Vol.2019-NL-240 No.17
2019/6/14

BERT の下位階層の単語埋め込み表現列を用いた感情分析の教師なし領域適応

白 静[1,a)]　田中 裕隆[2,b)]　曹 類[1,c)]　馬 ブン[1,d)]　新納 浩幸[3,e)]

概要：
BERT は Transformer で利用される Multi-head attention を 12 層（あるいは 24 層）積み重ねたモデルである．各層の Multi-head attention は，基本的に，入力単語列に対応する単語埋め込み表現列を出力しているが，BERT を feature based で利用する場合，各タスクで利用されるのは最上位層の単語埋め込み表現列である．一方，領域適応ではソース領域とターゲット領域の共通部分空間に各領域のデータを写影する手法が有力である．BERT の出力する単語埋め込み表現列から共通部分空間上の特徴ベクトルを構成することを考えた場合，最上位層は BERT の学習で利用したタスクに依存した形になるため，下位層の単語埋め込み表現列と比べて必ずしも最上位層のものが領域適応に対して最適とは限らない．ここでは，この点を確認するために行った感情分析の教師なし領域適応の実験を報告する．

キーワード：BERT，feature based，領域適応，下位階層，共通部分空間

Unsupervised Domain Adaptation for Sentimental Classification by Word Embeddings on the Lower Layer of BERT

Bai Jing[1,a)]　Tanaka Hirotaka[2,b)]　Cao Rui[1,c)]　Ma Wen[1,d)]　Shinnou Hiroyuki[3,e)]

1. はじめに

近年，自然言語処理の多くのタスクで，事前学習モデルを利用する有効性が示されている [5][6]．事前学習モデルは様々なものが提案されているが，その中でも BERT[1] が最も優れた性能を示している．

BERT は Transformer [10] で利用される Multi-head attention を 12 層（あるいは 24 層）積み重ねたモデルであり，各層の Multi-head attention は，基本的に，入力単語列に対応する単語埋め込み表現列を出力している．BERT のような事前学習モデルは feature based と fine tuning の 2 種類の利用方法がある．feature based で利用する場合，通常，BERT の出力の最上位層に現れる特殊 Token である [CLS] の埋め込み表現あるいはそれに続く単語埋め込み表現列を素性として利用する．BERT を感情分析の領域適応に利用する場合，ターゲット領域のラベル付きデータが利用できるのであれば，事前学習モデルを含めたモデル全体を fine tuning するアプローチが有効である．しかしターゲット領域のラベル付きデータが利用できない場合であっても，BERT の出力する単語埋め込み表現列が，文脈に依存したものであることを考えると，BERT の出力は領域依存の度合いが小さく，feature based の利用法を行っても領域適応に対しては有効であることが期待できる．

一方，感情分析の領域適応の手法は事例ベースのものと素性ベースのものに分けられるが，一般に素性ベースの手法の方が性能がよい [4]．素性ベースの手法は，概略，ソー

[1] 茨城大学大学院理工学研究科情報工学専攻
[2] 茨城大学工学部情報工学科
[3] 茨城大学大学院理工学研究科情報科学領域
Ibaraki University, Nakanarusawa 4-12-1, Hiachi, Ibaraki 316–8511, Japan
[a)] 19nd301r@vc.ibaraki.ac.jp
[b)] 16t4032n@vc.ibaraki.ac.jp
[c)] 18nd305g@vc.ibaraki.ac.jp
[d)] 19nd302h@vc.ibaraki.ac.jp
[e)] hiroyuki.shinnou.0828@vc.ibaraki.ac.jp

ス領域とターゲット領域の共通部分空間に各領域のデータを写影する手法とみなせる．BERT の出力する単語埋め込み表現列から共通部分空間上の特徴ベクトルを構成することを考えた場合，最上位層は BERT の学習で利用したタスクに依存した形になるため，必ずしも最上位層の単語埋め込み表現列が領域適応に対して最適であるとは限らない．ここでは，この点を確認するために行った感情分析の領域適応の実験を報告する．

2. 関連研究

感情分析の領域適応の研究は古くから行われている．ディープラーニングが出現する以前の研究はサーベイ論文 [4] や書籍 [9] に，その詳細がまとめられている．ディープラーニング以後は，画像分野で転移学習の研究が活発であり，その知見が言語の研究に応用されている．その代表的なアプローチが事前学習モデルの構築である．事前学習モデルは領域適応の問題に対して効果的な fine tuning に利用できるからである．

OpenAI GPT [6] はニューラルネット翻訳の Transformer [10] の decoder 部分を利用した言語モデルである [*1]．個別のタスクを解くネットワークをそのモデルに連結して利用する．ネットワークのパラメータを学習する際に，連結された言語モデルのパラメータも同時に更新する fine tuning を行うことで，転移学習（領域適応）が行える．言語モデルをタスクに応じて fine tuning するという観点では ULMFiT [2] も知られている．ただし ULMFiT はネットワークの構造を提案したものではなく，言語モデルの fine tuning による転移学習に特化した学習方法を提案している．ELMo [5] は文脈を考慮した単語の分散表現を導くモデルである．実体は 2 層の双方向 LSTM であり，大規模コーパスを利用して言語モデルを学習する．これが事前学習モデルとなり feature based の形で利用できる．

本論文で利用する BERT は従来の事前学習モデルを改善しており，様々なタスクで従来の事前学習モデルの性能を上回っている．このため本論文で扱う感情分析の領域適応であっても，その効果が期待できる．ただし BERT は基本的に fine tuning の形で利用するが，感情分析の領域適応では入力が文でなく文書であることから，feature based の利用が適していると考えられる．また事前学習モデルは領域に依存していないと考えられるので，feature based の利用であっても領域適応に有効であると予想できる．例えば，トピックモデルから得られるトピックベクトルも領域に依存していない情報と考えられるので，feature based の形で領域適応にタスクに有効であることが論文 [11] で示されている．また論文 [8] でも feature based の形で doc2vec [3] を感情分析の領域適応に利用している．本論文の手法もこれらの研究と同じく BERT から得られる情報を追加素性として利用する．

Ruder の博士論文 [7] では，転移学習を Transduction と Inductive に分類している．従来の領域適応分野の用語で言えば，Transduction が教師なし領域適応であり，Inductive が教師あり領域適応である．本論文で扱うのは教師なし領域適応であるため，fine tuning は基本的に利用できないが，BERT は feature based の利用も可能であるため，感情分析の領域適応に利用できる．

3. 提案手法

3.1 BERT

BERT の基本のパーツは Multi-head attention である．Multi-head attention は n 単語埋め込み表現列を入力として，各埋め込み表現をより適切なものに変換して出力する．つまり出力は変換された n 単語埋め込み表現列である．

Multi-head attention の概略を述べる．基本は self attention なので Q, K, V の 3 組が入力である．今，単語埋め込み表現が m 次元であったとする．Multi-head attention では m 次元ベクトルを $d_k(= m/k)$ 次元に圧縮する線形変換器を Q, K, V それぞれに対して用意する．Q, K, V の実体は $d_k \times d_k$ の線形変換行列である．Multi-head attention の入力は n 個の m 次元ベクトルであるが，これが先の圧縮機で $n \times d_k$ の行列 X に変換され，Q, K, V に渡され $n \times d_k$ の行列 XQ, XK, XV ができる．これらを Q', K', V' とおき，以下の式 [*2] により self attention を行う．

$$\texttt{softmax}\left(\frac{Q'K'^T}{\sqrt{d_k}}\right)V'$$

これは $n \times d_k$ の行列である．上記の処理を k 個並行して行うと，$n \times d_k$ の行列が k 個作成され，これらを横に連結することで，$n \times m$ の行列が作成できる．これを更に同次元に線形変換することで Multi-head attention の出力が作られる．

BERT はこの Multi-head attention を 12 層（あるいは 24 層）重ねたモデルである．結局，BERT は n 単語埋め込み表現列を入力とし，それをより文脈に合った n 単語埋め込み表現列に変換していると捉えることができる．

3.2 BERT の学習

BERT におけるパラメータは各層の Multi-head attention が持つパラメータである．つまり各層の持つ 3 つの次元圧縮の線形変換及び k 個の Q, K, V と最後の線形変換がパラメータである．

パラメータの学習に BERT では Masked Language Model と Next Sentence Prediction という 2 つのタスクを用いている．概略述べれば，Masked Language Model は文

[*1] 言語モデルは一種の事前学習モデルである．

[*2] Scaled Dot-Product Attention

中にマスクした単語を当てるタスクであり，Next Sentence Prediction は BERT に与えられた 2 つの文が連続しているものかどうか当てるタスクである．これらのタスクには人手による正解付けが必要なく，教師なしの枠組みで学習できることが特徴である．

3.3 Fine Tuning を用いた感情分析

領域の違いを無視して，単なる感情分析器の学習に BERT を利用する場合，fine tuning の利用法が可能である．この場合，入力文に対する BERT の出力である単語埋め込み表現列の先頭に現れる特殊 Token である [CLS] の埋め込み表現を入力文の特徴ベクトルとして扱い，それを入力とした分類器のネットワークを繋げるのが一般的である．学習では分類器のネットワークに BERT のネットワークを含めた全体のネットワークに対して行えばよい (図 1 参照)．

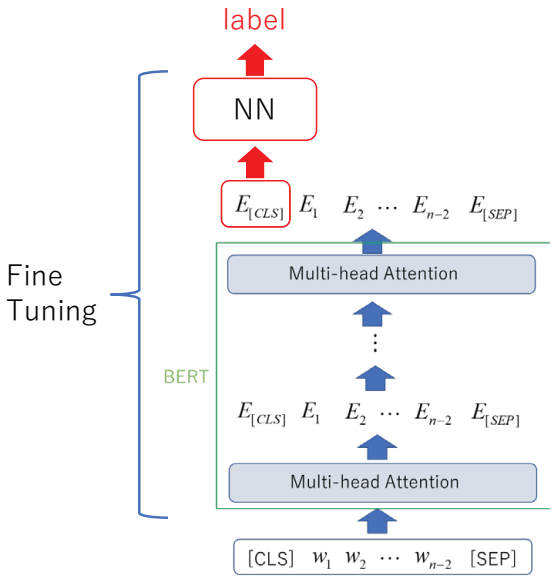

図 1 BERT の Fine Tuning

3.4 単語埋め込み表現列を用いた領域適応

BERT の入力は基本的に 1 文あるいは 2 文である．入力が文書の場合でも，文書を構成する複数の文を 1 つの文として扱えばよい．ただしその場合，単純に [CLS] の埋め込み表現を文書の埋め込み表現とするよりも，[CLS] の埋め込み表現に続く埋め込み表現列から，文書の埋め込み表現を構築した方がよい．

ここでは単純に各単語の埋め込み表現のベクトルの平均ベクトルを作り，それを大きさ 1 に正規化することで文書の埋め込み表現，つまり文書の特徴ベクトルを構築することにする．

3.5 BERT の下位階層の単語埋め込み表現列の利用

本論文で扱うタスクは感情分析の教師なし領域適応である．このため BERT を fine tuning に利用することはできない．ただし，前述したやり方で文書の特徴ベクトルを構築する場合，BERT の出力する単語埋め込み表現列が，文脈に依存したものであることを考えると，BERT の出力は領域依存の度合いが小さく，BERT の出力をそのまま使うだけでも領域適応に対しては有効であることが期待できる．つまり feature based の利用が可能と考えられる．

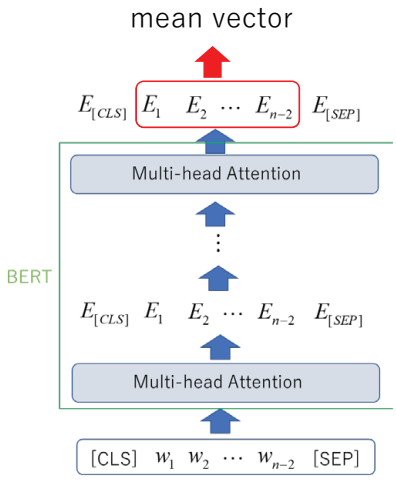

図 2 BERT を用いた文書特徴ベクトルの構築

一方，領域適応ではソース領域とターゲット領域の共通部分空間に各領域のデータを写影する手法が有力である．BERT の出力する単語埋め込み表現列から共通部分空間上の特徴ベクトルを構成することを考えた場合，最上位層は BERT の学習で利用したタスクに依存した形になるため，必ずしも最上位層の単語埋め込み表現列が領域適応に対して最適であるとは限らない．

本論文では BERT の出力する埋め込み表現列の平均ベクトルにより文書の特徴ベクトルを構築するが，BERT の最上位層の埋め込み表現列ではなく，一つ下の層の埋め込み表現列を利用して特徴ベクトルを構築することを提案する．

具体的には，まず，文書 d に対して bag of words モデルと TF-IDF から作られるベクトル v_b を作る．次に d を単語分割し，その単語列を BERT に入力し，単語埋め込み表現列を得る．最上位層の単語埋め込み表現列から作られる平均ベクトルを v_{-1} とする．また最上位層の一つ下の層の単語埋め込み表現列から作られる平均ベクトルを v_{-2} とする．注意といて v_b や v_{-1}，v_{-2} などは大きさ 1 に正規化しておく．提案手法は v_b と v_{-2} を連結したベクトル $[v_b; v_{-2}]$ を d の特徴ベクトルとすることである．

4. 実験

4.1 実験データ

実験で使用したデータセットは，以下のサイトで公開されている Amazon のレビュー文書である．評価の 4,5 を

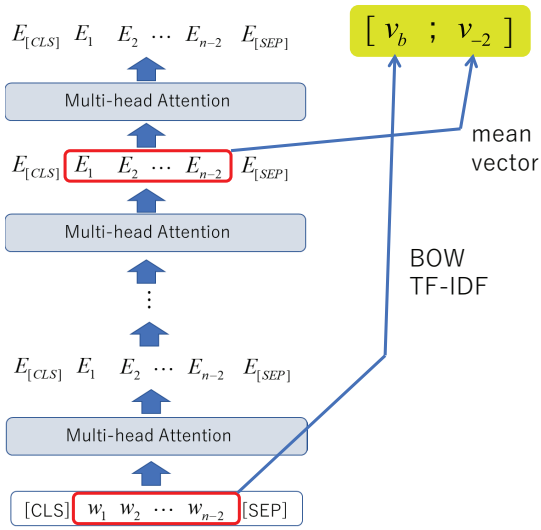

図 3 提案手法による文書特徴ベクトルの構築

positive，評価の 1,2 を negative とした感情分析データとして利用できる．

`https://webis.de/data/webis-cls-10.html`

このデータセットは (B) books，(D) DVD，(M) music の 3 つの領域を持ち，更にそれぞれの領域毎に訓練データ 2,000 文書，テストデータ 2,000 文書を持つ．領域適応の方向としては $B \to D$，$D \to M$，$M \to B$，$B \to M$，$M \to D$，$D \to B$ の 6 通りがある．

4.2 日本語 BERT 事前モデル

公開されている BERT の多言語モデル [*3] には日本語も含まれており，日本語のタスクに対して多言語の事前学習モデルを利用することも可能である．しかし，これを利用すると基本単位が文字になってしまい，適切ではないと考えられる．そこでここでは，日本語に対応した事前学習モデルとして，京都大学黒橋・河原研究室が以下で公開している日本語事前学習モデルを使用する．

`http://nlp.ist.i.kyoto-u.ac.jp/\`
`index.php?BERT 日本語 Pretrained モデル`

またこの事前学習モデルの入力となる文書は，同じく京都大学黒橋・河原研究室が公開している Juman++ [*4] で形態素解析を行い，形態素単位に分割した．

4.3 分類器の学習

$X \to Y$ の領域適応の実験では，領域 X の訓練データの 2,000 文書から分類器を学習し，学習できた分類器を用いて領域 Y のテストデータの 2,000 文書に対する正解率を求める．

[*3] `https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip`
[*4] `http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN++`

分類器は図 4 のような 3 層のニューラルネットワークで構築した．図 4 の L1，L2，L3 はそれぞれ線形変換であり，L1 は入力された文書の特徴ベクトルを 400 次元のベクトルに変換し，L2 はそれを 50 次元のベクトルに変換し，L3 は最後に 2 次元のベクトルとして出力する．L1, L2 の出力には活性化関数としてシグモイド関数を被せ，L3 の出力に対して `softmax_cross_entropy` により損失を求める．

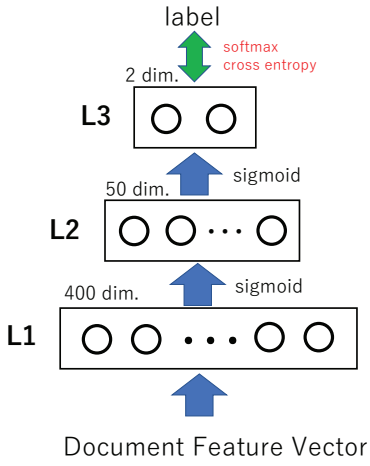

図 4 ニューラルネットワークによる分類器

4.4 実験結果

実験結果を表 1 に示す．v_b(BOW) は bag of words のモデルであり，領域適応の手法を施さない場合の結果である．また v_b（理想値）は，テストデータの領域の訓練データを用いて，分類器を学習した場合の正解率である．$[v_b; v_{-1}]$ が BERT を feature based で標準的に最上位層の情報を利用した場合（標準手法）の結果であり，$[v_b; v_{-2}]$ が BERT が最上位層より 1 つ下の層の情報を利用した場合，つまり提案手法の結果である．$M \to B$，$B \to M$，$D \to B$ の領域適応では提案手法は標準手法よりも高い正解率を出したが，6 つの領域適応の正解率の平均では，わずかに標準手法の方が勝っていた (図 5 参照)．

表 1 実験結果 (正解率)

領域適応	v_b 理想値	v_b BOW	$[v_b; v_{-1}]$ 標準手法	$[v_b; v_{-2}]$ 提案手法
B → D	0.8138	0.7760	**0.7970**	0.7949
D → M	0.8222	0.7824	**0.8032**	0.7942
M → B	0.7817	0.7318	0.7598	**0.7605**
B → M	0.8222	0.7658	0.7954	**0.8014**
M → D	0.8138	0.7708	**0.7868**	0.7814
D → B	0.7817	0.7512	0.7879	**0.7913**
平均	0.8059	0.7630	**0.7884**	0.7873

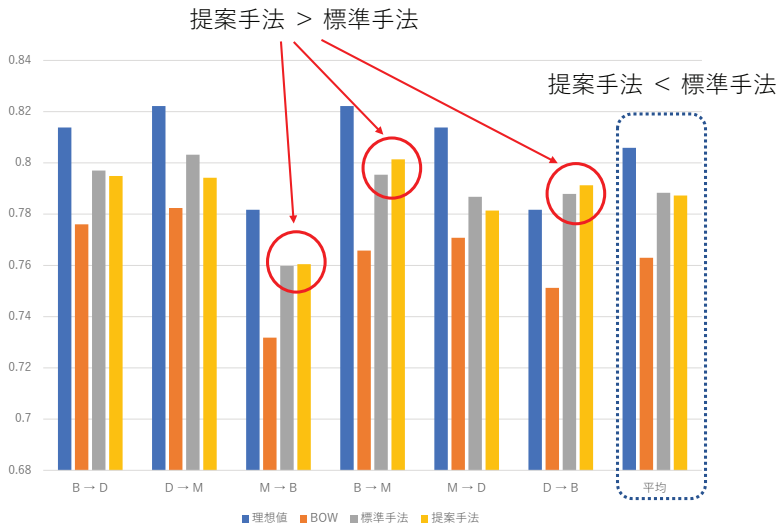

図 5 領域適応毎の標準手法との比較

5. 考察

5.1 より下位の階層の単語埋め込み表現列の利用

提案手法では BERT の出力の最上位から 1 つ下の層の単語埋め込み表現列を利用したが，より下位の階層の単語埋め込み表現列を利用することも考えられる．BERT のすべて層の出力に対して，前述した実験を行った．結果を表 2 に示す．

各領域適応を見ると，必ずしも最上位層 (-1) が最も高い正解率を出すとは限らないことがわかる．ただし 6 つの領域適応の正解率の平均でみると，下の階層ほど正解率は下がっていることも確認できる (図 6 参照)．

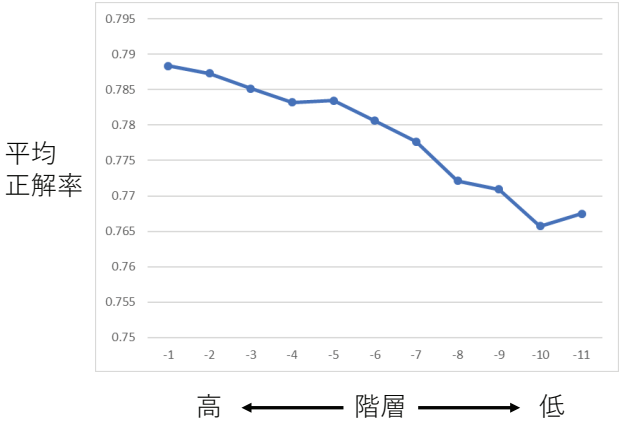

図 6 階層と平均正解率

領域適応では最上位層が必ずしも最良であるとは限らないため，下位の層の情報を併用してゆく手法を今後考えていきたい．

5.2 分散表現列との比較

ここでは BERT の単語埋め込み表現列を用いたが，分散表現データを利用しても，前述した実験は可能である．具体的には，文書内の各単語を分散表現データから分散表現に直し，それらから平均ベクトル v_e を作り，それを提案手法における v_{-2} の代わりに利用すればよい．

分散表現としては nwjc2vec [12] を用いて，提案手法と比較した．実験の結果を表 3 に示す．領域適応の手法を用いない v_b よりも，正解率が高いものもあったが，全体的にはほとんど効果はなかった．単語分散表現は BERT と同じような単語埋め込み表現ではあるが，BERT の方が有用であると言える．

表 3 分散表現との比較 (正解率)

領域適応	v_b BOW	$[v_b; v_{-2}]$ 提案手法	$[v_b; v_{-2}]$ nwjc2vec
B → D	0.7760	**0.7949**	0.7882
D → M	0.7824	**0.7942**	0.7701
M → B	0.7318	**0.7605**	0.7044
B → M	0.7658	**0.8014**	0.7788
M → D	0.7708	**0.7814**	0.7575
D → B	0.7512	**0.7913**	0.7657
平均	0.7630	**0.7873**	0.7608

5.3 Fine Tuning の利用

教師なし領域適応ではターゲット領域のラベル付きデータを利用しないので，fine tuning ができないが，領域の違いを無視すれば可能である．

ここでは BERT のソースと一緒に公開されている `run_classifier.py` [*5] を使うことで，実験データで fine tuning を行った．その結果を表 4 に示す．

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

表 2　階層ごとの識別精度

階層	B → D	D → M	M → B	B → M	M → D	D → B	平均
-1	**0.7970**	**0.8032**	0.7598	0.7954	**0.7868**	0.7879	**0.7884**
-2	0.7949	0.7942	0.7605	**0.8014**	0.7814	**0.7913**	0.7873
-3	0.7955	0.7942	0.7503	0.7974	0.7863	0.7872	0.7852
-4	0.7887	0.7943	0.7588	0.7970	0.7804	0.7801	0.7832
-5	0.7883	0.7869	**0.7610**	0.7962	0.7873	0.7811	0.7834
-6	0.7875	0.7916	0.7586	0.7869	0.7850	0.7741	0.7806
-7	0.7844	0.7922	0.7506	0.7862	0.7818	0.7709	0.7777
-8	0.7827	0.7816	0.7432	0.7825	0.7788	0.7640	0.7721
-9	0.7895	0.7826	0.7424	0.7765	0.7738	0.7608	0.7709
-10	0.7813	0.7717	0.7389	0.7731	0.7672	0.7623	0.7658
-11	0.7835	0.7780	0.7373	0.7756	0.7687	0.7618	0.7675

表 4　Fine Tuning との比較 (正解率)

領域適応	v_b BOW	$[v_b; v_{-2}]$ 提案手法	fine tuning
B → D	0.7760	**0.7949**	0.7699
D → M	0.7824	**0.7942**	0.7854
M → B	0.7318	**0.7605**	0.7364
B → M	0.7658	**0.8014**	0.7874
M → D	0.7708	**0.7814**	0.7474
D → B	0.7512	**0.7913**	0.7614
平均	0.7630	**0.7873**	0.7647

領域適応の手法を用いない v_b から正解率は改善されているが，feature based な利用と比べると大きく劣っている．これは文書の特徴ベクトルとして [CLS] の埋め込み表現を利用しているからだと考えている．本論文で行ったように，単語埋め込み表現列全体から文書の特徴ベクトルを構築し，そこから fine tuning することも可能である．今後はそれも試したい．

6. おわりに

本論文では感情分析の教師なし領域適応に対して，BERT の feature based な利用を試みた．その際に BERT の出力の最上位層の単語埋め込み表現列を用いるのではなく，その 1 つ下の階層の単語埋め込み表現列を用いることを提案した．Amazon データセットを利用した領域適応の実験では，半数の領域適応では効果があった．ただし，全体の平均でみるとわずかに，標準的な最上位層の単語埋め込み表現列を用いる手法よりも劣った．また感情分析の教師なし領域適応に対しては，BERT の feature based な利用法が有効であることも確認できた．今後は最上位層の単語埋め込み表現列と下位の層の単語埋め込み表現列を併用する手法を考えていきたい．また BERT の fine tuning の利用からも，感情分析の教師なし領域適応を試したい．

参考文献

[1] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv preprint arXiv:1810.04805* (2018).
[2] Howard, J. and Ruder, S.: Universal Language Model Fine-tuning for Text Classification, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339 (2018).
[3] Lau, J. H. and Baldwin, T.: An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation, *arXiv preprint arXiv:1607.05368* (2016).
[4] Pan, S. J. and Yang, Q.: A survey on transfer learning, *Knowledge and Data Engineering, IEEE Transactions on*, Vol. 22, No. 10, pp. 1345–1359 (2010).
[5] Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L.: Deep Contextualized Word Representations, *NAACL-2018*, pp. 2227–2237 (2018).
[6] Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I.: Improving language understanding by generative pre-training, *Technical report, OpenAI.* (2018).
[7] Ruder, S.: Neural Transfer Learning for Natural Language Processing, PhD Thesis, National University of Ireland, Galway (2019).
[8] Shinnou, H., Zhao, X. and Komiya, K.: Domain Adaptation Using a Combination of Multiple Embeddings, *PACLIC-32* (2018).
[9] Søgaard, A.: *Semi-Supervised Learning and Domain Adaptation in Natural Language Processing*, Morgan & Claypool (2013).
[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I.: Attention is all you need, *Advances in neural information processing systems*, pp. 5998–6008 (2017).
[11] 新納浩幸，佐々木稔：k 近傍法とトピックモデルを利用した語義曖昧性解消の領域適応，自然言語処理，Vol. 20, No. 5, pp. 707–726 (2013).
[12] 新納浩幸，浅原正幸，古宮嘉那子，佐々木稔：nwjc2vec: 国語研日本語ウェブコーパスから構築した単語の分散表現データ，自然言語処理，Vol. 24, No. 5, pp. 705–720 (2017).