# 【別添資料２】BERT を用いた予測モデルの学習実験

## 1. はじめに

分類器の精度向上のためには、付帯情報の各項目に対する記載の有無だけでなく、その意味内容を用いることが重要である。本研究では全ての手法に共通して機械学習用の共通ベクトル（年齢、性別、使用された ICD-10 コード、各付帯情報の項目の記載の有無）を用いているが、これに「付帯情報の文字列の意味内容」のベクトル表現を加えることにより、精度の向上が期待できる。

本実験では、近年深層学習ベースの自然言語処理の各種タスクにおいて従来手法の精度に優ったことから注目されている言語モデル獲得手法である BERT を用いて予測精度が向上するか試行を行った。

## 2. BERT とは

BERT(Bidirectional Encoder Representations from Transformers)は 2018 年に Google から発表された汎用言語モデル獲得のための手法である。Transformer という要素技術を元にしており、特定の領域のコーパス（大量の文章ソース）から別途の正解づけを必要とせず自動的に言語モデルが学習できることが大きな特徴である。このモデルは多くの自然言語処理に汎用的に用いることができるモデルであると言われており、一度獲得した BERT モデルを他のタスクに転用することが可能である。

そのため、一般的な使い方としては、何らかのコーパスにて学習された BERT モデルを（必要に応じてさらに追加で事前学習を行い）、その上で特定のタスクを解くための Deep Neural Network アーキテクチャに組み込んでファインチューニングする（重みを変更する）という方法が用いられる。本研究でもこれら一般的な方法に則って行った。

## 3. 対象データ

本実験では比較のために、昨年度ベースラインモデルとして開発した**「使用された病名 ICD-10 コードと、付帯情報の項目の有無」だけから仮原死因の変更の有無を XGBoost を用いて予測するモデル**の学習に用いたものと同じデータを用いた。

これは、まず突合 DB からランダムに抽出された 50 万件のうち、「全ての病名に ICD コードが付与でき、IRIS 処理が完了したもの」320,008 件を対象とし、さらに「何らかの付帯情報が存在するもの」81,688 件を実験用に抽出したデータである。

すでに昨年度、「年齢、性別、使用された ICD-10 コード（1553 列）、各付帯情報（２４列）」の計 1577 個の変数（**共通ベクトル**）を用いて、「確定原死因は、IRIS が決定し

た仮原死因から変更があるか否か」を勾配ブースティング決定木の一種である XGBoost を用いて学習し、付帯情報の意味内容を用いていないにも関わらず、Accuracy 90.3% を実現している。本年度では、付帯情報の意味内容まで考慮することで、この精度をさらに向上させるべく、前述の BERT モデルを用いた手法の検討を行った。

## 4. 用いた BERT ベースの学習手法

BERT によって言語モデルを獲得するためには、ドキュメント内に複数文書が存在する、という形のコーパスが必要である。例えば Wikipedia における日本語解説記事などが該当する。これは BERT の学習が、2 つの文章を入力として与え、マスクされた語を復元するタスクと共に、2 文の前後関係を解く、というタスクを同時に学習させる仕組みであることによるものである。

当初、突合 DB における「付帯情報の文字列」を用いて BERT モデルを学習させることを考えたが、死亡個票内の付帯情報は短い記述が非常に多く、複数文章が書かれているケースが十分に集まらないため、突合 DB を用いた BERT モデルの学習は断念し、既に一般に公開されている日本語 Wikipedia から学習済みである BERT モデルをそのまま利用することとした。日本語 Wikipedia は広く一般的な文章を含んでいるため、元となる言語モデルとしては、適切と考えられた。

本研究で用いた DNN（Deep Neural Network）のアーキテクチャは、**「共通ベクトル(1577 次元)」と「BERT モデルに付帯情報の文字列を通した結果得られるベクトル(768 次元)」を上位の全結合層で統合し、最終的に 2 値分類の結果を得る**もので、他のタスクでも非常に良く用いられるモデルである。しかし、実際の学習モデルの構築上は複数の考慮点がある。

1） 安定した結果を得るために学習率をどのように変化させるか
   - これについては学習率の減衰方法を複数施行した。

2） インバランスデータへの対応
   - 「1：変更あり」が「0：変更なし」に比べて圧倒的に多いインバランスデータであることから、データオーギュメンテーション、あるいは Class Weight を用いた重みづけ学習を試行した。

これらを踏まえ、本研究では 8 種類の手法を試し、結果を比較した。
以下にその詳細を記す。

### A） RedLR

- 学習データはそのまま使用
- 学習率を 0.0006 で開始し、2 エポック改善が無ければ学習率を 1/10 にする。

### B） BalRedLR

- データオーギュメンテーションを行ったもの。

学習データのうち結果が 1(変更あり)のデータを複製し、結果が 0(変更なし)のデータと同数にしている。

- 学習率を 0.0006 で開始し、2 エポック改善が無ければ学習率を 1/10 にします。

C) BalRedLR_BertOnly

- 共通ベクトルを用いず、付帯情報の文字列だけから、BERT のみで変更予測モデルを作成したもの。(付帯情報文字列以外の 1577 次元の共通ベクトルを削除)
- 学習データのうち結果が 1 のデータを複製し、結果が 0 のデータと同数にしている。
- 学習率を 0.0006 で開始し、2 エポック改善が無ければ学習率を 1/10 にする。

D) BalRedLR_Norm

BERT からの出力結果テンソルを BatchNormalization で正規化した後、共通ベクトル（付帯情報以外の 1577 次元）と結合したもの。

- 学習データのうち結果が 1 のデータを複製し、結果が 0 のデータと同数にしています。
- 学習率を 0.0006 で開始し、2 エポック改善が無ければ学習率を 1/10 にします。

E) CwRedLR

- 学習データはそのまま使用
- インバランスへの対応のため、データオーギュメンテーションを用いるのではなく、class_weight を指定し、学習時のクラスごとの重みづけを変更したもの。({0: 1.0、 1: 4.3})
- 学習率を 0.0008 で開始し、2 エポック改善が無ければ学習率を 1/10 する。

F) CyLR

- 学習データはそのまま使用。
- 学習率を 0.0001 から 0.0008 までの範囲で上下させ、サイクルさせたもの。

G) BalCyLR

- 学習データのうち結果が 1 のデータを複製し、結果が 0 のデータと同数にしています。
- 学習率を 0.0001 から 0.0008 までの範囲で上下させ、サイクルさせています。

H) CwCyLR

- 学習データはそのまま使用。
- class_weight を指定し、学習時のクラスごとの重みづけを変更。
  ({0: 1.0、 1: 4.3})
- 学習率を 0.0001 から 0.0008 までの範囲で上下させ、サイクルさせたもの。

## 5. 結果

上記 8 種類の手法での「仮原死因の変更有無」の予測結果を下記表 1 に示す。
（本報告書中の**表 E** と同じである）

**表 1：各学習手法の予測精度一覧**

| 手法 | | RedLR | BalRedLR | BalRedLR_BertOnly | BalRedLR_Norm | CwRedLR | CyLR | BalCyLR | CwCyLR |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy | | **0.914** | 0.866 | 0.819 | 0.863 | 0.877 | 0.900 | 0.858 | 0.882 |
| F1-score | | **0.591** | 0.535 | 0.391 | 0.548 | 0.543 | 0.564 | 0.510 | 0.549 |
| Precision | | **0.756** | 0.481 | 0.343 | 0.475 | 0.517 | 0.641 | 0.457 | 0.537 |
| Recall | | 0.485 | 0.602 | 0.455 | **0.648** | 0.571 | 0.504 | 0.576 | 0.561 |
| Confusion Matrix (正解, 予測) | (0, 0) | 13918 | 12890 | 12422 | 12750 | 13130 | 13656 | 12817 | 13235 |
| | (0, 1) | 328 | 1356 | 1824 | 1496 | 1116 | 590 | 1429 | 1011 |
| | (1, 0) | 1078 | 833 | 1140 | 737 | 897 | 1038 | 888 | 918 |
| | (1, 1) | 1014 | 1259 | 952 | 1355 | 1195 | 1054 | 1204 | 1174 |

## 6. 考察とまとめ

当初の予想では､BERT モデルによって｢付帯情報の文字列の意味｣を分散表現に変換し、上位層で共通ベクトルと合わせることで、共通ベクトル単体での学習に比べ大幅に精度が向上すると考えていたが、予想に反して昨年度のベースライン手法 （共通ベクトル単体での XGBoost、Accuracy 90.3%）と大差ない結果となった。

唯一 RedLR は Accuracy がベースラインに比べ 1%程度向上したが、後述する他の手法（TFIDF, LSI, WORD2VEC, DOC2VEC 等）での分散表現埋め込み（Embedding）を用いた方が効果が高く、期待した程ではない。また RedLR 以外の手法は全て昨年度のベースライン手法を下回る成績となっている。**このことから、以降の実験では、BERT 以外の方法で本実験を行うこととした**（XGBoost-Embed の各手法に絞った)。

また BalRedLR_BerOnly が示す通り､「共通ベクトル」の情報を全く使わず、付帯情報の文字列だけから判断したものは大幅に精度が低下していたことから、共通ベクトルの重要性が明らかとなった。

尚、前述の通り、死亡個票に書かれている付帯情報（手術の部位及び所見、解剖の部位及び所見、手段及び状況、その他付言すべき事柄、備考欄等）の文字列は短い記述が非常に多く、文章読解よりも寧ろ、特定の単語の出現の有無を端的に検出する方が有効である可能性が高い。そのため今回のタスクでは BERT の有効性が十分に発揮できなかったと考えられた。