf{o}_j = \sum_{i=1}^{N} \alpha_j(i)\mathbf{h}_i. \tag{21}$$

また，アテンションスコア $\alpha_j(i)$ は次式により算出される．

$$\alpha_j(i) = \frac{\exp(\mathbf{h}_i \cdot \mathbf{s}_j)}{\sum_{k=1}^{N} \exp(\mathbf{h}_k \cdot \mathbf{s}_j)}. \tag{22}$$

ここで，exp は指数関数を表す．j 番目の出力トークンの確率分布は，次式により求める．

$$p(y_j^{trg} | \mathbf{y}_{<j}^{trg}, \mathbf{w}) = \mathrm{softmax}(W_s \hat{\mathbf{s}}_j). \tag{23}$$

ここで W_s は，重み行列を表す．目的関数は次式で表される．

$$J(\theta) = -\sum_{(\mathbf{w}, \mathbf{y}^{trg}) \in \mathbf{D}} \log p(\mathbf{y}^{trg} | \mathbf{w}). \tag{24}$$

ここで，D はデータセットを表し，θ はモデルパラメータである．

3.2 Multi-task 学習

提案モデルでは，2 節で説明した NER モデルと 3.1 節の化合物名の言い換えモデルとの Multi-task 学習を行う．目的関数は式 (10) と式 (24) を用いる．言い換えモデルの教師データは PubChem 名称辞書から作成する．具体的には，PubChem 名称辞書に含まれる同一 ID の化合物ペアを学習することで，単語から単語への言い換えモデル（ある化合物のある表記から同一化合物の異表記への変換モデル）を学習する[1]．三つ以上同じ ID の化合物がある場合は，ランダムに二つ選択する．提案手法では，NER モデルと化合物名の言い換えモデルの式 (1),(2) および式 (4),(5) の LSTM パラメータと文字埋め込み層の重み行列を共有させることで Multi-task 学習を行う．これにより NER の LSTM 部は，異表記の化合物名に対しても類似するベクトルへ変換できるようになることが期待される．

[1] PubChem 名称辞書の ID が同じ場合，同じ化合物を表す．

表 1: 実験結果

モデル	Precision	Recall	F-score
BiLSTM-CRF[1]	90.58	88.89	89.73
提案手法	90.88	89.29	**90.08**
BiLSTM-CRF+CSE[3]	92.97	91.57	92.26
提案手法+CSE	92.87	91.97	**92.42**
Att-ChemdNER[5]	92.29	90.01	91.14

4 実験

4.1 実験設定

評価実験は、Luo ら [5] により前処理が行われている BioCreative IV の CHEMDNER データセットを用いた[2]．単語埋め込み層は，PubMed データ[3] に対して word2vec で事前学習したものを初期値として使用した．また，文字ベースの言語モデルの学習にも PubMed を使用した．PubChem 名称辞書から抽出した単語は，-,(,),[,] の前後に空白文字を挿入し，分割した．化合物名の言い換えモデルの教師データは，PubChem 名称辞書からランダムに抽出した 10 万単語対を使用した．単語埋め込み層，文字埋め込み層，文字 LSTM，NER 及び言い換えモデルのエンコーダ LSTM，言い換えモデルのデコーダ LSTM、文字言語モデルの LSTM の次元数はそれぞれ，$100, 25, 50, 200, 400, 2048$ とした．精度評価には，開発データに対して F-score が最も高い epoch のモデルパラメータを使用した．

4.2 実験結果

NER と化合物名の言い換えモデルの Multi-task 学習の有効性を検証するため，従来の BiLSTM-CRF モデル，BiLSTM-CRF+CSE モデルと，言い換えモデルを有する提案モデルを比較した．ベースラインモデルと提案モデルの違いは，言い換えモデルによる Multi-task 学習を行う構造のみである．表 1 に実験結果を示す．表 1 より，提案手法は，BiLSTM-CRF モデルと比べ，0.35 ポイント F-score が高いことが分かる．また，従来の BiLSTM-CRF+CSE モデルと比較して，提案手法+CSE は 0.16 ポイント F-score が上昇することが分かる．これらの結果より提案手法は NER の性能改善に寄与することが実験的に確認できる．また，提案手法は，本実験で使用したデータセットにおいて最高精度を示している Att-ChemdNER[5] より 1.28 ポイント高い精度を達成できたことが分かる．

5 おわりに

本研究では，化学文書に対する NER の性能を改善するために，化合物の同義語辞書を用いた言い換えモデルと NER モデルを Multi-task 学習する手法を提案した．BioCreative IV の CHEMDNER データセットを用いた実験において，BiLSTM-CRF+CSE に比べて，提案手法は 0.16 ポイント F-score が上昇した．また BioCreative IV の CHEMD-NER タスクで最高精度となっている，Att-ChemdNER より高い精度を達成できることを確認した．

今後は，言い換えモデルに使用するデータ量を増やすなどし，表記ゆれや別称に対してより強固なモデルに改良したい．

参考文献

[1] G Lample, M Ballesteros, S Subramanian, K Kawakami, and C Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 NAACL-HLT*, pp. 260–270, 2016.

[2] X Ma and E Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th ACL*, pp. 1064–1074, 2016.

[3] A Akbik, D Blythe, and R Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th COLING*, pp. 1638–1649, 2018.

[4] M Peters, M Neumann, M Iyyer, M Gardner, C Clark, K Lee, and L Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 NAACL-HLT*, pp. 2227–2237, 2018.

[5] L Luo, Z Yang, P Yang, Y Zhang, L Wang, H Lin, and J Wang. An attention-based bilstm-crf approach to document-level chemical named entity recognition. *Bioinformatics*, pp. 1381–1388, 2018.

[6] T Luong, H Pham, and C D. Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 EMNLP*, pp. 1412–1421, 2015.

[7] D Bahdanau, K Cho, and Y Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd ICLR.*, 2015.

[2] https://github.com/lingluodlut/Att-ChemdNER
本実験で使用したコーパスは単語境界の問題を考慮していない．

[3] https://www.nlm.nih.gov/databases/download/pubmed_medline.html