畳み込みニューラルネットワークを用いた表ラベリングによる固有表現認識と関係抽出

Youmi Ma　平岡達也　岡崎直観
東京工業大学
{youmi.ma@nlp., tatsuya.hiraoka@nlp., okazaki@}c.titech.ac.jp

概要

本稿では，BERT に基づく固有表現抽出と関係抽出の新しい手法として，固有表現と関係のラベルを表現する 2 次元の表を画像と見なし，畳み込みニューラルネットワークを用いて表の要素（ラベル）を予測する手法（TabERT-CNN）を提案する．実験結果から，提案手法は既存手法である TabERT と同程度の性能，すなわち現在の最高性能に匹敵する性能を示した．また，BERT の内部パラメータを更新しなくても提案手法は高い性能を発揮する一方，既存手法はパラメータ更新を省略すると性能が低下することから，BERT の内部で固有表現や関係ラベルの依存関係を獲得している可能性が示唆される．

1　はじめに

固有表現認識（NER）と関係抽出（RE）は，文中で固有表現に言及している箇所の認識と，固有表現スパン間の関係を抽出するタスクである．近年，BERT [1] などの事前学習済みモデルから文脈付き埋め込み表現を取り出し，NER と RE に特化した層を積み重ねることで，性能を向上させた例が多く報告されている [2, 3, 4, 5, 6, 7]．そのため，NER と RE の研究では，事前学習済みモデルの上に積み重ねる層の設計に焦点が当てられる．

NER と RE を BERT を用いた表埋め問題（table filling）[8] として解くモデルとして，TabERT が提案された [9]．表埋めによる NER と RE は，図 1 に示すように，表の対角要素に固有表現（NE），非対角要素に関係のラベルを予測することで，NER と RE を統一的に解く．TabERT は，BERT をエンコーダとして用いた埋め込み表現に bi-affine 変換を適用し，表の非対角要素を同時に予測する．この手法はシンプルでありながら，提案当時の最高性能を達成した．一方で，予測された関係ラベル間の依存関係

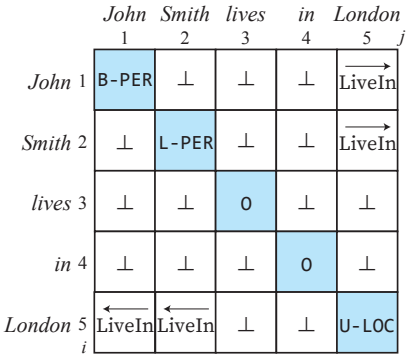


図 1　Table-Filling の概要．セル内の ⊥ は対応する関係がないことを示す．

を考慮していない．Ma ら [9] の論文では関係ラベルの予測の順番を工夫する実験を行ったが，ラベルを独立に予測する手法と有意な差が生まれなかった．

本稿では，表埋めにおける要素間の依存関係を考慮した手法として，TabERT-CNN を提案する．TabERT-CNN は，２次元の表のセルを画素，表全体を画像と見なし，画像・物体認識でよく用いられる２次元の畳み込みニューラルネットワーク（CNN）により表埋め行う．BERT の出力に CNN を適用することにより，近傍のセルの局所的な情報とラベル間の依存関係が考慮されると期待される．また，CNN の層を積み重ねることで，依存関係を考慮できる範囲を拡張し，離れたセル間の依存関係を取り込むことができる．

実験では，CoNLL04 [10]，ACE05[1] と ADE [11] データセットを用いる．評価実験では，TabERT-CNN は TabERT に匹敵する性能を示したが，両手法の明確な性能差は見られなかった．これは，BERT を NER と RE に適応させるための fine-tuning 過程において，固有表現や関係ラベルの予測に関する依存関係を BERT の内部で獲得しているためであると考える．この仮説を検証するため，fine-tuning 時に

1）https://catalog.ldc.upenn.edu/LDC2006T06

BERT 内部のパラメータを更新／固定することによる性能を比較する．さらに，BERT から埋め込み表現を取り出す層の位置を変える実験を行う．その結果，BERT の内部パラメータを更新しなくても TabERT-CNN は高い性能を発揮する一方で，TabERT 等の手法ではパラメータ更新を行わないと性能の大幅な低下が見られることから，トークンやラベル間の依存関係が BERT の内部で考慮されている可能性が示唆される．

2 提案手法

NER と RE の目的は，自然文で書かれた単語列 $S = w_1, w_2, \cdots, w_n$ から固有表現と関係の組 $(a_0\langle t_0\rangle, r, a_1\langle t_1\rangle)$ を抽出することである．ただし a_0, a_1 はそれぞれ一つ以上の $w_i, i \in \{1, \cdots, n\}$ からなる NE を表し，t_0, t_1 は a_0, a_1 に対応する NE ラベルである．r は NE の組 a_0, a_1 の関係ラベルを表す．NE と関係ラベルの集合をそれぞれ，$\mathscr{E}$，$\mathscr{R}$ とする．本稿では，TabERT に基づいて表埋め [8] を用いた新しい手法を提案する．提案手法の説明の前に，まず TabERT を紹介する．

2.1 TabERT

長さ n の入力に対し，$n \times n$ の行列 $\boldsymbol{Y}$ の要素を予測することで，NER と RE を同時に解く．図 1 に示すように，行列の対角要素 $Y_{i,i} \in \mathscr{E}$ は，i 番目のトークンに対応する NE のラベルを表し，非対角要素 $Y_{i,j} \in \mathscr{R}$ は，i 番目のトークンから j 番目のトークンへの関係ラベルである．ここで，NE ラベルは BILOU 表記 [12] に基づいて付与される．関係ラベルは向きを区別するものを採用し，行に対応する要素から列に対応する要素の関係の方向と定義する．また，NE が複数の単語にまたがる場合，NE を構成する全ての単語の要素に対して関係ラベルを付与する．例えば図 1 では，関係 (John Smith⟨PER⟩, LiveIn, London⟨LOC⟩) に対し，(John, London) に対応する $Y_{1,5}$ と (Smith, London) に対応する $Y_{2,5}$ には $\overrightarrow{\text{LIVEIN}}$ が付与され，(London, John) に対応する $Y_{5,1}$ と (London, Smith) に対応する $Y_{5,2}$ には $\overleftarrow{\text{LIVEIN}}$ が付与される．

TabERT では，表の上三角部分だけを埋めることで NER と RE を実行する [13, 14]．具体的には，BERT の出力から系列ラベリング問題で NER を解き，表の対角部分を埋める．この結果を基に，関係を示す非対角部分を同時に埋める．

2.2 TabERT-CNN

提案手法は TabERT [9] と同じく表埋めアーキテクチャを採用するが，ラベル間の依存関係を考慮するために 2 次元 CNN を用いる．提案手法（TabERT-CNN）は，TabERT と同様に表の上半分を使い，NE のラベル表記に BILOU 表記を採用する．

単語埋め込みは，事前学習済みの BERT モデルにより生成する．NER や RE でラベリングを行う単位は単語であるので，既存手法 [3, 9] に倣い，単語 w_i の埋め込みをそのサブワード $t_{\text{start}(i)}, \cdots, t_{\text{end}(i)}$ の埋め込みの最大値プーリングとして計算する．

$$\boldsymbol{e}_{w_i} := \max(\boldsymbol{e}_{t_{\text{start}(i)}}, \cdots, \boldsymbol{e}_{t_{\text{end}(i)}}). \tag{1}$$

ただし，$\boldsymbol{e}_t \in \mathbb{R}^{d_{\text{emb}}}$ はサブワード t に対する BERT の出力（d_{emb} は出力の要素数），$\max(\cdot)$ は最大値プーリング関数である．従って，$\boldsymbol{e}_{w_i} \in \mathbb{R}^{d_{\text{emb}}}$ である．

予測モデルでは，画像・物体認識で広く用いられる 2 次元 CNN [15] を，局所的な周辺情報を考慮した表埋め問題に応用する．2 次元 CNN で表の各要素の表現をエンコードすることで，カーネルサイズの範囲にある周辺単語の依存関係を考慮できる．また，CNN の層を積み重ねることにより，依存関係を考慮できる範囲を拡張できる．

具体的には，まず全ての単語対 (w_i, w_j) に対し，w_i, w_j の埋め込みを結合したベクトルを図 1（右）に示す順に並べ，テンソル $\mathsf{E} \in \mathbb{R}^{n \times n \times d_{\text{emb}}}$ を構築する．これを CNN への入力 $\mathsf{H}^{(0)}$ とする．

$$\mathsf{H}^{(0)}_{i,j,:} = \boldsymbol{h}^{(0)}_{i,j} := \mathsf{E}_{i,j,:} = [\boldsymbol{e}_{w_i}; \boldsymbol{e}_{w_j}]. \tag{2}$$

ここで，$[\cdot;\cdot]$ はベクトルの連結を表す．表の各要素 $\mathsf{H}^{(0)}_{i,j,:}$ の次元数（チャネル数）は $2d_{\text{emb}}$ であり，これを d_0 と書くことにする．

次に，前の層の出力から次の層の出力を計算する．(w_i, w_j) に対応する l 層目の出力値 $\mathsf{H}^{(l)}_{i,j,:} = \boldsymbol{h}^{(l)}_{i,j} \in \mathbb{R}^{d_l}$ は，$l-1$ 層目の出力値 $\mathsf{H}^{(l-1)} \in \mathbb{R}^{n \times n \times d_{l-1}}$ に，サイズが $d_h \times d_w$ のカーネル $\mathsf{K}^{(l)} \in \mathbb{R}^{d_l \times d_h \times d_w}$ とバイアス項 $\boldsymbol{b}^{(l)} \in \mathbb{R}^{d_l}$ を用いた畳み込み演算で計算する（d_l は l 層目の出力のチャネル数である）．

$$\mathsf{H}^{(l)}_{i,j,:} = \boldsymbol{h}^{(l)}_{i,j} := \boldsymbol{b}^{(l)} + \sum_{c=0}^{d_{l-1}} (\mathsf{K}^{(l)}_{c,:,:} \circ \mathsf{H}^{(l-1)}_{:,:,c})_{i,j}. \tag{3}$$

ここで，$\mathsf{K}^{(l)}_{c,:,:} \in \mathbb{R}^{d_h \times d_w}, \mathsf{H}^{(l-1)}_{:,:,c} \in \mathbb{R}^{n \times n}$ である．$\boldsymbol{P} \circ \boldsymbol{Q}$ は二次元相互相関を計算する演算であり，$\boldsymbol{P} \in \mathbb{R}^{(2a+1) \times (2b+1)}$ の場合，式 (4) により定義する．

$$(\boldsymbol{P} \circ \boldsymbol{Q})_{x,y} := \sum_{h=-a}^{a} \sum_{w=-b}^{b} P_{a+h,b+w} Q_{x+h,y+w}. \tag{4}$$

表 1 NER と RE の実験結果．RE と RE+はそれぞれ固有表現ラベルを考慮しない関係抽出と固有表現ラベルを考慮した関係抽出の F1 スコアである．† は $\text{BERT}_{\text{BASE}}$，‡ は $\text{BERT}_{\text{LARGE}}$，◇ は $\text{ALBERT}_{\text{XXLARGE}}$ [16] を用いた結果である．また，△ は F1 スコアのマイクロ平均であり，▲ は F1 スコアのマクロ平均である．

データセット	モデル	NER	RE	RE+
CoNLL04△	SpERT [3]†	88.9	-	71.5
	Table-Sequence [5]◇	90.1	**73.8**	73.6
	TabERT [9]†	90.2	72.8	72.6
	TabERT [9]‡	**90.5**	**73.8**	**73.8**
	TabERT-CNN †	**90.5**	73.2	73.2
ACE05△	Table-Sequence [5]‡	88.2	67.4	-
	Table-Sequence [5]◇	89.5	67.6	64.3
	PFN [7]◇	89.0	-	**66.8**
	TabERT [9]†	87.6	66.2	62.6
	TabERT [9]‡	88.4	67.5	64.6
	TabERT [9]◇	**89.8**	**67.7**	65.2
	TabERT-CNN †	87.8	65.0	61.8
ADE▲	SpERT [3]†	89.3	-	79.2
	Table-Sequence [5]†	89.7	80.1	80.1
	PFN [7]†	89.6	-	80.0
	PFN [7]◇	**91.3**	-	**83.2**
	TabERT [9]†	89.9	**80.6**	80.6
	TabERT-CNN†	89.7	80.5	80.5

CNN の最終層（L 層目）の出力の要素数を RE のラベル数とし（$d_L = |\mathcal{R}|$），関係ラベルを予測する．

$$\hat{Y}_{i,j} = \text{softmax}(\mathsf{H}^{(L)}_{i,j,:}) \tag{5}$$

固有表現ラベルは，$\mathsf{H}^{(L)}$ に対して $\boldsymbol{W}^{(\text{ent})} \in \mathbb{R}^{|\mathcal{E}|\times|\mathcal{R}|}$ と $\boldsymbol{b}^{(\text{ent})} \in \mathbb{R}^{|\mathcal{E}|}$ による線形変換を通して予測する．

$$\hat{Y}_{i,i} = \text{softmax}(\boldsymbol{W}^{(\text{ent})}\mathsf{H}^{(L)}_{i,i,:} + \boldsymbol{b}^{(\text{ent})}) \tag{6}$$

学習時は，式 (5) と (6) で計算されるラベルの予測確率分布と，正解のラベルの one-hot 表現との交差エントロピーの総和を目的関数とする．

3 実験と考察

3.1 実験結果

提案手法（TabERT-CNN）の性能を確認するため，CoNLL04 [10]，ACE05 と ADE [11] の三つのデータセットで実験し，その性能を表 1 に示した．データの前処理・分割は，既存研究に従う [3, 7]．

表 1 から，TabERT-CNN は CoNLL04 と ADE において，事前学習済みモデルを揃えた場合，TabERT に匹敵する性能を示した．なお，両データセットにおける TabERT-CNN や TabERT の性能は，他の手法と比較しても最高性能に近い．ところが，CNN でセル間の依存関係を組み込んでも，TabERT-CNN の性能は比較手法である TabERT を上回らなかった．

3.2 考察

TabERT-CNN が TabERT の性能を上回らなった理由として，NER と RE タスクの fine-tuning の過程で，トークンやラベル間の依存関係を BERT の内部で獲得していることが考えられる．この仮説を検証するため，BERT の内部のパラメータを fine-tuning 時に固定／更新する場合の比較を行う．併せて，NER と RE の層を接続する BERT の層を位置を変えながら，NER と RE の性能の変化を調べる．この実験では，提案手法の他に TabERT [9] と SpERT [3] を用いる．

SpERT [3] は，事前学習済み BERT モデルの出力を用いて，NER と RE をスパンおよびスパン間の分類問題として解く手法である．訓練データに含まれるスパンや関係を正例，ランダムにサンプリングしたスパンや関係を負例として，分類モデルを学習する．SpERT は予測モデルの設計が単純であるため，BERT 内部の働きを分析しやすいと思われる．

実験設定 $\text{BERT}_{\text{BASE}}$ [1] の異なる層の出力をサブワード埋め込みとして予測モデルに与え，学習を行う．なお，学習時には，事前学習済み BERT のパラメータを固定するかどうかを区別する．CoNLL04 [10] の訓練データでモデルを学習し，開発データで評価を行った結果を表 2 と表 3 に示す．

パラメータ更新による影響 表 2 と表 3 に示すように，BERT 内部のパラメータの更新を省略すると，全ての手法で性能の低下が見られる．特に，BERT のパラメータを固定した SpERT は性能の低下が著しく，BERT の構造の中でタスクに特化したパラメータが獲得されていると考えられる．これに対し，TabERT-CNN は BERT のパラメータを固定しても，比較的高い性能を示した．この結果から，計算資源などの制約により事前学習済みモデルのパラメータ更新が難しい場合，TabERT-CNN は TabERT や SpERT よりも高い性能を発揮すると期待される．なお，BERT のパラメータを固定した場合，10 層目以上（特に 12 層）の埋め込み表現を元に NER と RE を行うと，全ての検討対象において性能の低下が見られた．これは，BERT の上位階層がタスクに特化した情報を持つため [17]，事前学習タスクであるマスク単語予測に特化しすぎていると考えられる．

BERT の階層位置による影響 BERT のパラメータを更新する場合に関して，埋め込みを取り出す層を横軸，F1 スコアを縦軸に描画したグラフを図 2 に示す．図 2 により，BERT のより上位の層の出力

表 2　BERT 内部のパラメータを固定／更新する場合の固有表現認識の F1 スコア（CoNLL04 検証データ）．

手法	パラメータ更新	階層位置 0	1	2	4	6	8	10	12
SpERT [3]	なし	27.4	30.9	32.1	36.5	41.0	40.6	37.2	8.0
	あり	16.4	35.4	49.6	64.7	67.2	69.3	70.2	69.1
TabERT [9]	なし	62.4	68.0	74.8	78.6	81.4	82.0	81.5	80.2
	あり	66.9	78.2	84.1	87.4	88.7	88.2	88.5	88.5
TabERT-CNN	なし	80.3	81.1	83.1	85.1	86.6	86.2	86.0	85.9
	あり	80.5	83.7	85.6	87.0	88.4	88.4	88.3	88.0

表 3　BERT 内部のパラメータを固定／更新する場合の関係抽出の F1 スコア（CoNLL04 検証データ）．

手法	パラメータ更新	階層位置 0	1	2	4	6	8	10	12
SpERT [3]	なし	3.0	3.3	3.7	4.6	7.8	6.0	5.8	0.0
	あり	16.4	35.4	49.6	64.7	67.2	69.3	70.2	69.1
TabERT [9]	なし	28.8	37.4	39.3	47.1	53.0	54.0	55.9	51.7
	あり	36.0	47.9	60.9	66.5	71.3	70.5	71.0	70.7
TabERT-CNN	なし	53.5	54.8	57.6	64.4	66.2	67.1	64.4	61.5
	あり	54.0	59.9	62.3	67.8	70.6	70.3	70.1	70.6

を用いることは，全ての手法において性能の向上に寄与する．特に 0 層目から 6 層目までは，埋め込みの深層化による性能の向上幅が大きい．ところが，8 層目以降では性能の向上は緩やかになる．埋め込みを取り出す層による性能差は，NER よりも RE の方が顕著である．これは，単語対の主語・動詞関係などの長距離依存関係をエンコードするためには，BERT の中位以上の層が必要であるという，Jawahar ら [18] の報告と一致する．

CNN の効果　図 2 に示すように，BERT の下位層の出力を用いる場合，TabERT-CNN は他の手法よりも顕著に高い性能を発揮する．TabERT-CNN は 2 次元の CNN を用いてトークンの周辺の情報をエンコードできるため，文脈情報の統合が進んでいない下位層を埋め込み表現として採用すると，CNN による局所的な特徴の統合の効果が明確に現れるためと考えられる．しかし，用いる埋め込みが上位層になるにつれ，他の比較手法に対する TabERT-CNN の性能上の優位性は徐々に失われる．TabERT-CNN は，入力トークンのエンコードに軽量なアーキテクチャを採用した場合や，エンコーダの内部のパラメータを更新できないような状況において，優れたアーキテクチャであると考えられる．

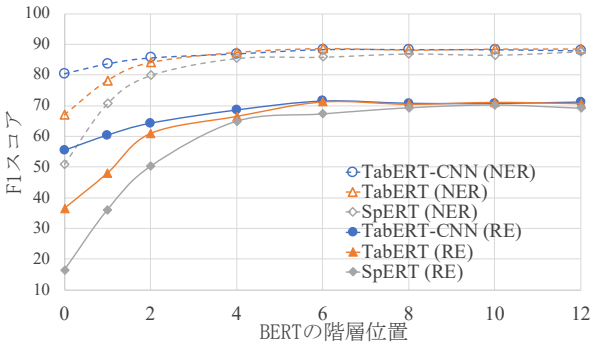


図 2　BERT の各層の出力を埋め込みとして用いた場合の各モデルの性能（BERT はパラメータ更新あり）．評価には CoNLL04 検証データを用いた．

4　おわりに

本稿では，事前学習済み学習済みモデル BERT の上に，CNN で NER と RE を同時に解く新しい手法（TabERT-CNN）を提案した．TabERT-CNN は，表埋めアーキテクチャに基づいており，表のセルを画素，表全体を画像とみなし，画像・物体認識でよく用いられる 2 次元 CNN を用いて NER と RE のラベルを予測する．CoNLL04，ACE05，ADE データセットにおける評価実験では，TabERT-CNN は既存手法である TabERT に匹敵する性能を示した．TabERT と TabERT-CNN の間に明確な性能差は見られなかった原因を探るため，fine-tuning 時に BERT 内部のパラメータを更新／固定することによる性能比較を行った．その実験結果によると，BERT の内部パラメータを更新しなくても TabERT-CNN は高い性能を発揮する一方で，TabERT 等の手法ではパラメータ更新を行わないと性能の大幅な低下が見られることから，トークンやラベル間の依存関係が BERT の内部で考慮されている可能性が示唆された．

今後は，TabERT-CNN のさらなる軽量化として，畳み込み演算の空間方向とチャネル方向への分割や，軽量な BERT モデルとの統合を検討したい．

謝辞

この成果は，国立研究開発法人新エネルギー・産業技術総合開発機構（ＮＥＤＯ）の委託業務（JPNP18002）の結果得られたものです.

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 (NAACL)**, pp. 4171–4186, 2019.

[2] David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. Entity, relation, and event extraction with contextualized span representations. In **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 5784–5789, 2019.

[3] Markus Eberts and Adrian Ulges. Span-based joint entity and relation extraction with transformer pre-training. In **24th European Conference on Artificial Intelligence (ECAI)**, 2020.

[4] Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. A joint neural model for information extraction with global features. In **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 7999–8009, 2020.

[5] Jue Wang and Wei Lu. Two are better than one: Joint entity and relation extraction with table-sequence encoders. In **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1706–1721, 2020.

[6] Zexuan Zhong and Danqi Chen. A frustratingly easy approach for entity and relation extraction. In **North American Association for Computational Linguistics (NAACL)**, 2021.

[7] Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. A partition filter network for joint entity and relation extraction. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 185–197, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1858–1869, 2014.

[9] Youmi Ma, Tatsuya Hiraoka, and Naoaki Okazaki. Named entity recognition and relation extraction using enhanced table filling by contextualized representations. 自然言語処理, Vol. 29, No. 1, p. to appear, March 2022.

[10] Dan Roth and Wen-tau Yih. A linear programming formulation for global inference in natural language tasks. In **Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004**, pp. 1–8, 2004.

[11] Harsha Gurulingappa, Abdul Mateen Rajput, Angus Roberts, Juliane Fluck, Martin Hofmann-Apitius, and Luca Toldo. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. **Journal of Biomedical Informatics**, Vol. 45, No. 5, pp. 885–892, 2012. Text Mining and Natural Language Processing in Pharmacogenomics.

[12] Lev Ratinov and Dan Roth. Design challenges and misconceptions in named entity recognition. In **Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)**, pp. 147–155, 2009.

[13] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. Table filling multi-task recurrent neural network for joint entity and relation extraction. In **Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers (COLING)**, pp. 2537–2547, 2016.

[14] Meishan Zhang, Yue Zhang, and Guohong Fu. End-to-end neural relation extraction with global optimization. In **Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)**, pp. 1730–1740, 2017.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, **Advances in Neural Information Processing Systems**, Vol. 25. Curran Associates, Inc., 2012.

[16] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In **International Conference on Learning Representations (ICLR)**, 2020.

[17] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in BERTology: What we know about how BERT works. **Transactions of the Association for Computational Linguistics**, Vol. 8, pp. 842–866, 2020.

[18] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 3651–3657, Florence, Italy, July 2019. Association for Computational Linguistics.

[19] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. Entity-relation extraction as multi-turn question answering. In **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)**, pp. 1340–1350, 2019.

[20] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In **International Conference on Learning Representations (ICLR)**, 2019.

表 4　実験に用いられた各データセットの統計的情報．

データセット	文の数 訓練	検証	テスト	$\lvert\mathcal{E}\rvert$	$\lvert\mathcal{R}\rvert$
CoNLL04	922	231	288	4	5
ACE05	10,051	2,424	2,050	7	6
ADE	4,272（10 分割交差検証）			2	1

表 5　TabERT-CNN のハイパーパラメータの概要．

	CoNLL04	ACE05	ADE
カーネルサイズ $F_h \times F_w$	3×3	5×5	3×3
層数 L	2	2	3
次元数 $d^{(l)}$	512	512	512|256
バッチサイズ	8	8	16
学習率（$\text{BERT}_{\text{BASE}}$）	5×10^{-5}		
学習率（その他）	1×10^{-3}		
ドロップアウト	0.3		
ウォームアップ期間	0.2		
エポック数	30		

A　データセット

本稿で用いられたデータセットの基本情報を表 4 に示す．なお，全ての実験における報告値は，ランダムシードが異なる 5 つの試行結果の平均である．

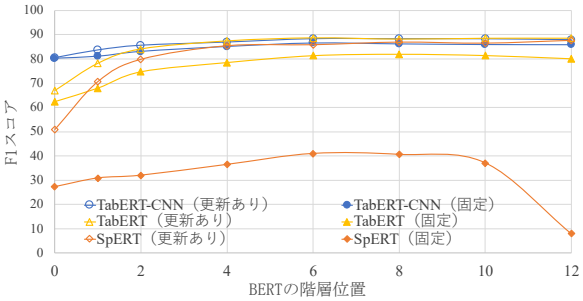
CoNLL04 は，新聞記事から集めたデータセットである．訓練・検証・テスト集の分割は既存研究 [13, 3] に従う．§ 3.1 では，まず訓練データと検証データでハイパー パラメータを選択してから，訓練データと検証デー タをあわせて訓練を行い，得られたモデルの性能を テストデータを用いて評価する．§ 3.2 では，訓練データでモデルを学習し，検証データでモデルの性能を評価する．

ACE05 は，新聞・フォーラムを含む多様なドメインから集めたデータセットである．Wadden ら [2] の前処理スクリプトを用いて，既存研究と同じ分割を作成する [19, 7, 5, 9]．なお，配布されたデータのうち，head 要素として記述された箇所を固有表現のスパンとして扱う．

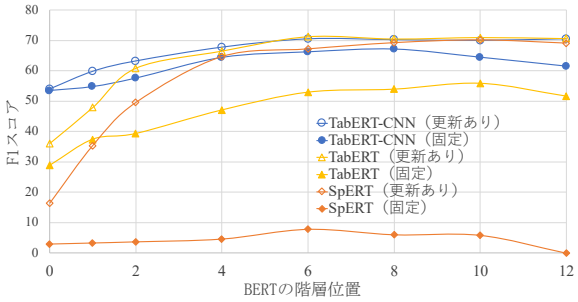
ADE は薬品の使用に関する医療レポートから集めたデータセットである．既存研究に従い [3, 7]，固有表現からオーバラップのある部分を取り除き，性能の測定は 10 分割交差検証を用いる [3]．

B　ハイパーパラメータ

TabERT-CNN におけるハイパーパラメータとその値を表 5 に示す．ただし，訓練時に使われた学習率のスケジューラは，まず学習率をゼロから表に示した値まで { ウォームアップ期間 } × { エポック数 } のエポックをかけて線形に上昇させ，次に線形で減衰



(a) 固有表現抽出の F1 スコア．



(b) 関係抽出の F1 スコア．

図 3　fine-tuning 時に BERT のパラメータを更新／固定する場合の CoNLL04 検証集における性能．（更新あり）／（固定）はそれぞれ fine-tuning 時に BERT のパラメータを更新／固定する場合の性能である．

させるものである．また，NER と RE のモデルとと事前学習済みの BERT モデルに対して，異なる学習率を用いて訓練する．訓練時のパラメータ更新は，AdamW [20] を用いる．

C　パラメータ更新による影響（続）

表 2 と表 3 に示した実験結果を図 3 に示す．図により，TabERT-CNN は BERT モデルのパラメータ更新を行わなくても，一定の性能を示した．なお，図 3 からも，パラメータ更新なしの上位階層（特に 12 層）の出力を使う場合，全検討対象において性能低下が確認できた．