金融文書を用いた事前学習言語モデルの構築と検証

Construction and Validation of a Pre-Trained Language Model Using Financial Documents

鈴木 雅弘[1*]　坂地 泰紀[1]　平野 正徳[1]　和泉 潔[1]
Masahiro Suzuki[1]　Hiroki Sakaji[1]　Masanori Hirano[1]　Kiyoshi Izumi[1]

[1] 東京大学大学院工学系研究科
[1] School of Engineering, The University of Tokyo

Abstract: BERT を始めとする事前学習言語モデルは，様々な自然言語処理のタスクにおいて成果を上げている．これらのモデルの多くは Wikipedia やニュース記事などの一般的なコーパスを用いているため，専門的な単語が使用される金融分野においては十分な効果が得られない．本研究では決算短信や有価証券報告書から事前学習言語モデルを構築する．また金融ドメインのタスクによって汎用モデルとの性能を比較する．

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].

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 モデルと ELECTRA モデルを提案する．Word2vec[5] や GloVe[6] 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[7] 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[8] 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．また ELECTRA[9] は BERT に GAN[10] のアイデアを加え，さらに Generator において最尤法 [11] を適用した．その結果 GLUE において，BERT より少ない計算量で高い精度を示した．日本語においても Wikipedia の記事から事前学習された BERT モデルが提案されている [12][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，一般的なコーパスのみ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 FinBERT[13] 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 [14] 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 を組み合わせたコーパスから Small サイズの事前学習 BERT モデルと ELECTRA モデルを構築する．また比較対象として Wikipedia のみからも Small サイズの事前学習モデルをそれぞれ構築する．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と Wikipedia からなるコーパスを組み合わせ，金融特有の専門単語を反映した金融 BERT モデル・金融 ELECTRA 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

*連絡先：東京大学大学院工学系研究科
〒 113-8686 東京都文京区本郷 7-3-1
E-mail: b2019msuzuki@socsim.org

[1] https://github.com/cl-tohoku/bert-japanese

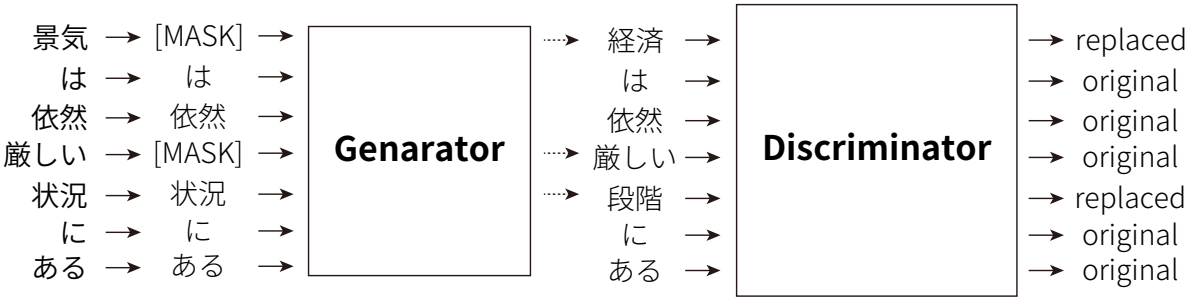

図 1: ELECTRA の概要図．Generator では Masking の対象となったトークンの予測を行う．Discriminator では Generator から出力されたトークンが元のトークンと置き換えられているかを判別する．例では「景気」「厳しい」「状況」の 3 単語が Masking の対象となり，そのうち「景気」「厳しい」の 2 単語が [MASK] に置換され Generator に入力される.「状況」は [MASK] に置換されずにそのまま入力される．Generator によって 3 単語がそれぞれ「経済」「厳しい」「段階」と予測される.「景気」「状況」の 2 単語がそれぞれ Generator によって「経済」「段階」に置換されたため，Discriminator では”replaced”のラベルが付与される．それ以外の単語は元のトークンから置換されなかったため”original”のラベルが付与される．

パスである Wikipedia のみから作成したモデルよりも高い性能を持つことを示した．

2 モデルの構築

BERT や ELECTRA といった言語モデルでは，大規模なコーパスに対してタスクを与えて事前学習 (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．

2.1 BERT の事前学習タスク

BERT の事前学習は，単語の穴埋め (Masked LM) と 2 文の接続性の判定 (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，各入力のトークンのうち 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 [7] 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する．

2.2 ELECTRA の事前学習タスク

ELECTRA の事前学習は，入力文の一部のトークンを置き換え，置き換えたトークンを検知するタスク (Replaced Token Detection) によって行われる．ELECTRA は図 1 のように Generator と Discriminator の 2 つのアーキテクチャによって構成され，それぞれに与えられたタスクを同時に学習するマルチタスクによって事前学習を行う．Generator と Discriminator のどちらも，BERT と同様に Transformer の Encoder を重ねたものである．ファインチューニングで使用されるのは Discriminator 部分のみである．Generator は Discriminator が学習しやすいように，Discriminator の 1/4 から 1/3 のサイズ (ELECTRA+は Discriminator と同じサイズ) に設定する．入力トークンの 15%を Masking の対象とする．そのうち 85%のトークンを [MASK] トークンに置換し，残りの 15%は元のトークンのまま Generator に入力する．Generator は Masking の対象となったトークンが元々どのトークンであったかを予測する．このタスクは BERT における Masked LM と似たタスクである．Discriminator には，Generator が予測したトークンを入力する．Discriminator では，入力されたトークンが Generator によって置き換えられたかを判別する 2 値分類タスクを行う．その際，Generator によって正しく予測されたトークンは Generator によって置換されていないものとラベリングする．ELECTRA

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]

における損失関数は式 (1) によって計算される．

$$\mathcal{L}_{\mathrm{ELECTRA}} = \mathcal{L}_{\mathrm{MLM}} + 50\mathcal{L}_{\mathrm{Disc}} \tag{1}$$

ここで $\mathcal{L}_{\mathrm{MLM}}, \mathcal{L}_{\mathrm{Disc}}$ はそれぞれ Generator，Discriminator におけるタスクの Loss である．BERT と同様，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する．

2.3 日本語における事前学習

BERT や ELECTRA では英語のコーパスを用いており，入力文をトークン化する際に半角スペースで分割し，その後 WordPiece[15] 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[16] 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．

2.4 使用データ

事前学習に用いる金融コーパスのテキストデータとして，3 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 EDINET[2]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．3 つ目は Wikipedia の日本語記事によるコーパスである．これら 3 つのデータセットから，金融コーパス (約 4,700 万文) を作成した．金融コーパスのデータサイズは約 8GB となった．また，金融コーパスとの比較のために，Wikipedia のテキストデータ (約 2,000 万文) のみから作成したコーパスも用いる．

表 2: 事前学習のハイパーパラメータ．Generator Size は，Transformer エンコーダーの層の数は 12，隠れ層の数は 256，Transformer エンコーダーの FFN の層数は 1024，Transformer エンコーダーの Attention Head の数は 4，Embedding の次元数は 128，学習率は 5e-4 で共通である．

パラメータ	BERT	ELECTRA	ELECTRA+
Generator Size	-	1/4	1/1
Train Steps	1.45M	1M	1M

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など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．各モデルのパラメータは [9] 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．ELECTRA+モデルは，Google が公開している ELECTRA-Small モデル[4]の Generator のサイズが Discriminator のサイズと同じであることを

[2] https://disclosure.edinet-fsa.go.jp/

[3] https://github.com/cl-tohoku/bert-japanese

[4] https://github.com/google-research/electra

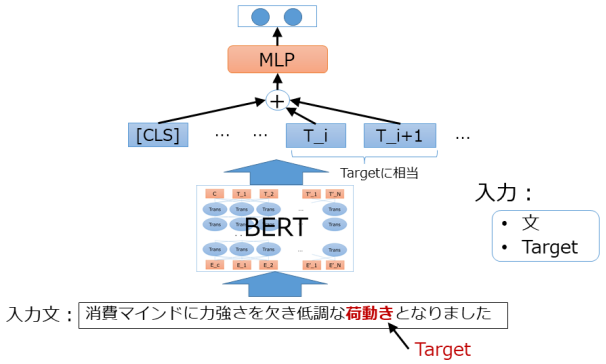

図 2: chABSA-detaset を用いた Aspect-Based Sentiment Analysis に関する実験のネットワーク概要図.

参考に作成した．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[5]を用いた．

3 評価実験

金融コーパスから構築したモデル (金融*) と Wikipedia から構築したモデル (Wiki*) 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．構築した金融モデルを評価するために，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 [17] 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 [18]，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．

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．Prec, Recall, F1 はマクロ平均である．

	Acc	Prec	Recall	F1
金融 BERT	**0.891**	**0.890**	**0.890**	**0.890**
金融 ELECTRA	0.872	0.876	0.867	0.869
金融 ELECTRA+	0.879	0.878	0.878	0.878
WikiBERT	0.877	0.887	0.870	0.873
WikiELECTRA	0.845	0.851	0.839	0.842
WikiELECTRA+	0.828	0.827	0.828	0.827

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．Prec, Recall, F1 はマクロ平均である．

	Acc	Prec	Recall	F1
金融 BERT	**0.929**	0.860	**0.882**	**0.870**
金融 ELECTRA	0.917	0.837	0.866	0.850
金融 ELECTRA+	0.903	0.815	0.826	0.821
WikiBERT	0.926	0.852	0.880	0.865
WikiELECTRA	0.843	0.421	0.500	0.457
WikiELECTRA+	0.929	**0.870**	0.860	0.864

表 5: chABSA-dataset を対象とした評価実験結果．Prec, Recall, F1 はマクロ平均である．

	Acc	Prec	Recall	F1
金融 BERT	**0.884**	**0.881**	0.880	**0.881**
金融 ELECTRA	0.881	0.877	**0.882**	0.879
金融 ELECTRA+	0.847	0.847	0.838	0.842
WikiBERT	0.862	0.864	0.853	0.857
WikiELECTRA	0.845	0.851	0.839	0.842
WikiELECTRA+	0.599	0.580	0.565	0.558

さらに，本論文では，TIS 株式会社が公開している chABSA-dataset[6]を用いて，Aspect-Based Sentiment Analysis に関する実験を行う．この実験では，図 2 のように入力に文と表現を入力し，その表現に関する文内でのセンチメントを出力するという問題設定にする．データセットには，Positive，Negative，Neutral のタグが付与されていたが，Neutral が他のタグに対して大幅に少なかったことから，Neutral を除外して実験を行う．ここでは，4,479 件 (うち 2,776 件が Positive) を学習データに，1,194 件 (うち 690 件が Positive) を検証データに，1,492 件 (うち 868 件が Positive) をテストデータに分割し，実験に使用する．

[5] https://github.com/huggingface/transformers

[6] https://github.com/chakki-works/chABSA-dataset

4 結果と考察

表 3 に日経新聞記事を対象に，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それぞれ示す．また表 5 に chABSA-dataset を用いてセンチメント出力を行った結果を示す．

表 3，4，5 より，各実験において金融モデルを用いた方が F1 値が高くなった．このことより，金融におけるタスクでは Wikipedia をコーパスとしたモデルよりも，金融テキストをコーパスとしたモデルの方が性能が良いことを示した．モデル間の比較では，BERT モデルが最も F1 値が高かった．金融の ELECTRA モデルと ELECTRA+モデルは，日経新聞記事データを対象とした実験ではモデルサイズの大きい ELECTRA+モデルの方が F1 値が高かったものの，決算短信と chABSA-dataset を対象とした実験ではモデルサイズの小さい ELECTRA モデルの方が F1 値が高かった．これは ELECTRA の Generator が Discriminator の 1/4 から 1/3 のサイズでより精度が高かったこと [9] と同様の現象が発生していると考えられる．実際 ELECTRA の論文では ELECTRA+のような Generator と Discriminator のサイズが同じモデルについての言及はない．また決算短信データを対象とした実験での WikiELECTRA と chABSA-dataset を対象とした実験での WikiELECTRA+は，他のモデルより F1 値が大きく低くなった．これは ELECTRA の学習が不安定なことによると考えられる．今回は金融ドメインから決算短信と有価証券報告書の 2 種類のデータのみを使用したが，今後は，新聞記事の金融面など金融に関連した別のテキストデータも併せて学習することで，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 Wikipedia の日本語記事と組み合わせて金融ドメインの事前学習モデルを構築し，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 から作成したモデルよりも高い性能を示した．また，chABSA-dataset を用いたセンチメント分析においても，金融 BERT や金融 ELECTRA が，Wikipedia から作成したモデルよりも高い性能を示した．

今後の課題として，因果関係を含む文の抽出やセンチメントの分析だけではなく，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，Optimzier やモデルの改良により ELECTRA の学習を安定することを目指す．さらに作成した各モデル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．

参考文献

[1] Nuno Oliveira, Paulo Cortez, and Nelson 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] Gabriele Ranco, Darko Aleksovski, Guido Caldarelli, Miha Grčar, and Igor Mozetič. The effects of twitter sentiment on stock price returns. PLoS ONE, 2015.

[3] B. Shravan Kumar and Vadlamani Ravi. A survey of the applications of text mining in financial domain. *Knowledge-Based Systems*, Vol. 114, pp. 128–147, 2016.

[4] Li Guo, Feng Shi, and Jun Tu. Textual analysis and machine leaning: Crack unstructured data in finance and accounting. *The Journal of Finance and Data Science*, Vol. 2, No. 3, pp. 153–170, 2016.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. CoRR, 2013.

[6] Pennington Jeffrey, Socher Richard, and Manning Christopher. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1532–1543, 2014.

[7] 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)*, pp. 4171–4186, 2019.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30 (NeurIPS)*, pp. 5998–6008. 2017.

[9] Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

[10] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[11] Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short, 2020.

[12] 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.

[13] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. Finbert: A pre-trained financial language representation model for financial text mining. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pp. 4513–4519, 2020. Special Track on AI in FinTech.

[14] 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.

[15] Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5149–5152, 2012.

[16] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pp. 230—237, 2004.

[17] 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.

[18] 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.