

金融ドメインにおける事前学習 BERT モデルの性能検証

鈴木 雅弘[†] 坂地 泰紀[†] 平野 正徳[†] 和泉 潔[†]

[†] 東京大学大学院工学系研究科
〒113-8656 東京都文京区本郷 7-3-1

E-mail: [†]mail@msuzuki.me, ^{††}{sakaji,izumi}@sys.t.u-tokyo.ac.jp, ^{†††}hirano@g.ecc.u-tokyo.ac.jp

あらまし 近年, BERT をはじめとした大規模コーパスによって事前学習された汎用言語モデルが盛んに用いられている。日本語においても Wikipedia による事前学習モデルがいくつか公開されている。一方で金融分野においては専門的な言い回しが用いられるため, 汎用モデルでは十分な効果が得られない可能性がある。本研究では, 金融ドメインのコーパスを用いた事前学習モデルを構築し, 金融ドメインのタスクにおける評価を行う。

キーワード 自然言語処理, 言語資源, BERT, 金融文書

Performance Validation of Pre-Trained BERT in the Financial Domain

Masahiro SUZUKI[†], Hiroki SAKAJI[†], Masanori HIRANO[†], and Kiyoshi IZUMI[†]

[†] School of Engineering, The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, 113-8656

E-mail: [†]mail@msuzuki.me, ^{††}{sakaji,izumi}@sys.t.u-tokyo.ac.jp, ^{†††}hirano@g.ecc.u-tokyo.ac.jp

Abstract Recently, general-purpose language models pre-trained on large corpora such as BERT have been widely used. In Japanese, several pre-trained models based on Wikipedia have been published. On the other hand, general-purpose models may not be sufficiently effective in the financial domain because of the use of specialized phrases. In this study, we construct a pre-training model using a corpus of the financial domain, and evaluate it on a task in the financial domain.

Key words Natural language processing, Language resources, BERT, Financial text

1. はじめに

近年, 決算短信や有価証券報告書, ニュース記事や証券レポートなど, インターネットで閲覧可能な金融文書が豊富に存在する。金融関連のテキストの分析は投資やマーケット分析に役立つ一方で, 毎日大量に作成されるテキストを人手によって全て分析することは難しい。そのため, 近年盛んにおこなわれているのが, 金融文書に自然言語処理 (NLP) を適用する金融テキストマイニングである。機械学習を用いた金融関連のツイートのセンチメント分析 [1][2] をはじめとして, 金融分野における自然言語処理に, 機械学習を適用する研究が多く存在する [3][4]。日本語の金融テキストマイニングにおいても複数の研究が存在する。坂地らは決算短信から稀な因果関係知識を発見する教師なしの手法を提案した [5]。彼らの手法は構文パターンを用いて基礎表現と因果関係知識を抽出するものである。鈴木らは, アナリストレポートを用いた株価動向予測を行った [6]。彼らは Bidirectional LSTM や Attention, MLP を組み合わせたネットワークを用いた。また事前実験として行ったアナリストレポートに含まれる意見文抽出タスクにおいて, Wikipedia や新聞記

事から構築した汎用的な分散表現より, アナリストレポートから構築した分散表現の方が精度が高いことを示した。

本研究では, 日本語金融コーパスによって事前学習を行った BERT モデルを提案する。Word2vec [7] や GloVe [8] などによる分散表現は, 教師なしのデータから知識を抽出し, テキストマイニングにおける重要な手法となっている。しかし金融ドメインにおいては特殊な単語が使用されるため, これらの単純な分散表現によるアプローチでは十分な効果が得られない。BERT [9] は事前学習によって各言語タスクの精度を大幅に改善した。BERT は Attention 機構をベースとした Transformer [10] によって主に構成される。まず大規模言語コーパスから事前学習し, その後出力に近いレイヤーのみを学習させるファインチューニングを組み合わせる。日本語においても Wikipedia の記事から事前学習されたモデルが提案されている [11]¹。しかし金融コーパスと一般的なコーパスとの間で語彙や表現の違いが大きいため, 一般的なコーパスで学習したモデルは金融テキストマイニングのタスクに最適とは言えない。また英語では

(注1): <https://github.com/cl-tohoku/bert-japanese>

FinBERT [12] として、Wikipedia や金融に関するニュース記事などを組み合わせたコーパスから構築した事前学習モデルが提案されている。しかしこれらはニュース記事にとどまり、金融の専門用語が多く用いられているわけではない。またファインチューニングや再事前学習 [13] を行うことも考えられるが、これらはネットワークの重みを変更するのみで、入力文をトークン化する時に必要なトークンの語彙を変更することはできない。本研究では、金融コーパスと Wikipedia によってそれぞれ Small サイズの事前学習 BERT モデルを構築する。これらの事前学習モデルを用い、金融ドメインのテキストを対象とした 2 つの評価実験を行い性能を評価する。

本研究の貢献は以下の通りである。

- 金融ドメインの文書からなるコーパスにより、金融特有の専門単語を反映した金融 BERT モデルを構築した。
- 構築した金融モデルを、金融ドメインで研究されているタスクを対象に実験を行い、一般的なコーパスである Wikipedia から作成した BERT モデルよりも高い性能を持つことを示した。

2. BERT モデルの構築

BERT [9] は、大規模なコーパスに対して単語の穴埋め (Masked LM) と 2 文の連続性の判定 (Next sentence prediction) の 2 つのタスクを学習させることで事前学習 (pre-training) し、各タスクでファインチューニングするという 2 段階から構成される。本節では 1 段階目である事前学習について述べる。

2.1 日本語における事前学習

BERT [9] では英語のコーパスを用いており、入力文をトークン化の際に半角スペースで分割し、その後 WordPiece [14] によるサブワード分割を行う。しかし、日本語の文章は半角スペースで分割することができない。そのため、本研究ではまず MeCab [15] によって形態素解析を行い、その後 WordPiece によるサブワード分割を行う。金融コーパスと Wikipedia のそれぞれによって構築された語彙によって、表 1 のように文をトークンに分割することが可能になる。表 1 の場合、金融コーパスによる語彙では「デリバティブ」や「先物」、「スワップ」を 1 語とし扱うのに対し、Wikipedia による語彙ではサブワードを用いて「デリ/##バ/##ティブ」「先/##物」「スワ/##ップ」のように分割して扱う。このように、Wikipedia による汎用的語彙には含まれないものの、金融文書においては登場する単語を、金融コーパスからモデルを作成することで扱うことができる。

2.2 使用データ

事前学習に用いるテキストデータとして、2 種類のデータを用いる。1 つ目は 2012 年 10 月 9 日から 2020 年 12 月 31 日にかけて開示された決算短信等のデータである。2 つ目は EDINET² にて、2018 年 2 月 8 日から 2020 年 12 月 31 日にかけて開示された有価証券報告書等の 2 種類データを用いる。これら 2 つのデータセットから、金融コーパス (約 2,700 万文) を作成した。金融コーパスのデータサイズは約 5.2 GB となった。また、金融コーパスとの比較のために、Wikipedia の日本語記事によるコー

表 1 各コーパスによって構築された語彙から、「デリバティブ取引には、先物取引やスワップ取引がある」という文をトークン化する例。[CLS] は文頭を、[SEP] は文末などを表す。"##" はサブワードに分割された語のうち、先頭でないものに付与される。

コーパス	トークン
金融	[CLS]/デリバティブ/取引/に/は/, /先物/取引/や/スワップ/取引/等 /が/ある./ / [SEP]
Wikipedia	[CLS]/デリ/##バ/##ティブ/取引/に/は/, /先/##物/取引/や/スワ/##ップ/取引/等 /が/ある./ / [SEP]

パス (約 2,000 万文) も作成した。Wikipedia コーパスのデータサイズは約 2.9 GB となった。

2.3 事前学習タスク

BERT の事前学習は、単語の穴埋め (Masked LM) と 2 文の連続性の判定 (Next sentence prediction) の 2 つのタスクの学習によって行われる。Masked LM では、各入力トークンのうち 15 % が Masking の対象となり、事前学習ではこの対象となったトークンを予測する。Masking の対象となったこれらのトークンのうちさらにそのうち 80 % のトークンが [MASK] トークンに置換され、10 % がランダムに別のトークンに置換される。残った 10 % のトークンは元のトークンのまま入力される。Next sentence prediction では、入力のうち 50 % は実際に存在する連続した 2 文を [SEP] トークンでつなぐ。残りの 50 % はランダムにサンプリングしたドキュメントから抽出し、実際には連続しない 2 文を [SEP] トークンでつなぐ。各入力について [SEP] トークンの前後の 2 文が実際に連続しているかを学習する。BERT の論文 [9] では 2 文を [SEP] トークンで接続して入力したとのみ記載があるが、本研究では計算の効率化のため、入力上限である 128 トークンになるまでサンプルの先頭と後尾にトークンを追加した。

2.4 ファインチューニング

BERT では事前学習を行ったモデルの出力に 2 層の MLP を加え、MLP のみを学習させることで各種タスクに対して適応 (ファインチューニング) する。BERT の最終層についても事前学習から学習させることもあるが、本研究では MLP の 2 層のみをファインチューニングで学習する。例えば 2 値分類の場合、[CLS] に対応する最終層の出力を $C \in \mathbb{R}^H$ (H は Hidden Size) とすると、出力 $Y \in \mathbb{R}^2$ は式 (1)(2) によって算出される。

$$U = \tanh(CW_1^T) \quad (1)$$

$$Y = \text{softmax}(\text{Dropout}(UW_2^T)) \quad (2)$$

ここで Dropout(\cdot) は [16] において提案された Dropout を出力の 10 % に対して適用する。Dropout は、特定の層の出力を学習時にランダムに 0 に落とすことで、過学習を防ぐ効果があるとされている。 $W_1 \in \mathbb{R}^{H \times H}$, $W_2 \in \mathbb{R}^{2 \times H}$ は重み行列である。

2.5 実験設定

サブワード分割のためのトークナイザーの学習についての

(注2) : <https://disclosure.edinet-fsa.go.jp/>

表2 事前学習のハイパーパラメータ。

ハイパーパラメータ	値
Number of layers	12
Hidden Size	256
FFN inner hidden size	1,024
Attention heads	4
Embedding Size	128
Learning Rate	5e-4
Batch Size	128
Train Steps	1M

実験設定は東北大学³によって作成されたモデルを参考にした。MeCabの辞書はIPAdicを用い、語彙数は32,768とした。このうち5語を未知語を表す[UNK]、文頭に挿入される[CLS]、2文の間や入力最後の最後に挿入される[SEP]、入力長を揃えるために入力される[PAD]、Masked LMタスクの際に用いられる[MASK]に割り当てた。また、新たにファインチューニングの際に単語を追加するために10語を、1文字の単語のために6,129語を割り当てた。BERTモデルのパラメータは[17]において用いられているSmallモデルを参考に、表2のように設定した。Learning Rateは10,000ステップまでWarmupを行い、そこから線形に減衰させた。表2におけるLearning Rateは、10,000ステップにおける値である。実装はPyTorchベースの実装⁴を用いた。100,000ステップにNVIDIA A100で約3時間、NVIDIA Tesla V100で約6時間を要した。

3. 評価実験

金融コーパスから構築したBERTモデル(金融BERT)とWikipediaから構築したBERTモデル(WikiBERT)に対し、ファインチューニングによる評価実験を行い性能を評価する。構築した金融BERTを評価するために、因果関係を含んでいる文を抽出するタスクを行う。因果関係を含んでいる文の抽出については、坂地ら[18]が取り組んでおり、彼らは経済・金融の記事を分析するために因果関係抽出を行っている。因果関係の抽出は、2つのタスクで構成されており、一つ目が因果関係を含んでいる文を抽出するタスク、もう一つが因果関係を示す原因と結果表現を抽出するタスクである。ここでは、坂地らが機械学習を用いて取り組んでいた因果関係を含む文の抽出を対象に金融BERTの性能検証を行う。この実験においては、日経新聞記事から作成されたデータを以下のように分割して用いる。1,305文(うち571文が因果関係文)を学習データに、327文(うち138文が因果関係文)を検証データに、413文(うち189文が因果関係文)をテストデータに分割し、実験に使用する。

加えて、坂地らは決算短信を対象に、同様の実験を行っており[19]、こちらのデータを対象にも実験を行う。この実験においては、決算短信から作成されたデータを以下のように分割して用いる。1,850文(うち243文が因果関係文)を学習データに、

463文(うち60文が因果関係文)を検証データに、578文(うち91文が因果関係文)をテストデータに分割し、実験に使用する。

4. 結果と考察

表3に日経新聞記事を対象に、因果関係を含む文の抽出を行った結果を示す。また、表4に決算短信を対象に、因果関係を含む文の抽出を行った結果を示す。さらに、それぞれのBERTをファインチューニングした際の最適なパラメータを表5と表6に示す。ここで、表5と表6のlr_lastlayerはBERTの最終層の重みを再学習させる際の学習率を示し、lr_classifyは因果関係を含む文か否かを判定するMLP層の重みを学習させる際の学習率を示す。

表3 日経新聞記事データを対象に因果関係を含む文の抽出における評価実験結果。Precision, Recall, F1はマクロ平均である。

	Accuracy	Precision	Recall	F1
金融BERT	0.898	0.898	0.898	0.898
WikiBERT	0.891	0.891	0.889	0.890

表4 決算短信データを対象に因果関係を含む文の抽出における評価実験結果。Precision, Recall, F1はマクロ平均である。

	Accuracy	Precision	Recall	F1
金融BERT	0.950	0.910	0.899	0.904
WikiBERT	0.908	0.824	0.838	0.831

表5 日経新聞記事データを対象に学習した際に求めた最適パラメータ

	Parameters
金融BERT	epoch: 23, batch_size: 256, lr_lastlayer: 0.000014, lr_classify: 0.000039
WikiBERT	epoch: 30, batch_size: 64, lr_lastlayer: 0.000063, lr_classify: 0.000355

表6 決算短信データを対象に学習した際に求めた最適パラメータ

	Parameters
金融BERT	epoch: 22, batch_size: 128, lr_lastlayer: 0.001037, lr_classify: 0.000010
WikiBERT	epoch: 25, batch_size: 32, lr_lastlayer: 0.004229, lr_classify: 0.041382

表3と表4より、日経新聞記事データにおいても、決算短信データにおいても、金融BERTを用いた方がF1値が高くなった。このことより、まだ1つのタスクしか実験していないが、金融におけるタスクではWikipediaベースのBERTよりも、金融BERTの方が高い性能を示す可能性を示すことができた。加えて、日経新聞記事データにおける結果よりも、決算短信データにおける結果の方が、よりWikiBERTよりも大きな差をつけてF1値が高いことを確認することができた。これは、金融BERTが決算短信データから作成されていることが起因していると考えられる。一方、日経新聞記事データを用いた実験にお

(注3) : <https://github.com/cl-tohoku/bert-japanese>

(注4) : <https://github.com/huggingface/transformers>

いては、WikiBERT の結果も金融 BERT の結果も大きな差は見られなかった。これは、金融 BERT が新聞記事などの汎用的な文書から作成されていないことが原因であると考えられる。今後は、決算短信や有価証券報告書だけではなく、新聞記事や Wikipedia などのデータも加えて学習させることで、より高い性能を示す金融 BERT が構築できると考えられる。

5. ま と め

本論文では、決算短信等のデータと有価証券報告書等のデータを用いて金融 BERT を構築し、その性能を確認した。金融ドメインで研究されている因果関係を含む文の抽出タスクを対象に、日経新聞記事から作成された評価データと決算短信から作成された評価データを用いて実験を行い、Wikipedia から作成した BERT モデルよりも高い性能を示した。特に、決算短信から作成された評価データにおいては、大きく性能が向上することを確認することができた。

今後の課題として、因果関係を含む文の抽出だけではなく、株価予想や要約などの他のタスクにおいて高い性能を示すことができるかを検証していく。また、作成した金融 BERT の公開に向けた準備を進めていく。

謝辞 本研究は、JSPS 科研費 (JP21K12010) の助成を受けました。

文 献

- [1] N. Oliveira, P. Cortez, and N. Areal, "The impact of microblogging data for stock market prediction: Using twitter to predict returns, volatility, trading volume and survey sentiment indices," *Expert Systems with Applications*, vol.73, pp.125–144, 2017.
- [2] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, and I. Mozetič, "The effects of twitter sentiment on stock price returns," *PLoS ONE*, 2015.
- [3] B.S. Kumar and V. Ravi, "A survey of the applications of text mining in financial domain," *Knowledge-Based Systems*, vol.114, pp.128–147, 2016.
- [4] L. Guo, F. Shi, and J. Tu, "Textual analysis and machine learning: Crack unstructured data in finance and accounting," *The Journal of Finance and Data Science*, vol.2, no.3, pp.153–170, 2016.
- [5] H. Sakaji, R. Muro, H. Sakai, J. Bennett, and K. Izumi, "Discovery of rare causal knowledge from financial statement summaries," *The 2017 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFER)*, pp.602–608, 2017.
- [6] M. Suzuki, H. Sakaji, K. Izumi, H. Matsushima, and Y. Ishikawa, "Forecasting net income estimate and stock price using text mining from economic reports," *Information*, 2020.
- [7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, 2013.
- [8] P. Jeffrey, S. Richard, and M. Christopher, "Glove: Global vectors for word representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp.1532–1543, Association for Computational Linguistics, 2014.
- [9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *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)*, pp.4171–4186, Association for Computational Linguistics, Minneapolis, Minnesota, June 2019.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems 30 (NeurIPS)*, eds. by I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, pp.5998–6008, Curran Associates, Inc., 2017. <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>
- [11] 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*, 2019.
- [12] Z. Liu, D. Huang, K. Huang, Z. Li, and J. Zhao, "Finbert: A pre-trained financial language representation model for financial text mining," *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, ed. by C. Bessiere, pp.4513–4519, International Joint Conferences on Artificial Intelligence Organization, July 2020. Special Track on AI in FinTech.
- [13] 仁木裕太, 坂地泰紀, 和泉 潔, 松島裕康, "再事前学習した bert を用いた金融文書中の因果関係知識有無の判別," *人工知能学会全国大会論文集*, 2020.
- [14] M. Schuster and K. Nakajima, "Japanese and korean voice search," *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp.5149–5152, 2012.
- [15] T. Kudo, K. Yamamoto, and Y. Matsumoto, "Applying conditional random fields to japanese morphological analysis," *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pp.230–237, Association for Computational Linguistics, 2004.
- [16] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.
- [17] K. Clark, M.-T. Luong, Q.V. Le, and C.D. Manning, "Electra: Pre-training text encoders as discriminators rather than generators," 2020.
- [18] 坂地泰紀, 増山 繁, "新聞記事からの因果関係を含む文の抽出手法," *電子情報通信学会論文誌 D*, vol.J94-D, no.8, pp.1496–1506, 2011.
- [19] 坂地泰紀, 酒井浩之, 増山 繁, "決算短信 pdf からの原因・結果表現の抽出," *電子情報通信学会論文誌 D*, vol.J98-D, no.5, pp.811–822, 2015.