障害レポートの分類問題に対するデータ選択を用いた BERT モデルの精度向上

勝又 智[1]　小町 守[1]　真鍋 章[2]　谷本 恒野[2]

[1] 首都大学東京 [2] 富士電機株式会社

katsumata-satoru@ed.tmu.ac.jp, komachi@tmu.ac.jp,
{manabe-akira, tanimoto-kouya}@fujielectric.com

1　はじめに

近年，BERT [1] を用いた研究が盛んに行われている．BERT は Transformer を元に，大規模な単言語データを使用して事前学習を行い，その後，解きたいタスクで fine-tuning を行う．Devlin ら [1] は BERT を用いることで英語の分類問題に対して高い性能を得ることを報告している．日本語に関する研究に関しては，柴田ら [7] が BERT を使うことで構文解析の性能が向上することを報告している．

大規模日本語データとして Wikipedia と BCCWJ [4] が知られており，柴田ら [7] の事前学習済み BERT は Wikipedia データを元に事前学習を行っている．本研究は，日本語の障害レポートの分類問題に対して BERT の適用を行う．障害レポートは特定の分野に対しての記述であり，Wikipedia データとは大きく分野が異なる．図 1 に，Wikipedia で学習した BERT の Masked Lanuguage Model（MLM）に対する loss を示す．BCCWJ の各分野のデータと比較して，障害レポートの loss は大きいことがわかる．

本研究は，単言語データから障害レポートの分野に類似するデータの選択を行い，選択されたデータを用いて BERT の追加学習を行うことで，事前学習モデルと下流タスクで扱うデータ間の分野適用を行った．本研究が対象とする障害レポートにはラベルありデータは少なく，対象分野のラベルなしデータも即座に用意するのは権利などの問題で難しい．そのため，追加学習には一般に使用可能なラベルなしデータを使用した．

Wikipedia を用いた事前学習済み BERT に対して，選択されたデータを用いて分野適応を行い，障害レポートに対して分類実験を行った結果，分野適応を行わなかった場合と比較して，性能向上が確認された．また，BERT の追加学習による結果，対象分野に対する MLM の loss が減少し，その減少と分類精度の間で，相関が確認された．

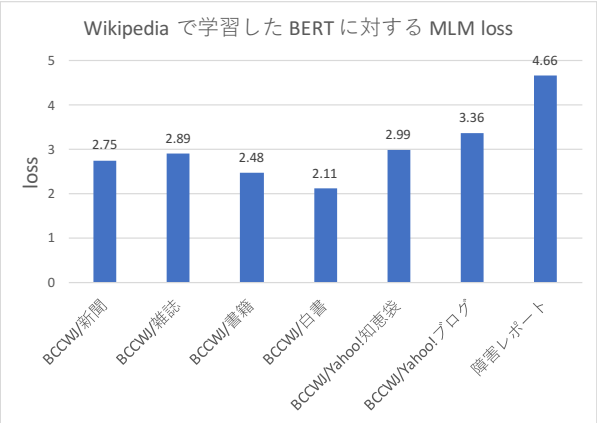

図 1: Wikipedia で学習した BERT に対する各データの MLM の loss．左から BCCWJ のコアデータ，障害レポートである．

2　関連研究

2.1　BERT

BERT [1] の学習は，大規模単言語データを用いて事前学習を行い，その後，解きたいタスクに対して fine-tuning を行う．Devlin ら [1] は事前学習データとして Wikipedia と BookCorpus を用いている．このときの事前学習の目的関数は，Masked Language Model（MLM）と Next Sentence Prediction（NSP）を用いる．これらの目的関数は文書単位の単言語データに対して学習が可能である．BERT を用いた研究は，解きたいタスクが分類問題の場合，入力が [CLS] に対応する BERT の出力 $h_{[CLS]}$ を用いてクラスを推定することが多い [1, 6]．Devlin ら [1] や Sun ら [6] は分類タスクにおいてこのように BERT を用いることで高い性能が得られることを報告している．日本語の BERT の事前学習は Wikipedia で学習したもの

表 1: 障害レポートの各文と対応するラベルの例

障害レポートの各文	状況	原因	措置	その他
給水サンプリング 装置 pH 計指示値不良	✓			
製品不具合				✓
原因は KCL 液の不適下, KCL 溶液箱の蓋は 完全閉しないこと。		✓	✓	

[7][1]，Twitter で学習したもの[2]が公開されている．

Han and Eisenstein [2] は英語の系列ラベリングにおいて，BERT の分野適応を行い，性能向上を報告している．彼らは対象分野として歴史的な英語と Twitter の 2 種類を調査しており，分野適応として，解きたいタスクのラベルなしデータを用いて，事前学習済みの BERT に追加学習し，その後 fine-tuning を行っている．本研究は，彼らと違い対象分野のラベルなしデータを用意することが難しい状況を想定している．そのため，事前学習済みの BERT に対して，最終的に解きたいタスクのデータに類似した分野のデータを見つけ，それを用いて BERT を追加学習する．

2.2 データ選択

類似分野のデータ選択手法として Moore and Lewis [5] による N-gram LM を用いた手法が知られている．この手法は一般分野と目的となる分野に対して，一般分野 LM N と目的分野 LM I を作成する．次に，これらの LM（$\mathrm{LM}_{\mathrm{model}} \in \{I, N\}$）から単言語データの文 s に対してエントロピー H を求める．その後，このエントロピーの差（式 1）を文 s に対して計算し，この値が大きいものから順に目的となる分野に類似しているとしてデータ選択を行う．

$$H(s|I) - H(s|N) \tag{1}$$

$$H(s|\mathrm{LM}_{\mathrm{model}}) = \frac{1}{|s|} \sum_{i=1}^{|s|} \log P_{\mathrm{LM}_{\mathrm{model}}}(s_i)$$

ただし，$|s|$ は文長である．

本研究も同様にして，単言語データの各文に対して，単言語データで学習した LM と目的分野で学習した LM からエントロピーの差分を計算する．そして，値の大きい文から順に追加学習に使用する．

[1]柴田ら [7] の他にも https://github.com/yoheikikuta/bert-japanese などが存在する．

[2]https://github.com/hottolink/hottoSNS-bert

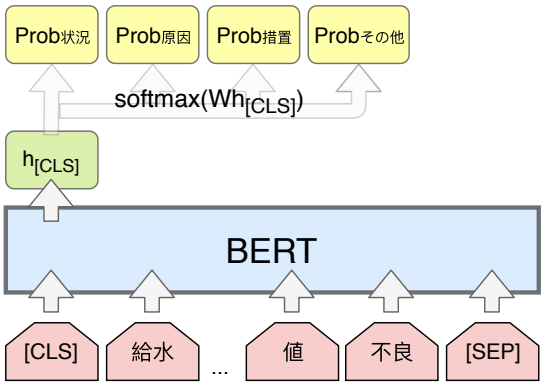

図 2: BERT を利用したマルチラベル分類モデル．

3 障害レポートに対する分野適応

3.1 障害レポートの分類問題

本研究は障害レポートの各文に対してマルチラベル分類を行う．具体的には，障害レポートの各文が**故障状況**，**故障原因**，**措置対策**，**その他**に属する 4 ラベル分類を行う．表 1 に実際の障害レポートの例を示す．表 1 は障害レポートの各文に対して，故障状況，故障原因，措置対策，その他のラベルを付与した例である．1 行目の文に対しては故障状況のラベルを，2 行目の文に対してはその他のラベルが付与されている．一方で，3 行目の文に対しては故障原因と措置対策の 2 つのラベルが付与されている．このように，本研究で扱うデータはマルチラベルデータである．

本研究では，マルチラベル分類として One-vs-Rest を用いた．分類モデルとしては図 2 のように，BERT を用いた．入力 x に対して，以下の数式のように BERT の出力 $h_{[CLS]}$ を用いて，各ラベルごとに付与するかしないかの確率 $P_{label}(y|x)$ を求める．この確率を用いて，各ラベルごとに 2 値分類を行い，入力 x に対して各ラベルを付与するかどうか（$y \in \{\mathrm{positive}, \mathrm{negative}\}$）を予測する．

$$P_{label}(y|x) = \mathrm{softmax}(W h_{[CLS]}[t : t+2]) \tag{2}$$

$$\mathrm{Predict}_{label} = \arg\max_{y}(P_{label}(y|x)) \tag{3}$$

$$t = \begin{cases} 0 & (label = \text{故障状況}) \\ 2 & (label = \text{故障原因}) \\ 4 & (label = \text{措置対策}) \\ 6 & (label = \text{その他}) \end{cases}$$

ただし，$W \in \mathbb{R}^{2C \times n}$ はこの分類で学習されるパラメータであり，C がクラス数（本研究では 4），n が隠れ層の次元数である．

3.2 データ選択を用いた分野適応

本研究は，以下の手順で学習を行う．

1. 単言語データで BERT の事前学習を行う．(Base)
2. データ選択を行い，得られたデータから BERT の追加学習（MLM）を行う．
3. 追加学習された BERT を用いて，マルチラベル分類を学習する．

ただし，BERT の事前学習については公開されているモデルを使用する．

本研究では，BERT の追加学習を行う際，目的関数として MLM のみ用いる．これはデータ選択の結果得られるデータが文のランキングであり，文章の構造になっておらず，NSP の学習を行うことができないためである．

目的分野の LM を訓練する際に，分類の学習で使用する学習データのみを用いた．また，データ選択の際に使用する文数について，いくつかの値で追加学習を行い，分類の学習で使用する開発データの MLM に対する loss が最も低いモデルを分類の学習に使用した．

4 実験

4.1 実験設定

本研究では，火力発電に関する障害レポートに対して実験を行う．この障害レポートは 961 件存在し，そのレポート内の総文数は 6,439 文である．この障害レポートを分割し，学習データに 5,205 文，開発データに 580 文，評価データに 654 文使用した．これらのデータおよび単言語データの単語分割には Juman++ (2.0.0-rc2)[3]を使用した．

データ選択に使用する単言語データとして，Wikipedia と BCCWJ をそれぞれ使用した．さらに，データ選択の際に使用する文数は $\{200K, 400K, 600K, 800K, 1M, 2M, 4M, 6M\}$ で探索した[4]．これらの値の中から，開発データに対する MLM の loss の値に基づいて最も良い追加学習モデルを決定した．また，追加学習はバッチサイズ 32 で行い，6M モデルが 1 エポック学習できるようにするため，全てのモデルで 190K イテレーション学習した．この場合，1M は同じ文を約 6 回学習に使用することになる．このように，同一文を追加学習時に使用する際は，異なるマスクを使用するようにしている．

[3]https://github.com/ku-nlp/jumanpp

[4]BCCWJ については総文数が 6M に満たないため，代わりに全ての文を使用した．

表 2: 追加学習時の開発データに対する MLM の loss と F-score の平均．loss 内の太字が BCCWJ と Wikipedia でそれぞれ最も低い値を指し，下線部が全てのパラメータの中で最も低い値を指している．

	MLM loss		平均 F	
	BCCWJ	Wikipedia	BCCWJ	Wikipedia
200K	3.85	3.55	85.20	85.50
400K	3.67	3.41	85.88	86.69
600K	3.52	3.38	85.99	86.62
800K	**3.51**	**3.31**	85.50	86.36
1M	3.53	3.36	85.30	85.94
2M	3.56	3.39	**86.62**	86.65
4M	3.64	3.45	86.22	**86.94**
6M	3.69	3.47	86.40	86.47
Baseline		7.00		85.02
Random		3.71		85.49

本研究では，Base に対して分類を学習したモデルを Baseline とする．さらに，開発データから決定したモデルが追加学習の際に使用した文数と同じ数だけ Wikipedia からランダムに文を選び，これらを用いて Base に追加学習を行い，分類を学習したモデル Random も比較手法として実験した．

実験に使用する BERT の事前学習モデルは柴田ら [7] の公開しているものを使用した．追加学習には Devlin ら [1] の実装を元に，MLM の学習のみ行うよう変更した．分類には Chainer の実装[5]を元に，マルチラベル分類を行うよう修正した．データ選択に使用する LM の学習には KenLM[6]を使用し，5-gram LM を推定した．言及していないハイパーパラメータは，元の実装のものを使用した．

本研究ではマルチラベルの分類問題を，各ラベルごとに付与されるかされないかの 2 値分類として扱った．そのため，ラベルごとに精度の評価を行った．評価尺度として，各ラベルごとにラベルを正しく付与できた場合を正解として， Precision と Recall，F-score を使用した．分類問題はエポック数 5 で学習し，開発データに対して，各ラベルごとの F-score を合計した値が最も大きいものを最終的な分類モデルとして選んだ．また，全ての実験結果はシード値を変えて 3 回実験を行い，平均したものである．

[5]https://github.com/chainer/models/tree/master/bert

[6]https://kheafield.com/code/kenlm/

表 3: 分類実験の結果．太字がその列内で最も良いスコアを示す．平均 F は各ラベルの F-score に対してマクロ平均を計算した結果．

	故障原因			故障状況			措置対策			その他			
	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F	Prec.	Rec.	F	平均 F
Baseline	83.69	93.75	88.42	86.38	91.23	88.73	68.09	79.85	73.45	94.68	84.65	89.37	84.99
Random	90.17	90.42	90.22	86.52	**92.28**	**89.31**	69.17	76.56	72.31	92.37	86.26	89.17	85.25
BCCWJ	89.37	**94.17**	91.68	86.10	91.23	88.59	**70.56**	83.15	76.25	94.55	**86.51**	**90.35**	86.72
Wikipedia	**92.49**	91.67	**92.06**	**86.53**	**92.28**	**89.31**	68.65	**86.45**	**76.51**	**95.21**	85.50	90.08	**86.99**

4.2 追加学習の実験結果

表 2 に追加学習の際の，開発データに対する MLM の loss を示す．Wikipedia の上位 800K 文を追加学習に使用した場合が，開発データに対して最も良い結果だった．この結果から，Random は Wikipedia からランダムに 800K 文抽出して，Base に対して追加学習を行った．Baseline と Random の結果から，MLM の追加学習を行うことで開発データに対する loss が減少することがわかる．

4.3 分類の実験結果

表 3 に分類の実験結果を示す．BCCWJ は Baseline に対して BCCWJ の 800K 文を使用して追加学習を行ったものであり，Wikipedia は Baseline に対して Wikipedia の 800K 文を使用して追加学習をしたものである．また，平均 F は各ラベルに対する F-score の値について，マクロ平均を計算したものである．この結果から，BCCWJ と Wikipedia はどちらも追加学習をしなかった Baseline と比較して平均 F が 1 ポイント以上向上していることがわかる．また，データ選択を行わなかった Random に対しても，BCCWJ や Wikipedia は平均 F が 1 ポイント以上向上していることを確認した．

5 分析と考察

表 3 の Baseline と Random の結果から，目的関数を MLM のみにすることで多少の精度の改善が行われたと考えられる．これは，Liu ら [3] の研究でも同様のことが報告されている．

表 2 に開発データに対する各モデルの F-score の平均を示す．この結果と，表 2 の結果からピアソンの積率相関と，スピアマンの順位相関を計算したところ，それぞれ -0.51 と -0.61 であった．これらの結果は $p < 0.05$ で統計的に有意であった．このことから，事前学習時の MLM の loss の値と，下流タスクである分類問題の精度の間には相関があると考えられる．

6 おわりに

本研究は，最終的に解きたい分野のラベルなしデータが少量の際に，データ選択手法を用いて選ばれた文を用いて BERT の追加学習を行うことで精度が向上することを示した．また，BERT の追加学習を行うことで，対象分野の MLM の loss が減少し，この減少と分類精度の間で相関が確認された．

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. In *Proc. of NAACL-HLT*, pp. 4171–4186, 2019.

[2] Xiaochuang Han and Jacob Eisenstein. Unsupervised domain adaptation of contextualized embeddings for sequence labeling. In *Proc. of EMNLP*, pp. 4237–4247, 2019.

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[4] Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. Balanced corpus of contemporary written Japanese. *Language resources and evaluation*, Vol. 48, No. 2, pp. 345–371, 2014.

[5] Robert C. Moore and William Lewis. Intelligent selection of language model training data. In *Proc. of ACL*, pp. 220–224, 2010.

[6] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune BERT for text classification? *arXiv preprint arXiv:1905.05583*, 2019.

[7] 柴田知秀, 河原大輔, 黒橋禎夫. BERT による日本語構文解析の精度向上. 言語処理学会 第 25 回年次大会, pp. 205–208, 2019.