第2回ミーティング
Transformer, BERT の追加調査，モデル比較

小池 正基

2023 年 3 月 25 日

1　はじめに

今回は自然言語処理モデルの Transformer,BERT について理解し，日本語の NLP モデルのより詳しい調査を行うことを目的とする．

2　Transformer

Transformer [1] とは，Encoder-Decoder 内の RNN を Self-Attention に変更したモデルである．ネットワーク全体が Self-Attention と Feed Forward Network のみで構成されているため，並列計算が容易であり，RNN より高速で学習可能である．図 1 に Transformer モデルの構造を簡略化した図を示す．今回は，Attention と Feed Forward について，より厳密に説明する．

2.1　Attention

Attentionn とは，文章内の単語同士の内積によって単語の重要度を算出する計算である．入力は入力単語 $EmbeddingX$ にそれぞれの重み W を掛け合わせた分散表現である Query，Key，重み Value で構成される．Query と Key の内積によりその関連度を計算し，Query と Key の次元数の平方根 $\sqrt{d_k}$ で割った後に softmax 関数を適用することで求められる．実際には，一連の Query に対して Attention を

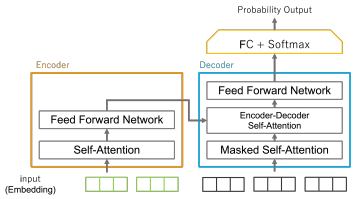

図 1: Transformer の構造

同時に計算し，行列 Q にまとめている．以下に実際の式を示す．

$$\begin{aligned} Attention(Q,K,V) &= softmax(\frac{QK^T}{\sqrt{d_k}})V \\ Q &= XW^Q \\ K &= XW^K \\ V &= XW^V \end{aligned} \tag{1}$$

このとき，$\sqrt{d_k}$ でスケールしているのは，Query と Key が過剰に大きくなることで性能が下がるのを防ぐためである．

2.2 Multi-Head Attention

Multi-Head Attention とは，各単語に対して 1 組の Query，Key,Value を持たせるのではなく，比較的小さい Query,Key,Value を複数個用意し，それぞれで Attention の計算を行う方法である．最終的にそれらを 1 つのベクトルに纏めたものをその単語の潜在表現とする．

$$\begin{aligned} MultiHead(Q,K,V) &= Concat(head_1,...,head_h)W^O \\ where head_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{2}$$

このように Attention を複数回に分割することで，異なる潜在表現から有益な情報を集めることができる．

2.3 Feed Forward Network

Attention の計算を行なって得られた分散表現を Feed Forward Network に通し，順伝搬を行う．順伝搬は各単語ごとに独立して 2 層のニューラルネットワークで構成される．しかし，重み W は共有パラメータである．

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{3}$$

以上の手順を終え出力された単語分散表現は，softmax によって一番高い確率を示した単語を出力する．

3 Bidirectional Encoder Representations from Transformers (BERT)

Bidirectional Encoder Representations from Transformers (BERT) [2] は，Transformer のエンコーダを用いた事前学習モデルである．あらゆる NLP タスクに fine-tuning 可能な高性能なモデルである．図 2 に BERT の構造例を示す．BERT は事前学習に注力した構造になっており，両方向からの学習を行わせるため，文章内でマスクされた単語を予測するタスクである Masked Language Model (MLM) と，入力された 2 つの文章が連続か予測するタスクである Next Sentence Prediction (NSP) という 2 つのタスクによって学習する．

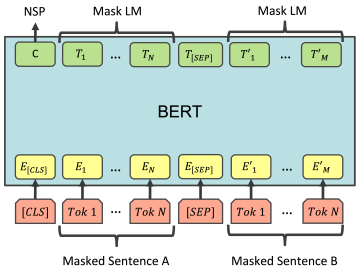

図 2: BERT の構造

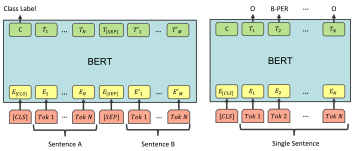

図 3: BERT による fine-tuning

3.1 事前学習

BERT の事前学習で行う 2 つのタスクについて詳しく説明する．Masked Language Model (MLM) は，入力される単語の 15%を [Mask] トークンで置換し，元のトークンを推測するタスクである．このとき，fine-tuning 時には [Mask] トークンが無いことを考慮し，一定の確率で [Mask] トークンではなく，他のランダムなトークンに置換している．

Next Sentence Prediction (NSP) は，2 文の入力が連続したタスクかを推察するタスクである．このタスクを行うときは，専用の出力 C を用いて予測する．

3.2 fine-tuning

BERT で fine-tuning を行う場合．適切なタスク固有の入出力を接続するだけで可能である．図 3 に BERT による fine-tuning の実行例を示す．例えば感情分析などの識別タスクを行う場合は入力は一つの文章，出力は C に接続することで fine-tuning でき，文章の品詞解析などのトークンレベルのタスクを行う場合は入力は一つの文章，出力は T に接続することで fine-tuning できる．

4 日本語 NLP モデルの比較

前回の調査では必要な情報が不十分であったため，再調査を行う．使用するモデルを評価するにあたって，モデルの最終更新日，性能評価を考慮する．性能評価には日本語 NLP モデルのベンチマークである JGLUE を採用した．

性能が高いモデルを候補として表 4 に示す．表表 4 より，早稲田大学 RoBERTa が高スコアを記録しており，文書分類タスクである MARC -ja や自然言語推論タスクである JNLI のスコアが高いため，言語理解が必要な当研究に最適だろうと考える．

Model	Date	MARC-ja	JSTS	JNLI	JCommonsenceQA
早大 RoBERTa	2022/10/15	0.969	0.890	0.928	0.900
XLM RoBERTa	2023/04/07	0.964	0.918/0.884	0.919	0.840
日本語 LUKE	2022/11/09	0.965	0.932/0.902	0.927	0.893
日本語 DeBERTa V2	2023/03/18	0.968	0.892	0.919	0.890

5 おわりに

今回は，自然言語処理において特に重要な Transformer,BERT の論文内容の解説と，日本語 NLP モデルの再検討を行った．今後は既存のモデルを利用して，自由記述文から知能レベルを推定するプログラムの作成を行う．

参考文献

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, page 5998–6008, 2017.

[2] 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 (Long and Short Papers)*, pages 3213–3223, 2019.