論文読みまとめ
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

井上智裕

2021 年 1 月 6 日

1 導入

1.1 扱う問題

言語モデルの事前学習（自然言語処理）

1.2 問題意識

自然言語処理タスクの改善には、言語モデルの事前学習が有効であることが示されている。事前学習された言語モデルを下流のタスクに適用する方法として特徴量ベースのアプローチとファインチューニングによるアプローチがある。特徴量ベースのアプローチは事前学習により得られた表現をタスク固有のアーキテクチャにおいて特徴量の一部として扱う方法である。単語埋め込みベクトルの事前学習にはこのアプローチがとられ、文の埋め込みや段落の埋め込みに対してもその手法が一般化されてきた。一方で、ファインチューニングは事前学習によって得られたパラメタを初期値としてタスクに対して学習させる方法である。これらの方法では、共通して事前学習において一方向性の言語モデルを使用するため、文レベルのタスク（文間の関係を予測するための自然言語推論や言い換えなど）や質問応答のようなトークンレベルのタスクなど、前後両方向の文脈情報が重要であるタスクに適さないという問題がある。

既存手法についてより具体的に言及する。ELMo という手法は左から右、右から左への言語表現を連結することで各トークンの表現を得る手法であるが、これは教師なし特徴量ベースのアプローチであり、深い双方向性は持たない。また、近年の手法として、文脈を考慮して教師なしで事前学習した文または文書エンコーダーを教師ありの下流タスクにファインチューニングするアプローチが取られてきた。この手法ではゼロから学習するパラメタが少ないという利点があるが、事前学習に使用される言語モデルは一方向性のものである。OpenAI GPT ではこの手法が取られており、以前に最先端の結果を多くの文レベルタスクで残している。この他に、自然言語推論や機械翻訳など大規模なデータセットを用いた教師ありタスクでは、転移学習が有効であることが示されている。

2 理論

この論文では、ファインチューニングによるアプローチの改善として BERT (Bidirectional Encoder Representations from Transformers) を提案している。BERT モデルは、ラベル付けされていないデータを用いて複数のタスクで事前学習が行われ、下流タスクのラベル付きデータを用いてファインチューニングされる。つまり、BERT は下流タスクが異なっていても統一されたアーキテクチャであるという特徴を持つ。

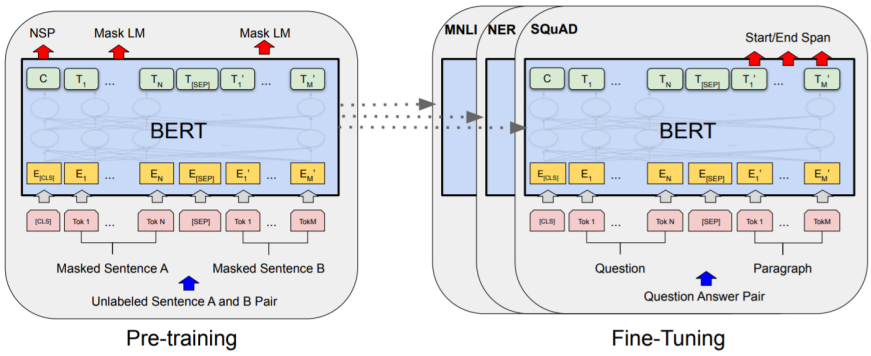

図 2.1　BERT の事前学習とファインチューニング

2.1　モデル構造

BERT のアーキテクチャは多層双方向 Transformer のエンコーダーである。Transformer のブロック数を L、隠れ層のサイズを H、self-attention ヘッドの数を A として、この論文で提案されたモデルは $\text{BERT}_{\text{BASE}}$ (L=12, H=768, A=12, パラメタ総数=1.1 億個) と $\text{BERT}_{\text{LARGE}}$ (L=24, H=1024, A=16, パラメタ総数=3.4 億個) である。モデルへの入力列は単一文の場合も複数の文の場合もあるため、最初の文頭を表す [CLS] トークンと文同士を分離する [SEP] トークンが挿入される。入力表現はトークン、セグメントおよび位置の埋め込み表現を加算されることで得られる。

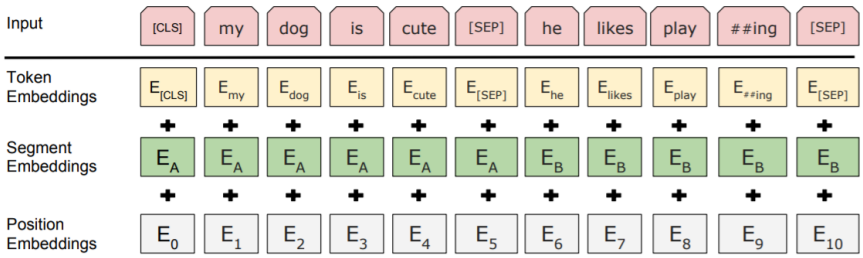

図 2.2　BERT の入力表現

2.2　事前学習

BERT では、従来の左から右への一方向の言語モデルを用いない。代わりに masked language model (MLM) と next sentence prediction (NSP) の 2 つの教師なしをタスクを事前学習に導入する。

MLM は、深い双方向表現の学習のために入力トークンの一部（この論文では 15%）をランダムにマスクして、そのマスクされたトークンを予測するタスクである。これにより双方向の事前学習モデルを得ることができるが、[MASK] トークンはファインチューニングの際には現れないため、事前学習とのミスマッチが生じるという欠点がある。これを軽減するために常に [MASK] トークンに置き換えるのではなく、選択されたトー

クンのうち 80% を [MASK] トークンに置き換え、10% をランダムなトークンに置き換え、残りの 10% を変更前のトークンに戻す。

NSP は、言語のモデル化では直接捉えることのできない 2 つの文の関係性を考慮するモデルの構築のために行われる次文予測タスクである。具体的にはある文 A に対して、別のある文 B が実際に続く次の文かそうでないのかを二値で予測するタスクである。

2.3 ファインチューニング

Transformer の self-attention 機構により、BERT は適切な入出力を入れ替えることができるため、ファインチューニングは容易である。すなわち、ファインチューニングにおいては各タスク固有の入出力を BERT に与えるだけで良い。

3 実験

どんな実験設定か？どんな点で他の手法より良くなったか？

3.1 GLUE

General Language Understanding Evaluation (GLUE) は、8 つの自然言語理解タスクの集合体である。32 のバッチサイズを使用し、すべての GLUE タスクについて、データ上で 3 エポック分のファインチューニングを行った。また、各タスクについて、Dev セット上で最適な学習率（5e-5, 4e-5, 3e-5, 2e-5 のいずれか）を選択した。結果は図 3.1 のようになった。$\text{BERT}_{\text{BASE}}$ および $\text{BERT}_{\text{LARGE}}$ の両方がすべてのタスクにおいて従来のシステムを大幅に上回る結果となった。

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

図 3.1 GLUE の結果

3.2 SQuAD v1.1

Stanford Question Answering Dataset (SQuAD v1.1) は、クラウドソーシングされた質問と回答のペアの集合である。質問とその答えを含む Wikipedia の一節が与えられた場合、答えに相当するテキストスパンを予測するタスクである。学習率は 5e-5、バッチサイズは 32 で、3 エポックでファインチューニングを行った。なお、SQuAD でのファインチューニングに先立ち、最初に TriviaQA でファインチューニングを行った。結果は図 3.2 のようになった。F1 スコアで、アンサンブルでは +1.5、シングルモデルでは +1.3 トップリーダーボードシステムを上回った。

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

図 3.2　SQuAD v1.1 の結果

3.3　SQuAD v2.0

SQuAD 2.0 タスクは、SQuAD 1.1 に段落に答えが存在しないという選択肢を追加したものである。学習率 5e-5、バッチサイズ 48 で 2 エポックのファインチューニングを行った。結果は図 3.3 のようになった。従来の SoTA モデルと比較して、F1 スコアで +5.1 の改善が見られた。

System	Dev		Test	
	EM	F1	EM	F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

図 3.3　SQuAD v2.0 の結果

3.4　SWAG

Situations With Adversarial Generations (SWAG) には、根拠のある常識的な推論を行うタスクであり、与えられた文に対して続く文として、4 つの選択肢の中から最も妥当なものを選択するタスクである。学習率 2e-5、バッチサイズ 16 の 3 エポックでモデルをファインチューニングした。結果は図 3.4 のようになった。$\text{BERT}_{\text{LARGE}}$ は ESIM+ELMo システムを +27.1%、OpenAI GPT を +8.3% 上回った。

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

図 3.4　SWAG の結果

3.5　アブレーションスタディ：事前学習タスクによる影響

BERT の深い双方向性の重要性を確かめるために事前学習の条件を変える実験を行った。1 つ目として NSP タスクなしで MLM タスクのみを行うモデル、2 つ目として NSP タスクなしで MLM タスクの代わりに標準的な LTR (Left-to-Right) 言語モデルを用いて学習される左コンテクストのみのモデル、3 つ目として LTR モデルに BiLSTM を加えたモデルに対して、事前学習を行った。結果は図 3.5 のようになった。これより、NSP タスクの除外によって QNLI、 MNLI、 SQuAD 1.1 において性能が著しく低下することが確認できる。また、MLM による双方向性表現の学習をなくすことで、全てのタスクにおいて性能は低下し、MRPC と SQuAD で大きな低下が見られた。また、LTR モデルに BiLSTM を加えたモデルでは SQuAD では改善が見られるが、GLUE では性能が低下した。

Tasks	Dev Set				
	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
$\text{BERT}_{\text{BASE}}$	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

図 3.5　事前学習タスクによる影響

3.6　アブレーションスタディ：モデルサイズによる影響

BERT モデルの構造のうち、Transformer のブロック数 L、隠れ層のサイズ H、self-attention ヘッドの数 A の影響を調べる。パラメタを変更したときの GLUE タスクでの結果は図 3.6 のようになった。これより、下流タスクのデータが少ない MRPC のようなタスクに対しても、大きなモデルほど高い精度を示すことが確認できる。また、モデルサイズを大きくすることで、機械翻訳や言語モデリングのような大規模なタスクだけではなく小さなスケールのタスクにおいても改善が見られることが確認できた。

Hyperparams				Dev Set Accuracy		
#L	#H	#A	LM (ppl)	MNLI-m	MRPC	SST-2
3	768	12	5.84	77.9	79.8	88.4
6	768	3	5.24	80.6	82.2	90.7
6	768	12	4.68	81.9	84.8	91.3
12	768	12	3.99	84.4	86.7	92.9
12	1024	16	3.54	85.7	86.9	93.3
24	1024	16	3.23	86.6	87.8	93.7

図 3.6　モデルサイズによる影響

3.7　アブレーションスタディ：BERT を用いた特徴量ベースの手法

ここまでの BERT の結果はすべて、事前学習されたモデルに単純な分類層を追加し、下流のタスクですべてのパラメータを共同で微調整するというファインチューニングアプローチを使用していた。しかし、事前学習されたモデルから固定の特徴量を抽出する特徴量ベースのアプローチには、一定の利点がある。すなわち、

第一にすべてのタスクが Transformer エンコーダアーキテクチャで簡単に表現できるわけではないため、そういった場合に対してタスク固有のモデルアーキテクチャを追加することで対応できること、第二に計算コストの高い表現計算を事前に行い、その表現を用いて計算コストの低い実験を多く行うことで計算コストを低減できることである。ここでは、固有表現抽出タスクである CoNLL-2003 NER タスクに BERT を適用して、2 つのアプローチを比較する。結果は図 3.7 のようになった。特徴量ベースのアプローチでもっとも性能の良いものとファインチューニングアプローチを比較すると、その差はわずかに F1 スコアで 0.3 であり、BERT がファインチューニングと特徴量ベースのアプローチの両方で有効であることが確認できる。

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

図 3.7　BERT によるファインチューニングアプローチと特徴量ベースアプローチの比較

4　結論・展望

近年、言語モデルにおいて転移学習が行われるようになり、教師なしの事前学習が多くの言語理解システムにおいて不可欠であることが実証されてきた。特に、低リソースのタスクであっても、事前学習を行ったモデルを利用することで深い一方向性アーキテクチャの恩恵を受けることを可能にしてきた。この論文は、深い双方向性を持ったモデルにおいても共通の事前学習を行ったモデルが幅広いタスクに適用できることを示している。