深層言語表現モデルに対するマルチヘッド・多層アテンションによる英語文法誤り検出

金子 正弘　小町 守
首都大学東京

kaneko-masahiro@ed.tmu.ac.jp komachi@tmu.ac.jp

1 はじめに

ニューラルネットワークは大規模コーパスで学習された際に，最も真価を発揮することが知られている．一般的に生コーパスは大規模に存在しており，生コーパスだけから学習することが可能である**言語表現モデル**[1]は NLP において最も大規模コーパスの恩恵を受けることが可能なモデルの一つと言える。そのため，言語表現モデルを様々なタスクに活用する研究が盛んに行われている．

文法誤り検出における大規模生コーパスを活用した研究としては，生コーパスから擬似文法誤りデータを作成し学習データとして活用する研究 [5, 2] などがある．しかし，我々の知る限り大規模生コーパスで事前学習された言語表現モデルを文法誤り検出に活用した研究は存在しない．

事前学習されたモデルをファインチューニングする研究の多くは，事前学習されたモデルの最終層の情報だけを用いて予測を行っている．一方で，ニューラルネットワークは層ごとに異なる表現を学習していることが知られている．例えば，浅い層が局所的な統語関係の学習に最適であり，深い層が長距離関係をモデル化すること [4] や，文脈付き単語分散表現の学習の際に，タスクごとに異なる層の情報を活用することが精度向上に繋がること [3] が報告されている．文法誤り検出などの統語的な情報が重要となるタスクでは，表層情報は深層情報と同様に重要であると考えられる．そのため，様々な層の情報を活用可能なモデルが求められる．

これらのことから，我々は事前学習された深層言語表現モデルの最終層だけを用いるのではなく文法誤り検出に適した表現を多様な層から構築可能なモデルを提案する．図 1 は提案手法である**マルチヘッド・多層アテンション**を示している．マルチヘッドは入力ベクトルの異なる部分空間の情報を別々に扱い結合するため，単一ヘッドと比較して様々な層の情報を活用するように学習すると考えられる．そのため，本モデルでは各層に対してマルチヘッドアテンションを計算する．

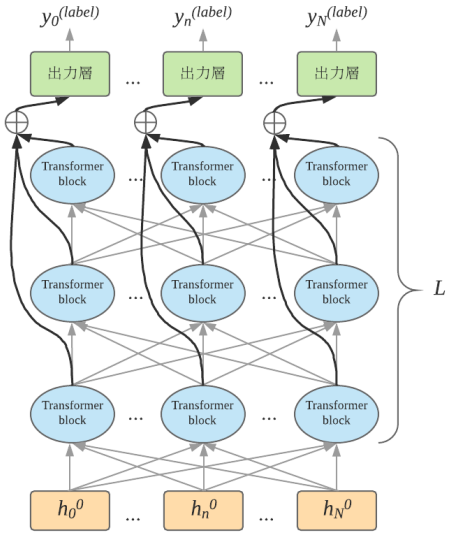


図 1: マルチヘッド・多層アテンションモデル．

本稿の主要な貢献は以下の 3 つである：

- 3 つの文法誤り検出データセット (FCE, CoNLL14 と JFLEG) において，提案手法が既存手法を大幅に上回り世界最高精度を達成した．
- 大規模コーパスで事前学習された言語表現モデルに対して，与えられたタスクに最適な表現を各層から構築するマルチヘッド・多層アテンションを提案した．
- 分析から，層に対するアテンションのヘッド数を増やすことで多様な層から情報を獲得することを示した．さらに，様々な層に対してアテンションを計算することが入力文中のさまざまなトークンの情報を活用する効果があることも明らかにした．

2 言語表現モデル

事前学習された Bidirectional Encoder Representations from Transformers (BERT) [1] に対して，マル

[1]言語モデル，マスク言語モデル，次文を予測することで言語表現を学習するモデルなど [1]

チヘッドアテンションを各層に適応し，文法誤り検出データを用いてファインチューニングを行う (multi-head multi-layer attention, MHMLA).

2.1 BERT

BERT は Transformer モデル [8] のエンコーダーを元に両方向の情報を活用し言語表現を獲得する．大規模データを用いて言語表現を学習し各タスクに適応することで言語理解タスクなどで世界最高精度を達成している．

文法誤り検出（系列ラベリング）における BERT の学習方法を説明する．入力として系列 $S = w_0, \cdots, w_n, \cdots, w_N$ が与えられたとき，以下のように BERT を定式化することができる：

$$h_n^0 = W_{\mathrm{e}} w_n + W_{\mathrm{p}} \quad (1)$$

$$h_n^l = \mathrm{transformer\_block}(h_n^{l-1}) \quad (2)$$

$$y_n^{(\mathrm{BERT})} = \mathrm{softmax}(W_{\mathrm{o}} h_n^L + b_{\mathrm{o}}) \quad (3)$$

w_n は対象トークン，N は系列長であり，式 (1) は入力ベクトルを作成している．W_{e} は単語埋め込みのための重み行列であり，W_{p} は位置エンコーディングのための重み行列である．`transformer_block` は自己アテンションと全結合層から構成されており，h_n^l はその出力である．l ($l \geq 1$) は対象とする層であり，L はモデルの層数である．式 (3) は出力層であり，W_{o} は重み行列，b_{o} はバイアス，$y_n^{(\mathrm{BERT})}$ は予測である．

W_{e}，W_{p} と `transformer_block` はマスクされた単語と次の文を予測することで事前学習される．式 (3) は事前学習時と主タスク時では異なる出力層を用いる．タスクごとの出力層と BERT のすべてのパラメーターはトークン単位の交差エントロピー誤差を最小化することでファインチューニングされる．

2.2 マルチヘッド・多層アテンション

MHMLA は式 (2) の各 `transformer_block` の出力 h_n^l に対してアテンションを計算する．まず，アテンションのためのベクトル v_n^l を算出する：

$$v_n^{l,j} = W_{\mathrm{v}}^{l,j} h_n^l + b_{\mathrm{v}}^{l,j} \quad (4)$$

W_{v} は重み行列，b_{v} はバイアス，j はヘッド数である．我々は非線形変換を h_n^l に対して適応し k_n^l を求める．アテンションスコア a_n^l は以下のように計算される：

$$k_n^{l,j} = \mathrm{ReLU}(W_{\mathrm{k}}^{l,j} h_n^l + b_{\mathrm{k}}^{l,j}) \quad (5)$$

$$a_n^{l,j} = W_{\mathrm{a}}^{l,j} k_n^l + b_{\mathrm{a}}^{l,j} \quad (6)$$

W_{k} と W_{a} は重み行列，b_{k} と b_{a} はバイアスである．この時マルチヘッドは以下のように計算される：

$$\tilde{a}_n^{l,j} = \frac{\exp(a_n^{l,j})}{\sum_{t=1}^{L} \exp(a_n^{t,j})} \quad (7)$$

$$head_n^j = \sum_{t=1}^{L} \tilde{a}_n^{t,j} v_n^{t,j} \quad (8)$$

$\tilde{a}^l$ はアテンション重みスカラーである．$\tilde{a}^l$ と $k_n^{l,j}$ を使いコンテキストベクトル c_n を計算し，最後にタスクのラベルを予測する：

$$c_n = \mathrm{concat}(head_n^1, \cdots, head_n^J) \quad (9)$$

$$y_n^{(\mathrm{label})} = \mathrm{softmax}(W_{\mathrm{o}} c_n + b_{\mathrm{o}}) \quad (10)$$

J はヘッドの合計数，W_{o} と b_{o} は出力層の重み行列とバイアスである．我々のモデルは BERT と同様にトークン単位の交差エントロピー誤差により最適化される．

3 実験

3.1 データセット

我々はファインチューニング，開発と評価のために FCE を用いる．FCE は文法誤り検出において最も一般的なコーパスである．28,731 文を学習， 2,222 文を開発，2,720 文を評価に用いる．そして，追加の評価データとして CoNLL14 と JFLEG を用いる．CoNLL14 は 1,312 文からなり 2 人により正誤のタグ付けがされており，我々は先行研究と同様にそれぞれの正解データに対して評価する．JFLEG は 747 文から構成され 4 つの正解があり，先行研究にならい 4 つの正解を組み合わせて評価データを作成した．JFLEG は本来文法誤り訂正用のコーパスであり訂正箇所がタグ付けされていないため，動的計画法を用いてタグ付けを行った．

3.2 実験設定

我々は公開されている $\mathrm{BERT}_{\mathrm{BASE}}$ を実験に用いた[2]．このモデルは小文字化されていないデータにより学習され，12 層，768 次元の隠れ層，16 個の自己アテンションヘッドにより構成されている．MHMLA の層アテンションのヘッド数は 12 とし，バッチサイズ 32 で最大 5 エポックで学習した．学習データの最大文長は 128 トークンとした．最適化のために学習率 5e-05 で Adam を用いた．h_n^l，$k_n^{l,j}$ と $\tilde{a}_n^{l,j}$ に対して dropout を係数 0.3 で適用した．評価指標として $\mathrm{F}_{0.5}$ を用いた．

3.3 結果

表 1 は FCE，CoNLL14-{1,2} と JFLEG における文法誤り検出の結果を示している．(Rei and Søgaard, 2019)，(Rei et al., 2017) と (Kasewa et al., 2018) は Bi-LSTM を元にした既存手法である．(Rei and

[2] `https://github.com/google-research/bert`

	FCE			CoNLL14-1			CoNLL14-2			JFLEG		
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
(Rei and Søgaard, 2019) [6]	65.53	28.61	52.07	25.14	15.22	22.14	37.72	16.19	29.65	72.53	25.04	52.52
(Rei et al., 2017) [5]	60.67	28.08	49.11	23.28	18.01	21.87	35.28	19.42	30.13	-	-	-
(Kasewa et al., 2018) [2]	-	-	55.6	-	-	28.3	-	-	35.5	-	-	-
$BERT_{BASE}$ w/o pre-train	48.83	11.32	29.37	11.44	7.78	10.45	18.24	9.30	15.30	58.84	13.22	34.82
$BERT_{BASE}$	**69.80**	37.36	59.47	34.11	**33.54**	34.00	45.99	33.92	42.94	**78.06**	36.28	63.45
AvgL	68.08	41.14	60.19	34.98	32.05	34.35	45.41	35.21	42.92	77.34	37.02	63.50
SHMLA	68.41	**43.75**	61.48	33.58	34.39	33.73	45.54	34.85	42.90	76.83	38.74	64.02
MHMLA	68.88	43.47	**61.67**	**35.72**	33.50	**35.25**	**46.46**	**35.51**	**43.76**	77.74	**38.82**	**64.76**

表 1: 文法誤り検出の結果．異なる初期値で 3 回モデルを学習させた結果の平均である．

層	1	2	3	4	5	6	7	8	9	10	11	12
アテンション重みの平均	0.0843	0.0836	0.0823	0.0841	0.0825	0.0822	0.0848	0.0824	0.0848	0.0813	0.0834	0.0835

表 2: MHMLA の各層に対するアテンション重みの平均．

Søgaard, 2019) は学習データとして FCE だけを用いている．(Rei et al., 2017) と (Kasewa et al., 2018) は学習の際 FCE に疑似データを追加している．

1 つ目のベースライン $BERT_{BASE}$ w/o pre-train は，事前学習されずランダムで初期化された BERT 構造のモデルを FCE で学習している．2 つ目のベースライン $BERT_{BASE}$ はオリジナルの BERT である．3 つ目のベースライン averaged layers (AvgL) は最終層の出力を使うのではなく各 h_n^l (式 2) を線形変換し平均した隠れ層を出力層の入力として用いている．4 つ目のベースライン SHMLA (single-head multi-layer attention) は単一ヘッドのアテンションを層に対して計算する．

既存手法と $BERT_{BASE}$ w/o pre-train を比較すると $BERT_{BASE}$ の精度が大幅に向上しており，事前学習された言語表現モデルは文法誤り検出の精度向上につながることがわかる．そして，$BERT_{BASE}$，AvgL と SHMLA と比較して MHMLA はすべてのデータセットでもっとも $F_{0.5}$ が高い．これらのことから各層からマルチヘッドアテンションを用いて最適な中間表現を構築することは文法誤り検出に有効であることがわかる．

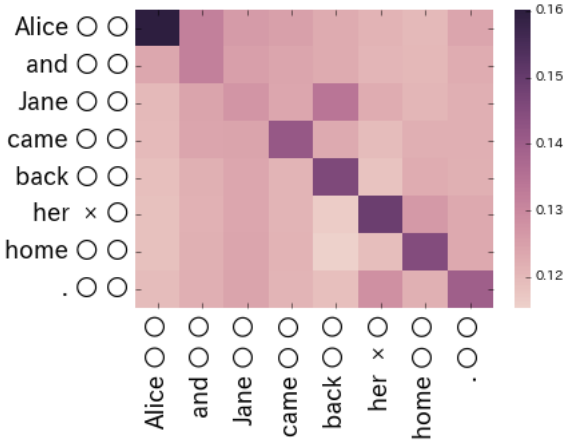
4 分析

MHMLA は最終層以外の情報を考慮しているのか？ MHMLA の目的は最終層以外の情報も考慮することでタスクに適した中間表現を構築することである．目的が達成されているかを調査するために，MHMLA のアテンションの平均を計算し分析する．表 2 は FCE 評価セットにおける MHMLA の各層に対するアテンション重み $\tilde{a}_n^{l,j}$ 式 (7) の平均である．各層でほぼ均一のアテンション重みになっていることがわかる．この結果から，MHMLA は最終層からだけでなく幅広い層の情報を活用して隠れ層を構築していると言える．

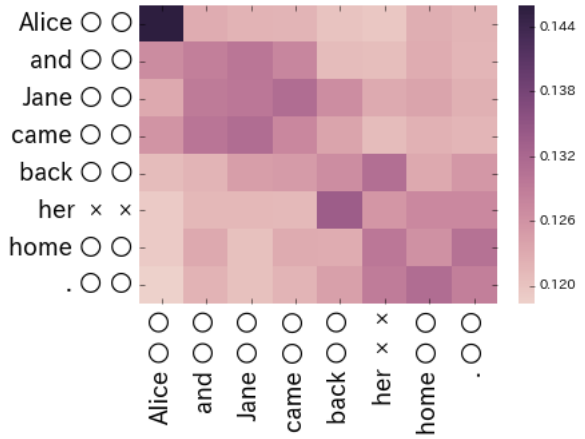
様々な層の情報を活用する利点． 層に対してアテンションを設ける利点として各層の統語的や意味的な情報の違いを考慮可能にすることが挙げられるが，多様な層に対してアテンションを構築することの利点は明確ではない．そのため，多様な層の表現を活用する利点について調査する．

まず，最終層で考慮されているトークンの多様さを調査するために，$BERT_{BASE}$ の最終層と MHMLA の各層の自己アテンション重みの平均に対する標準偏差を調べる．$BERT_{BASE}$ と MHMLA の標準偏差はそれぞれ 0.0080 と 0.0039 であり，MHMLA のほうが小さい．$BERT_{BASE}$ と比較して標準偏差が小さいことは MHMLA の自己アテンション重みがより均一になっていることを示している．このことから，アテンションを用いて様々な層の情報を考慮することで広範囲のトークンの情報を活用していることがわかる．

さらに，具体例を用いて自己アテンションの上記の傾向について調査する．図 2a と 2b は入力文として *Alice and Jane came back her home .* が与えられたときの $BERT_{BASE}$ と MHMLA それぞれの自己アテンションの可視化である．入力文の正解文は *Alice and Jane came back home .* である．MHMLA は各層の自己アテンションを各層に対するアテンションによる重み付き和を計算し可視化している．可視化から $BERT_{BASE}$ は最終層で対角成分の隠れ層だけを用いているが，MHMLA は対角成分だけでなく周辺単語の隠れ層も直接考慮していることがわかる．この例では，came back に対して her が誤りであることを判断する



(a) $\text{BERT}_{\text{BASE}}$ の最終層における自己アテンションの可視化.



(b) 各層の自己アテンションに対して重み付き和された MHMLA の可視化.

図 2: 自己アテンションの可視化．各トークンには 2 つのタグが与えられている．1 つ目は正解であり 2 つ目は検出器の予測を表している．タグ×と○はそれぞれ誤りと正しいトークンである．

には，対象単語だけでなく周辺単語の情報も重要となるため MHMLA は正しく検出できたと考えられる．

5 先行研究

各層の中間表現の活用． 文脈付き単語分散表現 ELMo は大規模コーパスを用いて深い Bi-LSTM を事前学習し，各層の隠れ層からタスクに特化した文脈付き単語分散表現を構築する [3]．我々と同様に層ごとの情報を活用しベクトルを構築しているが，単語分散表現のみが各層の情報を考慮している点と事前学習された Bi-LSTM のパラメーターは更新しない点が我々の手法と異なる．

Takase らは生成タスクにおける出力時のランクの問題を解決するために，各層の中間表現を確率分布計算のために活用した [7]．本研究も同じように各層の情報を考慮しているが，我々の目的は異なる役割を持った表現が層ごとに学習され，各層から与えられたタスクに適した中間表現を構築することである．そして，彼らは各層から確率分布を計算しているのに対し，我々は各層から中間表現を構築している点も異なる．

文法誤り検出における生コーパスの活用． 生成モデルを用いて生コーパスから擬似誤りデータを生成し，文法誤り検出の追加のデータとして活用する研究がある [5, 2]．疑似誤りデータ作成のために誤文から正文を生成するモデルを学習する必要があり，生成された疑似誤りデータは最初に活用できる正誤文のパラレルコーパスの質と量に大きく依存する．そのため，事前学習されたモデルを活用したモデルと比較して生コーパスのスケールの恩恵を受けにくいと考えられる．

6 おわりに

我々は，大規模データで事前学習された深層言語表現モデルに対してマルチヘッド・多層アテンションを構築するモデルを提案した．提案手法は文法誤り検出において大幅に $\text{F}_{0.5}$ を向上させた．分析からヘッド数を増やすことで，より多様な層に対してアテンションを計算するように学習することを明らかにした．さらに，層ごとに活用しているトークンの情報が異なるため，さまざまな層に対してアテンションを構築することは最終層だけを用いたときと比較してより多様なトークンを考慮して予測することを示した．

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. *arXiv*, 2018.

[2] Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection. In *EMNLP*, pp. 4977–4983, 2018.

[3] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *NAACL*, pp. 2227–2237, 2018.

[4] Matthew Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. Dissecting Contextual Word Embeddings: Architecture and Representation. In *EMNLP*, pp. 1499–1509, 2018.

[5] Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. Artificial Error Generation with Machine Translation and Syntactic Patterns. In *BEA*, pp. 287–292, 2017.

[6] Marek Rei and Anders Søgaard. Jointly Learning to Label Sentences and Tokens. In *AAAI*, 2019.

[7] Sho Takase, Jun Suzuki, and Masaaki Nagata. Direct Output Connection for a High-Rank Language Model. In *EMNLP*, pp. 4599–4609, 2018.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NIPS*, pp. 5998–6008, 2017.