Constructing and Analyzing Domain-Specific Language Model for Financial Text Mining

Masahiro Suzuki[a,*], Hiroki Sakaji[a], Masanori Hirano[a] and Kiyoshi Izumi[a]

[a]*The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan*

ARTICLE INFO

Keywords:
language models
domain-specific pre-training
financial market
natural language processing

ABSTRACT

The application of natural language processing (NLP) to financial fields is advancing with an increase in the number of available financial documents. Transformer-based models such as Bidirectional Encoder Representations from Transformers (BERT) have been successful in NLP in recent years. These cutting-edge models have been adapted to the financial domain by applying financial corpora to existing pre-trained models and by pre-training with the financial corpora from scratch. In Japanese, by contrast, financial terminology cannot be applied from a general vocabulary without further processing. In this study, we construct language models suitable for the financial domain. Furthermore, we compare methods for adapting language models to the financial domain, such as pre-training methods and vocabulary adaptation. We confirm that the adaptation of a pre-training corpus and tokenizer vocabulary based on a corpus of financial text is effective in several downstream financial tasks. No significant difference is observed between pre-training with the financial corpus and continuous pre-training from the general language model with the financial corpus. We have released our source code and pre-trained models.

1. Introduction

In financial markets, many companies publish a large volume of textual content, in addition to the relevant numerical data to reduce the information asymmetry between companies and investors. Specifically, periodic disclosure documents such as financial results, securities reports, 10-K, and 10-Q contain a great deal of textual information. This information includes the topics of prevailing business situations, management discussion and analysis (so-called MD&A), and current business risks, which cannot be derived from numerical data. Moreover, many companies have recently begun publishing documents like investor relation (IR) reports more frequently to provide additional information to their investors. Similarly, corporate social responsibility (CSR) reports are published to conform to the environmental, social, and governance (ESG) criteria. In addition to the documents published by companies for the benefit of investors, other types of textual information sources, such as stock reports by analysts, microblogs by investors, and news texts related to financial markets, are also available. For example, Twitter content has been used to perform sentiment analyses of financial markets (Oh and Thomas, 2008; Zhou, Zhao and Wang, 2011; Smailović, Grčar, Lavrač and Žnidaršič, 2013).

As the volume of such content has dramatically increased in recent years, natural language processing (NLP) methods are being widely used to process these data (Kumar and Ravi, 2016). In particular, periodic disclosure documents are now being published in machine-readable formats such as HTML or JSON. These advancements have also accelerated the usage of NLP in financial markets. For instance, long short-term memory (LSTM) and support-vector machine (SVM) models have been applied to predict stock prices from financial texts (Bar-Haim, Dinur, Feldman, Fresko and Goldstein, 2011; Ranco, Aleksovski, Caldarelli, Grčar and Mozetič, 2015; Akita, Yoshihara, Matsubara and Uehara, 2016).

In recent NLP research, several language models have been proposed, and the ones based on the Transformer architecture have performed considerably well on language tasks (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin, 2017; Devlin, Chang, Lee and Toutanova, 2019). Following Bidirectional Encoder Representations from Transformers (BERT), which is the most representative language model developed in recent years, many Transformer-based language models have been proposed (Radford, Narasimhan, Salimans and Sutskever; Radford, Wu,

*Corresponding author

`b2019msuzuki@socsim.org` (M. Suzuki); `sakaji@sys.t.u-tokyo.ac.jp` (H. Sakaji); `research@mhirano.jp` (M. Hirano); `izumi@sys.t.u-tokyo.ac.jp` (K. Izumi)

ORCID(s): 0000-0001-8519-5617 (M. Suzuki); 0000-0001-5030-625X (H. Sakaji); 0000-0001-5883-8250 (M. Hirano)

Child, Luan, Amodei and Sutskever, 2019; Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, Agarwal, Herbert-Voss, Krueger, Henighan, Child, Ramesh, Ziegler, Wu, Winter, Hesse, Chen, Sigler, Litwin, Gray, Chess, Clark, Berner, McCandlish, Radford, Sutskever and Amodei, 2020; Yang, Dai, Yang, Carbonell, Salakhutdinov and Le, 2019; Lan, Chen, Goodman, Gimpel, Sharma and Soricut, 2019; Liu, Ott, Goyal, Du, Joshi, Chen, Levy, Lewis, Zettlemoyer, Stoyanov and Allen, 2019b; Clark, Luong, Le and Manning, 2020). These models have mainly been trained with general corpora such as Wikipedia[1] or Common Crawl.[2]

Although these cutting-edge language models have been applied to perform various functions related to financial markets, further advancements are required to expand their usage. If the text processing for financial documents were made more enriching, a more beneficial analysis via NLP could be performed to extend the current uses of NLP in finance.

Domain-specified language models are highly necessary to process text documents to obtain information for applying to financial markets. In particular, text documents relevant to financial markets often include domain-specific technical language or jargon not commonly used in general conversation or other fields. Thus, language models developed using generalized text corpora such as Wikipedia are unsuitable for analyzing financial text. Several studies have considered adapting the BERT architecture to the financial domain in English. Araci (2019) continued to perform pre-training on a financial dataset from the checkpoint of a BERT model pre-trained from general corpora. Liu, Huang, Huang, Li and Zhao (2020) used corpora of financial text and general corpora for pre-training from scratch. These models differ in that financial text is applied for domain-adaptive continuous pre-training or is applied for pre-training from scratch. It is unclear which method is more effective in financial language tasks. In adapting such methods to the financial domain in English, Peng, Chersoni, Hsu and Huang (2021) found that the corpus used to train a language model has a significant impact on its performance but its vocabulary does not. By contrast, especially in non-European languages such as Japanese, many financial terms have been imported from European languages (mainly from English) and used directly with same pronunciations. Consequently, general language models are incompatible with these words. For example, in Japanese, the financial terminology "derivative" is typically pronounced as it is in English, and it is written with a phonogram, not with the ideograms, which are more commonly used in Japanese writing. Thus, although the word "derivative" is recognized as a single word in a frequently-used English BERT tokenizer, it is recognized as three words —deri/va/tive— in a popular Japanese BERT tokenizer. This occurs because such technical terms are specific to the financial field and are not common among the generalized sets of words learned by those tokenizers, and such unfamiliar phonogramic words tend to be divided into minimum parts such as syllables. If an important word in finance is split into multiple tokens, performance in downstream tasks may be degraded because the model cannot recognize the word. In languages such as Japanese, in which technical terms are split into multiple tokens by a general tokenizer, it may be beneficial to adapt the tokenizer to the financial domain.

Therefore, in this study, we consider the training process of domain-specific language models in finance, particularly in Japanese. We also discuss a methodology to solve incompatibility problems with such language models. We construct and compare multiple language models and develop a methodology for constructing language models that perform better in this specific domain. The implementations and models used in this study are made available online.

The main contributions and findings of this study are summarized as follows:

- We construct and compare methods for adapting language models to the financial domain, such as pre-training methods and vocabulary adaptation. Our implementations and pre-trained models are publicly available.
- We confirm that the adaptation of a pre-training corpus and tokenizer vocabulary with a financial corpus is effective on downstream financial tasks.
- We demonstrate a case in which a tokenizer trained on a general corpus may split a single word of financial terminology into several tokens in Japanese and that a tokenizer trained on the financial corpus does not exhibit this issue.
- Furthermore, we demonstrate that there is no significant difference between pre-training with the financial corpus and continuous pre-training from the general language model with the financial corpus.

[1] https://www.wikipedia.org/

[2] https://commoncrawl.org/

2. Related Work

Pre-trained Language Models

Natural language processing technologies have considerably advanced by the combination of distributed representation and recurrent neural networks (RNNs) such as the LSTM models (Hochreiter and Schmidhuber, 1997). Erhan, Bengio, Courville, Manzagol, Vincent and Bengio (2010) demonstrated that unsupervised pre-training yields better initial training parameters than random initial values and also highlighted the benefits of pre-training. Mikolov, Chen, Corrado and Dean (2013) proposed word2vec as a method of expressing words by vectors instead of in one-hot format. Word2vec builds a distributed representation from corpora without labeled training data based on a distribution hypothesis that the meanings of the words are formed by surrounding words. Le and Mikolov (2014) proposed Doc2vec as an extension of word2vec, and constructed a distributed representation of sentences rather than words. Bojanowski, Grave, Joulin and Mikolov (2017) proposed fastText to respond to unknown words by adding a subword. GloVe (Pennington, Socher and Manning, 2014) supplements word2vec's local representations by adding a global cooccurrence representation of words with matrix factorization, which improved calculation efficiency and accuracy. Dai and Le (2015) demonstrated the effectiveness of initializing LSTMs with the weights of the language models and autoencoders trained on unlabeled data. Peters, Neumann, Iyyer, Gardner, Clark, Lee and Zettlemoyer (2018) proposed ELMo, which uses a bidirectional LSTM (Schuster and Paliwal, 1997) to pre-train a distributed representation, which makes it possible to construct a context-dependent representation. They also used the output of the pre-trained model in the main task. Other approaches (Peters, Ammar, Bhagavatula and Power, 2017; McCann, Bradbury, Xiong and Socher, 2017) have been developed to use a distributed representation obtained by pre-training as a fixed parameter and used the output of the pre-trained model as part of the input of the main task. Howard and Ruder (2018) proposed Universal Language Model Fine-tuning (ULMFiT). ULMFiT consists of three steps in training, including pre-training a language model (LM) on a general corpus, fine-tuning the LM on the target data, and finally fine-tuning a classifier on the target task. GPT-1 (Radford et al.) uses 12 layers, each with the same form as the decoder layers of the Transformer (Vaswani et al., 2017) architecture. It is a kind of attention mechanism in which each token is weighted in parallel by multiple layers. The pre-training process of GPT-1 aims at maximizing the likelihood of the target token being present in a context window. By increasing the size of models and datasets, GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) achieved fine-tuning with few-shot or one-shot learning in downstream tasks without supervised learning.

BERT (Devlin et al., 2019) provided the most significant breakthrough in recent years. The base BERT model comprises 12 layers of the encoder layers of Transformer. BERT performs two tasks as pre-training. The first is bidirectional training with a masked language model (MLM, masked LM), and the other is next sentence prediction (NSP). XLNet (Yang et al., 2019), based on Transformer-XL (Dai, Yang, Yang, Carbonell, Le and Salakhutdinov, 2019), has achieved higher accuracy than BERT with a larger corpus and greater computational complexity. Transformer-XL can input the information of a longer sequence and can learn relative position information. Like GPT, XLNet maximizes the likelihood as a task. In contrast to unidirectional model architectures, training is performed by changing the order of input in the model. ALBERT (Lan et al., 2019) improves on BERT by modifying its parameters and tasks. ALBERT reduces the number of parameters by sharing the parameters of each layer and decomposing the embedding matrix. Additionally, ALBERT applies sentence order prediction (SOP) rather than NSP, because NSP is inefficient as a pre-training task. RoBERTa (Liu et al., 2019b) improves on the tuning method of BERT. It removes NSP and performs dynamic masking with large minibatches and also improves the input format. Based on the idea of knowledge distillation (Hinton, Vinyals and Dean, 2014), Sanh, Debut, Chaumond and Wolf (2019) proposed DistilBERT, which reduces the number of parameters without a significant decrease in performance by training a small student model to converge with a large teacher model. ELECTRA (Clark et al., 2020) applies a generative adversarial network (GAN) (Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville and Bengio, 2014) to BERT and employs another pre-training task (replaced token detection) in the discriminator. ELECTRA has achieved higher accuracy with a smaller amount of calculation than BERT owing to the GAN-like architecture. The pre-training process of ELECTRA includes a replaced token detection (RTD) task, in which the model replaces some tokens in the input sentence and detects the replaced tokens. Yamaguchi, Chrysostomou, Margatina and Aletras (2021) also reported that judging the shuffled word order or detecting randomly replaced tokens performed better as pre-training tasks than MLM. Aroca-Ouellette and Rudzicz (2020) explored various pre-training tasks and demonstrated that MLM should be included as a pre-training multitask and that other tasks outperformed NSP.

Domain-Adaptive Pre-training

Language models can be trained in various ways according to the task and domain to which they are applied. It has been demonstrated that supervised learning between pre-training with a general corpus and fine-tuning with a sufficiently labeled corpus near the target domain can improve the performance of language models (Phang, Févry and Bowman, 2018; Chakrabarty, Hidey and McKeown, 2019). It has also been demonstrated that performance can be improved by continuing pre-training with the corpus used in the main task, even in the absence of adequately labeled data (Howard and Ruder, 2018; Sun, Qiu, Xu and Huang, 2019; Logeswaran, Chang, Lee, Toutanova, Devlin and Lee, 2019). Liu, He, Chen and Gao (2019a); Sun et al. (2019) demonstrated that performing fine-tuning and multi-task learning simultaneously for multiple target tasks — instead of fine-tuning each task individually — improves the overall performance of the model. Some models have also been pre-trained from scratch using the corpora of specific domains, rather than general corpora, and have been used to process data such as biomedical text (Lee, Yoon, Kim, Kim, Kim, So and Kang, 2019), clinical documents (Huang, Altosaar and Ranganath, 2020), scientific publications (Beltagy, Lo and Cohan, 2019), medical and health data (Rasmy, Xiang, Xie, Tao and Zhi, 2021), or legal text (Chalkidis, Fergadiotis, Malakasiotis, Aletras and Androutsopoulos, 2020).

Similarly, several BERT-like models suitable for processing information relevant to the financial field have been proposed. Araci (2019) used the Reuters TRC2 corpus (National Institute of Standards and Technology (U.S.), 2018) to perform domain-adaptive pre-training on a BERT model that was pre-trained on BookCorpus[3] and English Wikipedia. Their BERT model continued to perform pre-training on the dataset of the target task on the checkpoint of the BERT model pre-trained from the general corpora, and it did not differ significantly in performance. Liu et al. (2020) used the corpora of financial text (FinancialWeb, YahooFinance, and RedditFinanceQA) and general corpora (BookCorpus and English Wikipedia) to pre-train a learning model from scratch. They performed six self-supervised pre-training tasks rather than MLM and NSP as in the original BERT. Peng et al. (2021) compared the performance of financial BERT models with financial and general vocabularies. They reported that vocabulary was not necessarily important in financial downstream tasks, although pre-training with a financial corpus was effective. These studies demonstrated the benefits of pre-training from scratch (Liu et al., 2020; Peng et al., 2021) and domain-adaptive continuous pre-training from the checkpoint of a BERT model pre-trained on general corpora (Araci, 2019). However, they compared their financial BERT models with BERT trained on the general corpora, and their models were constructed in English. Therefore, in the present work, we compare these pre-training methods and the influence of tokenizers over languages other than English.

Financial Text Mining

We summarize some related works on general financial text mining. Bollen, Mao and Zeng (2011) demonstrated that Twitter moods were useful for forecasting the Dow Jones Industrial Average. The researchers used a self-organizing fuzzy neural network to forecast the indicator, with which they were able to predict its rise and fall with an accuracy of over 80%. Schumaker and Chen (2009) proposed a machine learning approach to predict stock prices by analyzing financial news articles. Their method predicted indicators and stock prices; however, it did not analyze inter-industry relations. Sakaji, Sakai and Masuyama (2008) proposed a method to automatically extract basis expressions indicating economic trends from newspaper articles using a statistical approach. Additionally, Koppel and Shtrimberg (2006) proposed a method to classify a company's news stories based on their apparent impact on the company's stock performance. Ito, Sakaji, Izumi, Tsubouchi and Yamashita (2018) proposed a neural network model designed to visualize online financial textual data, which determined the sentiment of words and their categories. Lastly, Milea, Sharef, Almeida, Kaymak and Frasincar (2010) predicted the MSCI euro index (examining whether its value would increase, decrease, or remain the same) based on fuzzy grammar fragments extracted from a report published by the European Central Bank. The above-mentioned studies extracted information for investors or predicted stock prices using information extracted from the text data. In this study, we consider a general language model specialized for financial markets, which is a necessary basis for financial text mining. This has recently become a popular approach.

Regarding the application of the pre-trained language model, Bingler, Kraus, Leippold and Webersinke (2022) pre-trained BERT to analyze corporate climate change risk disclosure documents using documents related to climate change. Using the pre-trained BERT, they analyzed the actual state of information disclosure by companies before and after their support for the Task Force on Climate-related Financial Disclosures (TCFD). The analysis suggests that most companies increase the disclosure of information on items of low importance; however, it also suggests that disclosure of important information such as management strategy has not made much progress. Mittal, Chauhan and

[3] https://github.com/soskek/bookcorpus

Nagpal (2022) used BERT to create a sentiment index from news articles. They demonstrated that prediction errors are improved by incorporating the created sentiment index into price prediction. Kölbel, Leippold, Rillaerts and Wang (2020) built a BERT to judge transition and physical risks. Using the constructed BERT, they indexed the degree of corporate climate change risk disclosure and verified the relationship with corporate credit default swap (CDS) spreads. The results demonstrated that since the Paris Climate Agreement, companies' disclosure of transition risks leads to an increase in CDS spreads and the incorporation of risks. They also found that disclosure of physical risk leads to lower spreads and less uncertainty about the future. Sonkiya, Bajpai and Bansal (2021) used FinBERT (Araci, 2019) to create market sentiment indicators from news and comments on financial information sites. By incorporating the created sentiment index into the GAN model, they demonstrated a prediction accuracy that surpassed that of ARIMA, RNN, and a simple GAN model in predicting Apple's stock price.

3. Models

3.1. Language Model

In this study, we investigate the types of training method suitable to adapt a language model to the financial domain. The focus of our work is not proposing a new model; therefore we adopt the most widely used language model. We apply BERT, which has often been used in recent NLP tasks recently, as a primary language model. Considering that the most suitable training method differs between language models, we also apply ELECTRA.

3.1.1. BERT

BERT was proposed by Devlin et al. (2019), and it achieved the highest performance in many NLP tasks. The basic structure is a stack comprising 12 layers with the same form as the encoder layers of Transformer (Vaswani et al., 2017) . Two tasks are performed in the pre-training process of BERT, including the use of a masked language model (MLM) and next sentence prediction (NSP). The loss function is calculated as

$$\mathcal{L}_{\mathrm{BERT}} = \mathcal{L}_{\mathrm{MLM}} + \mathcal{L}_{\mathrm{NSP}}, \tag{1}$$

where $\mathcal{L}_{\mathrm{MLM}}$ and $\mathcal{L}_{\mathrm{NSP}}$ are the loss values of the task in the MLM and NSP tasks, respectively.

MLM is the task of predicting the original token before masking for 15% of the input tokens provided to BERT. Of the targeted 15% of the tokens, 80% are replaced with [MASK] tokens, 10% are replaced with random tokens, and the remaining 10% are not replaced. MLM is formulated as follows. Let $\boldsymbol{x} = [x_1, \cdots, x_n]$ be the input token, where n is the number of tokens. The hidden layer, which is the output obtained by inputting tokens into BERT, is expressed as $h_B(\boldsymbol{x}) \in \mathbb{R}^{n \times d_{\mathrm{model}}}$, where d_{model} is the embedding dimension of the hidden layers. For the index $t \in \{1, 2, \cdots, n\}$ where x_t is a masked token, the probability of the original input token is calculated as

$$p_B(x_t|\boldsymbol{x}) = \frac{\exp(e(x_t)^{\top} h_B(\boldsymbol{x})_t)}{\sum_{x' \in V} \exp(e(x')^{\top} h_B(\boldsymbol{x})_t)}, \tag{2}$$

where $e(x_t) \in \mathbb{R}^{d_{\mathrm{model}} \times |V|}$ is the embedding of the token x_t and V is the vocabulary of the model. The loss in the MLM is the negative log-likelihood, which is calculated as

$$\mathcal{L}_{\mathrm{MLM}} = \mathbb{E}\left(- \sum_{i \in \boldsymbol{m}} \log p_B(x_i|\boldsymbol{x}) \right), \tag{3}$$

where $\boldsymbol{m} \subset \{1, 2, \cdots, n\}$ is the set of indices such that the tokens x_t with $t \in \boldsymbol{m}$ are the masking targets, and $\mathbb{E}(\cdot)$ represents the expected value.

NSP is a binary classification task that determines whether two sentences are consecutive or randomly connected. 50% of the sentences in the entire dataset are set to be consecutive, and the remaining 50% of the sentences are not consecutive. A special token called the [SEP] token is inserted between the two sentences to be discriminated. In the task, the output of the [CLS] token inserted at the beginning of the input is used to predict whether the two target sentences are continuous. The NSP task is formulated as follows: Let $h_B(\boldsymbol{x})_1 \in \mathbb{R}^{d_{\mathrm{model}}}$ be the output of the [CLS] token (at the beginning of the sentence, the index of which is 1) of the BERT model obtained for the input token $\boldsymbol{x}$. The output of the NSP task is obtained from connecting three layers of multilayer perceptron (MLP) to this output as

$$s_{\mathrm{NSP}} = \tanh(W_1 h_B(\boldsymbol{x})_1 + b_1) \tag{4}$$

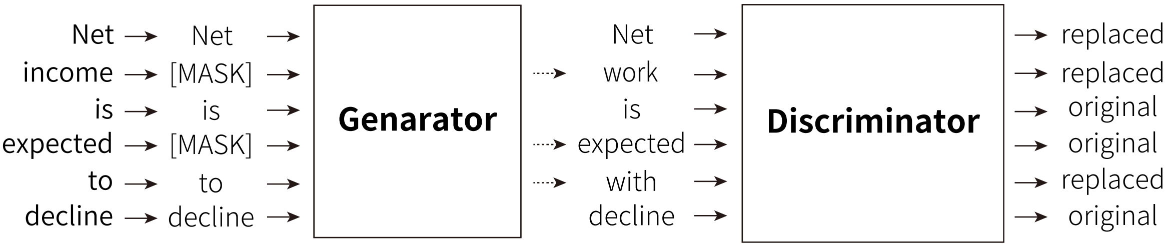


Figure 1: Overview of ELECTRA. The generator predicts the target tokens for masking. The discriminator determines whether the tokens from the generator were replaced with the original tokens. The three words "income," "expected," and "to" are subject to masking, and the two words "income" and "expected" are replaced with the [MASK] token and input into the generator. Although the word "to" is also subject to masking, it is not replaced with the [MASK] token in this example. The word "to" is input as-is, without being replaced with [MASK]. The generator predicts the three words as "work," "expected," and "with." Because the generator replaces "income" and "to" with "work" and "to," respectively, these words are labeled as "replaced." The other words are labeled "original" because they were not replaced from the original token.

$$y_{\text{NSP}} = \text{softmax}(W_2 s_{\text{NSP}} + b_2), \tag{5}$$

where $W_1 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}, W_2 \in \mathbb{R}^{2 \times d_{\text{model}}}, b_1 \in \mathbb{R}^{d_{\text{model}}}, b_2 \in \mathbb{R}^2$ are the trainable parameters. The loss function in NSP is calculated as

$$\mathcal{L}_{\text{NSP}} = \mathbb{E}\left(-l_{\text{NSP}} \log y_{\text{NSP},0} - (1 - l_{\text{NSP}}) \log y_{\text{NSP},1}\right), \tag{6}$$

where l_{NSP} takes 0 when the two target sentences are not actually consecutive, it takes 1 when they are continuous.

3.1.2. ELECTRA

ELECTRA is a model proposed by Clark et al. (2020), which has achieved higher performance than BERT with a comparable computational cost for the pre-training process. ELECTRA consists of two architectures, including a generator and a discriminator, as illustrated in Figure 1. Both the generator and the discriminator are the encoders of the Transformer, as with BERT. The generator is set to 1/4 (small model) or 1/3 (base size) the size of the discriminator in the hidden dimensions, feedforward network (FFN) hidden dimensions in the Transformer layers, and the number of the attention heads of the Transformer. This avoids an excessively strong generator that poses too great a challenge for the discriminator. In fine-tuning tasks, only the discriminator is used. As opposed to the original GAN (Goodfellow et al., 2014), the generator and the discriminator are pre-trained simultaneously rather than separately. In the pre-training process of ELECTRA, two tasks are performed, including an MLM task in the generator, which is almost the same as in BERT, and RTD in the discriminator. The loss function is calculated as

$$\mathcal{L}_{\text{ELECTRA}} = \mathcal{L}_{\text{MLM}} + \lambda \mathcal{L}_{\text{RTD}}, \tag{7}$$

where $\mathcal{L}_{\text{MLM}}$ and $\mathcal{L}_{\text{RTD}}$ are the loss values of the task in MLM and RTD respectively. λ is the weight for the discriminator objective in the loss, which we set to 50 according to Clark et al. (2020).

In the MLM task, 15% of input tokens are the targets for masking, which is the same as in BERT. Of these, 85% of the tokens are replaced with the [MASK] tokens, and the remaining 15% tokens are input into the generator with the original tokens. The generator predicts the tokens that were the original targets for masking. $\mathcal{L}_{\text{MLM}}$ is calculated in the same way as BERT (Equation 2 and 3).

In RTD, the discriminator is trained to detect replaced tokens. Let $e(x_t)^\top h_G(\boldsymbol{x})_t$ be the output of the generator neural networks except for the final softmax, where x_t is a masked token, $e(x_t) \in \mathbb{R}^{d_{\text{model}} \times |V|}$ is the embedding of the token x_t, V is the vocabulary of the model, and $h_G(\boldsymbol{x}) \in \mathbb{R}^{n \times d_{\text{model}}}$ is the hidden layer, which is the output from the generator. The sampled vector from the Gumbel distribution (Jang, Gu and Poole, 2017; Maddison, Mnih and Teh, 2017) is added to the output of the generator as

$$\boldsymbol{v}_t = e(x_t)^\top h_G(\boldsymbol{x})_t + \boldsymbol{g}_t, \tag{8}$$

where $\boldsymbol{v}_t = (v_{t,1}, v_{t,2}, \cdots, v_{t,\dim(V)}) \in \mathbb{R}^{\dim(V)}$ follows a categorical distribution with the Gumbel-max trick. $g_{t,j} \sim \text{Gumbel}(0,1)$ is the jth component of $\boldsymbol{g}_t$. $g_{t,j}$ is sampled from $g_{t,j} = -\log(-\log(u_{t,j}))$, where $u_{t,j} \sim \text{Uniform}(0,1)$. The token targeted for masking in the generator is replaced with the vocabulary of the index $s_t := \arg\max_j v_{t,j}$, which is the component j with the maximum component $v_{t,j}$ of $\boldsymbol{v}_t$. The non-masked input tokens are common between the generator and the discriminator inputs. The ith component x'_i of the input token $\boldsymbol{x'} = [x'_1, \cdots, x'_n]$ to the discriminator is calculated as

$$x'_i = \begin{cases} V(s_i) & (x_i \in \boldsymbol{m}) \\ x_i & (x_i \notin \boldsymbol{m}), \end{cases} \tag{9}$$

where $V(s_i)$ is the s_ith token of the vocabulary V. This equation is equivalent to replacing only the tokens targeted for masking in $\boldsymbol{x}$. Let the hidden layer from the discriminator and the input token sequence be $h_D(\cdot) \in \mathbb{R}^{n \times d_{\text{model}}}, x'_i (i = 1, \cdots, n)$, respectively. The ith probability output in RTD is expressed as

$$v_{D_i} = \text{GELU}(W_3 h_D(\boldsymbol{x'_i}) + b_{3,i}) \tag{10}$$

$$y_{D_i} = \sigma(w_4 v_{D_i} + b_{4,i}), \tag{11}$$

where $\sigma(\cdot)$ is a sigmoid function, $W_3 \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}, w_4 \in \mathbb{R}^{d_{\text{model}}}, b_{3,i} \in \mathbb{R}^{d_{\text{model}}}, b_{4,i} \in \mathbb{R}$ are the trainable parameters. Following Clark et al. (2020), GELU (Hendrycks and Gimpel, 2016) is adopted as the activation function. The loss function of the discriminator is calculated by

$$\mathcal{L}_{\text{RTD}}(\boldsymbol{x'}) = \mathbb{E}\left(\sum_{i=1}^{n} -\mathbb{1}(x'_i = x_i) \log y_{D_i} - \mathbb{1}(x'_i \neq x_i) \log(1 - y_{D_i}) \right), \tag{12}$$

where $\mathbb{1}(\cdot)$ indicates that only the elements in parentheses the conditions of which are positive are used. At that time, the tokens correctly predicted by the generator are labeled as the original tokens.

As with BERT, to improve calculation efficiency, tokens are added at the beginning and the end of the input until the upper limit of 128/512 tokens is reached.

3.2. Tokenizer

BERT and ELECTRA use an English corpus. When tokenizing a sentence with these language models, a sentence is divided by space separators, and then subwords are divided by WordPiece (Schuster and Nakajima, 2012). The method of dividing words into subwords aims to improve performance by reducing the number of unknown words and vocabulary by dividing the input sentence into units smaller than words. In addition to WordPiece, byte-pair encoding (BPE) (Sennrich, Haddow and Birch, 2016) and SentencePiece (Kudo and Richardson, 2018) tokenizers are also used as subword division methods. Here, we apply WordPiece, which is used in the original BERT. In contrast to English, Japanese writing cannot be divided into words by spaces. We use MeCab (Kudo, Yamamoto and Matsumoto, 2004) as a morphological analyzer to perform word division. To implement the proposed approach, we use Hugging Face's Tokenizers library.[4] For the parameter settings for subword division, we refer to the Japanese BERT model.[5] The number of words in the vocabulary is 32,768. Five tokens in the vocabularies such as [UNK], [CLS], [SEP], [PAD], and [MASK] represent an unknown token, the beginning of a sentence, the space between sentences and the end of an input, a padding input to align the input length, and a masked token for the MLM task, respectively. 6,129 words are set to be included in the vocabulary for a single word to recognize many Chinese characters peculiar to Japanese.

In this study, we construct two types of tokenizers, a Wikipedia tokenizer and a Fin & Wikipedia tokenizer. The Wikipedia tokenizer is constructed from the Wikipedia corpus, whereas the Fin & Wikipedia tokenizer is constructed by combining a financial corpus and Wikipedia. Both the corpora are described in the section 3.3. Table 1 displays an example of the tokenization of words related to finance by the constructed Wikipedia and Fin & Wikipedia tokenizers. The words that would be naturally recognized by a single token are divided into multiple tokens by the Wikipedia tokenizer, while they are recognized as a single token by the Fin & Wikipedia tokenizer. Other settings such as the word division, the number of words, the special tokens, etc. used to build the two tokenizers are common to the tokenizers; the only difference between the tokenizers is the corpus they are built on.

[4] https://github.com/huggingface/tokenizers

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

Table 1
Example of word tokenization for each tokenizer.

Word	Wikipedia Tokenizer	Fin & Wikipedia Tokenizer
デリバティブ (derivative)	デリ/バ/ティブ (deri/va/tive)	デリバティブ (derivative)
コンプライアンス (compliance)	コンプ/ライ/アンス (comp/li/ance)	コンプライアンス (compliance)
ポートフォリオ (portfolio)	ポート/フォ/リオ (port/fo/lio)	ポートフォリオ (portfolio)
ガバナンス (governance)	ガ/バナ/ンス (go/verna/nce)	ガバナンス (governance)

3.3. Corpus

For the corpus for pre-training, we use the following three types of text data in Japanese:

- Financial results
  - Disclosed from October 9, 2012, to December 31, 2020
  - Released for a certain period of time on TDnet[6]
- Securities reports
  - Disclosed from February 8, 2018 to December 31, 2020
  - Released for a certain period of time on EDINET[7]
- Wikipedia
  - Version on June 1, 2021
  - Approximately 20 million sentences (approximately 2.9 GB in size)
  - Downloaded from Wikimedia webpage[8]

A financial corpus is created from two datasets including financial results and securities reports (approximately 27 million sentences and 5.2 GB in size).

4. Experiments

We investigate methods suitable for adapting language models to financial domains. There are two main ways to fit language models into the financial domain. The first is to pre-train them from scratch using a corpus of text related to the financial domain (Liu et al., 2020). The second involves further pre-training, which continues with a financial corpus after training with a general corpus (Araci, 2019). We investigate which of these methods performs better for downstream financial tasks. Furthermore, when constructing the language models, we vary the corpus used for pre-training, the corpus for the tokenizer, and the architecture of the language model. Furthermore, we investigate the impact of these effects on the performance of downstream tasks.

4.1. Experimental Setup

Table 2 presents the models constructed in this study. We construct models for both small and base sizes, except where noted otherwise. The differences between the sizes are described below: First, we construct models that use the Wikipedia corpus and tokenizers with the BERT architecture as the baseline (Model 1). We use an existing model for the base size.[9] Additionally, we construct three models (Models 2-4) as pre-trained BERT models using the financial

[6] https://www.release.tdnet.info/inbs/I_main_00.html

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

[8] https://dumps.wikimedia.org/

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

Table 2
Models constructed in this study. For the small size, we construct all the models in this table. For the base size, we also construct all the models except Model 1. In the column listing the pre-training types, PT indicates pre-training and DAPT refers to domain-adaptive pre-training, respectively.

#	Architecture	Pre-training Type	Corpus	Tokenizer Corpus
1	BERT	PT	Wikipedia	Wikipedia
2	BERT	PT	Fin & Wikipedia	Fin & Wikipedia
3	BERT	PT	Wikipedia	Fin & Wikipedia
4	BERT	PT	Fin & Wikipedia	Wikipedia
5	BERT	DAPT	Wikipedia→Fin	Wikipedia
6	BERT	DAPT	Wikipedia→Fin	Fin & Wikipedia
7	ELECTRA	PT	Wikipedia	Wikipedia
8	ELECTRA	PT	Fin & Wikipedia	Fin & Wikipedia

corpus. Since the performance of the model using only the financial corpus without Wikipedia was lower in our fine-tuning experiments, we combine both the financial corpus and Wikipedia for pre-training and to construct a tokenizer (Model 2). The second model uses Wikipedia for pre-training and the financial corpus and Wikipedia for the tokenizer (Model 3). The third model uses the financial corpus and Wikipedia for pre-training, and Wikipedia alone for the tokenizer (Model 4).

For domain-adaptive pre-training (DAPT) models, we also construct two BERT models trained firstly on Wikipedia and then trained further on the financial corpus (Models 5 and 6). The tokenizer of the first of these models is Wikipedia (Model 5) and the other is trained on the financial corpus and Wikipedia (Model 6).

We also construct two pre-trained models using the ELECTRA architecture (Models 7 and 8). For pre-training and tokenizer, the first of these models uses Wikipedia for pre-training and the tokenizer (Model 7), and the other one uses the financial corpus and Wikipedia (Model 8).

We adopt the Wikipedia BERT model (#1 in Table 2) as a baseline to compare the methods of adapting language models to the financial domain with pre-training methods and vocabulary adaptation. Since the proposal of BERT, various state-of-the-art models have been proposed. Therefore, in this study, in addition to the Wikipedia BERT model, we also adopt the base-size models of RoBERTa[10] and ALBERT[11] as additional baseline models. Although these models are pre-trained on Japanese general corpora, the RoBERTa model uses Wikipedia and CC-100[12] as corpora for pre-training, Juman++[13] for word segmentation, and SentencePiece for subword segmentation. Additionally, the ALBERT model uses Wikipedia and livedoor news corpora[14] (Japanese online news corpora) for pre-training, and SentencePiece for subword segmentation (word segmentation is not performed). It is difficult to make a simple comparison because the settings for constructing the language model are different.

The parameters of each model are set as displayed in Table 3 with reference to Clark et al. (2020). Following the description in the GitHub repository of BERT[15], we set the learning rate to 20% of the pre-training for domain-adaptive pre-training. The number of training steps is the same as that for pre-training. The learning rate is warmed up to 10,000 steps and then decreased linearly.

4.2. Implementation

The models are implemented using the PyTorch-based[16] framework. The PyTorch library includes a package for distributed training.[17] However, this package assumes that the same number of GPUs with the same amount of memory is used for each node. In this case, the batch size must be adjusted to the GPU with the smallest memory among the GPUs used in distributed training, and more GPUs are required to construct a given model. Therefore, in this study, we

[10] https://huggingface.co/nlp-waseda/roberta-base-japanese

[11] https://huggingface.co/ken11/albert-base-japanese-v1-with-japanese-tokenizer

[12] https://data.statmt.org/cc-100/

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

[14] https://www.rondhuit.com/download.html#ldcc

[15] https://github.com/google-research/bert

[16] https://github.com/huggingface/transformers, https://pytorch.org/

[17] https://pytorch.org/docs/stable/distributed.html

Table 3
Parameters used to pre-train constructed models. The learning rate displays the value in 10,000 steps, and the value in parentheses is the learning rate at the time of domain-adaptive pre-training. Generator size is given by the ratio of the size of the discriminator to that of the discriminator in the ELECTRA model, and "-" means that an entry does not include the structure of the generator because it does not follow the ELECTRA architecture.

Parameters	BERT-small	ELECTRA-small	BERT-base	ELECTRA-base
Number of layers	12	12	12	12
Hidden Size	256	256	768	768
FFN Size	1024	1024	3072	3072
Attention Heads	4	4	12	12
Embedding Size	128	128	512	512
Learning Rate	5e-4 (1e-4)	5e-4	1e-4 (2e-5)	2e-4
Batch Size	128	128	256	256
Generator Size	-	1/4	-	1/3
Train Steps	1.45M	1M	1M	766K

vary the implementation of the library to enable the use of GPUs with different capacities for each node. The source code used to construct the models is published online[18] along with the constructed pre-trained models.[19]

5. Evaluation

The general language understanding evaluation (GLUE) benchmark (Wang, Singh, Michael, Hill, Levy and Bowman, 2018) is typically used to evaluate general language models. However, general language tasks like GLUE cannot evaluate financial domain-specific models. In English, financial language tasks such as sentiment analysis (Malo, Sinha, Korhonen, Wallenius and Takala, 2014; Cortis, Freitas, Daudert, Huerlimann, Zarrouk, Handschuh and Davis, 2017) and causality detection (Mariko, Abi-Akl, Labidurie, Durfort, De Mazancourt and El-Haj, 2020) have been established.

In this study, we evaluate the performance of language models in terms of three tasks: aspect-based sentiment analysis, section prediction, and causality detection. As these tasks are common in finance, we are able to confirm the effectiveness of the pre-training methods and tokenizers from the perspective of their adaptability to these tasks.

The following settings are used in all the three experiments: The batch size is set to 32. We search for hyperparameters from 1 to 10 for the number of epochs, from 10^{-6} to 10^{-4} for the learning rate of the language model (BERT or ELECTRA), and from 10^{-5} to 10^{-3} for the learning rate of the classification layers. We use PyTorch[20] (version 1.8.1) to implement the models, Optuna[21] (version 2.0.0) for hyperparameter optimization and a cross-entropy loss function, and Adam (Kingma and Ba, 2015) as the optimization algorithm. We perform five trials of hyperparameter searches for each dataset. We use the test F1 with the highest validation F1 among these five trials as the result of a single experiment. For each of the three evaluation experiments and each model, we perform 25 experiments by repeating five-fold cross-validation five times with different seeds. Of the datasets used for each evaluation experiment, 20% are used as the testing data. Of the remaining 80%, 15% are assigned as the validation data and the remainder are assigned as the training data.

Aspect-Based Sentiment Analysis

Aspect-based sentiment analysis (ABSA) is a task that predicts sentiment based on an input sentence. In ABSA, even if the same sentence is input, different targets may evoke different sentiments. For example, the sentence "The increase in comparable-store sales for the thirteen weeks ending November 3, 2018 was driven by strong sales in our toys, housewares, electronics, and floor covering departments, partially offset by a decrease in our furniture and food departments," has two sentiments. If we select "toys" as targets, we may obtain a positive sentiment. By contrast, if we select "food departments" as a target, we may obtain a negative sentiment.

[18] https://github.com/retarfi/language-pretraining/tree/v1.0

[19] https://huggingface.co/izumi-lab

[20] https://github.com/pytorch/pytorch

[21] https://github.com/optuna/optuna

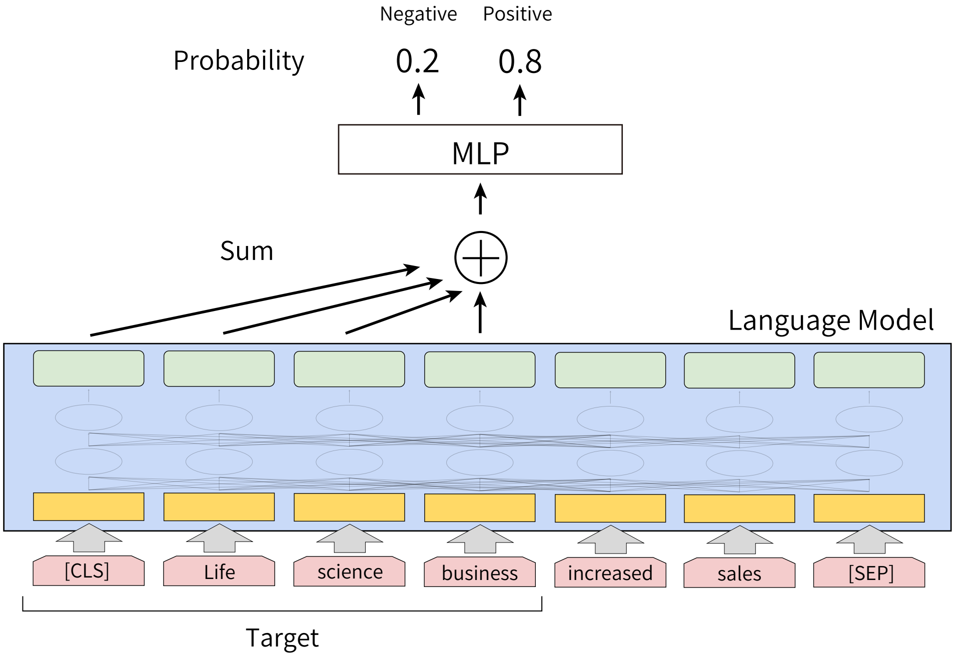


Figure 2: Overview of aspect-based sentiment analysis. The input is the sentence "Life science business increased sales" and the sentiment target is "life science business." The input sentence is divided into tokens as "Life / science / business / increased / sales" and input to a language model. The output of the [CLS] token and the outputs of the target "life," "science," and "business" tokens are summed as the output of the hidden layer in the last layer of BERT or ELECTRA. The summed vector is processed by MLP and softmax to make positive or negative predictions for the target word.

Here, we employ the chABSA dataset[22] as the ABSA dataset. The chABSA dataset is created using Japanese annual securities reports from 2014 to 2018. These annual Japanese securities reports are available online.[23] The dataset is tagged with positive, negative, and neutral sentiments according to the target words. However, we exclude neutral sentences in the experiment because there are significantly fewer neutral tags than other tags. In total, 7,465 sentences are used.

Figure 2 illustrates an overview of the experiment formulated as follows: Let $\boldsymbol{x} = [x_1, x_2, \cdots, x_n]$ be an input token sequence. In BERT and ELECTRA, x_1 is the [CLS] token inserted at the beginning of the sentence, and x_n is the [SEP] token inserted at the end of the sentence or at the connection. Let $S \subset \{2, \cdots, n-1\}$ and $\mathrm{LM}(\boldsymbol{x}) \in \mathbb{R}^{n \times d_{\mathrm{model}}}$ be the set of index numbers of the target token sequence of the sentiment prediction and the output of the hidden layer of the final layer of the LM (BERT or ELECTRA) obtained from the input token sequence $\boldsymbol{x}$ respectively. In this set, d_{model} is the number of the dimensions of the hidden layer of the language models. Let $\mathrm{LM}(\boldsymbol{x})_i \in \mathbb{R}^{d_{\mathrm{model}}}$ be the ith component of $\mathrm{LM}(\boldsymbol{x})$. The outputs of the [CLS] token and the target tokens from the language model are summed as follows:

$$c_{\mathrm{ABSA}} = \sum_{i \in (\{1\} \cup S)} \mathrm{LM}(\boldsymbol{x})_i. \tag{13}$$

The positive and negative probabilities are the outputs from the three-layer MLP as follows:

$$y_{\mathrm{ABSA}} = softmax(W_6(\tanh(W_5 c_{\mathrm{ABSA}} + b_5)) + b_6), \tag{14}$$

where $W_5 \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{model}}}$, $W_6 \in \mathbb{R}^{2 \times d_{\mathrm{model}}}$, $b_5 \in \mathbb{R}^{d_{\mathrm{model}}}$, $b_6 \in \mathbb{R}^2$ are trainable parameters.

[22] https://github.com/chakki-works/chABSA-dataset
[23] https://github.com/chakki-works/CoARiJ

Section Prediction

Section prediction classifies an input sentence into a defined written section in a given document. We used a Japanese summary of financial statements[24] for the section prediction. This summary is published once every quarter and consists of sections including "business results," "financial status," "future forecast information," "balance sheet," "income statement," "cash flow," and "notes on financial statements." Each section includes sentences that explain the numerical information and content. In this task, methods use these sentences as input and predict the section to which they belong in a given document. In this study, we use "business results," "financial status," "future forecast information," and "notes on financial statements" as sections; therefore, we addressed classification tasks with four labels. We exclude other sections because they mainly contained numerical information with relatively little text data.

We use data from 839 companies' summaries of financial statements published from December 2021 to February 2022. From these data, we extract 28,291 sentences, comprising 8,350 sentences in the "business results" section, 4,258 sentences in the "financial status" section, 8,422 sentences in the "future forecast information" section, and 7,261 sentences in the "notes on financial statements" section.

The experiment is formulated as follows: Let the output of the [CLS] token (index 1) of the LM be $\mathrm{LM}(\boldsymbol{x})_1$, where $\boldsymbol{x}$ is the input token sequence. The probabilities of the sections are output from the three-layer MLP as follows:

$$y_{\mathrm{SP}} = \mathrm{softmax}(W_8(\tanh(W_7\,\mathrm{LM}(\boldsymbol{x})_1 + b_7)) + b_8), \tag{15}$$

where $W_7 \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{model}}}$, $W_8 \in \mathbb{R}^{4 \times d_{\mathrm{model}}}$, $b_7 \in \mathbb{R}^{d_{\mathrm{model}}}$, $b_8 \in \mathbb{R}^4$ are trainable parameters.

Causality Detection

Causality detection classifies sentences according to whether they include a causal relation. We prepare a dataset of Japanese newspaper articles to detect causality. Specifically, we randomly collect 2,045 sentences from Nikkei news articles between 1995 and 2005, and five annotators tag the collected sentences. In this dataset, we define the sentences determined by three or more annotators as causal sentences. Of these, 68%, 12%, and 20% are used for the training, validation, and testing data, respectively.

For example, the sentence "From the economic downturn, store sales fell" includes causality. On the other hand, "I took a direct flight from Japan to Hawaii" does not include a causal relationship. The causality detection dataset includes sentences similar to those above.

The experiment is formulated as follows: Let the output of the [CLS] token (index 1) of the LM be $\mathrm{LM}(\boldsymbol{x})_1$, where $\boldsymbol{x}$ is the input token sequence. The probability of whether a sentence containing causality is output from the three-layer MLP is given as

$$y_{\mathrm{CD}} = \mathrm{softmax}(W_{10}(\tanh(W_9\,\mathrm{LM}(\boldsymbol{x})_1 + b_9)) + b_{10}), \tag{16}$$

where $W_9 \in \mathbb{R}^{d_{\mathrm{model}} \times d_{\mathrm{model}}}$, $W_{10} \in \mathbb{R}^{2 \times d_{\mathrm{model}}}$, $b_9 \in \mathbb{R}^{d_{\mathrm{model}}}$, $b_{10} \in \mathbb{R}^2$ are the trainable parameters.

6. Results

Tables 4 and 5 present the results for the small and base models, respectively. When we compare ABSA, section prediction, and causal detection, ABSA tends to achieve the highest score; however, this is not statistically significant. The column "Average" displays the average scores for the three tasks. The PT model with the Fin & Wikipedia corpora for pre-training and the tokenizer (Model 2) and the DAPT model with the Fin & Wikipedia corpus for pre-training and the tokenizer (Model 6) perform relatively well on average. However, the statistical significance between these models is not observed at the 1% level.

7. Discussion

First, we discuss the performances of pre-training (PT) and domain-adaptive pre-training (DAPT). Both the DAPT BERT model and the PT model with Fin & Wikipedia corpora and Fin & Wikipedia tokenizers (Models 2 and 6) achieve higher performance than the Wikipedia PT BERT model (Model 1). This demonstrates that domain adaptation using both the PT and DAPT methods is effective. Although there is no significant difference between Models 2 and 6, the DAPT model (Model 6) achieves slightly higher performance than the PT BERT model with the Fin &

[24] https://www.release.tdnet.info/inbs/I_main_00.html

Table 4
Results of small models. In the column of pre-training types, PT indicates pre-training and DAPT indicates domain-adaptive pre-training, respectively. The top scores for the tasks and the average between the tasks are displayed in bold. ** indicates significance at 1% compared to the average result of Model 6, which is the highest in the table.

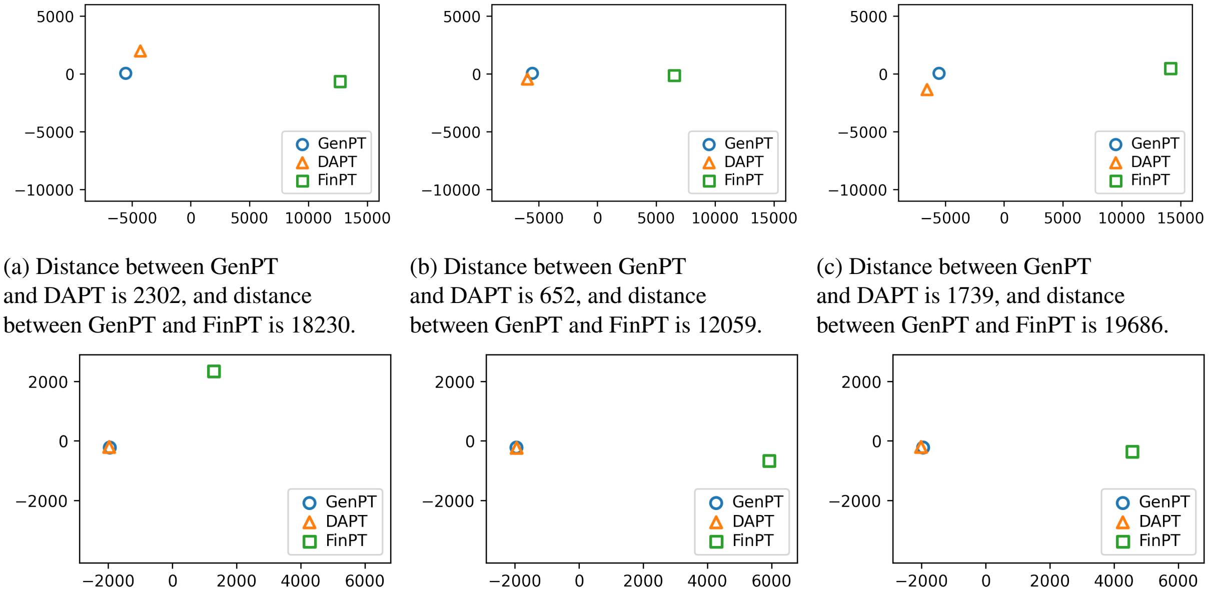
#	Architecture	Pre-training Type	Corpus	Tokenizer Corpus	ABSA	Section Prediction	Causal Detection	Average
1	BERT	PT	Wikipedia	Wikipedia	.924	.695	.882	.834**
2	BERT	PT	Fin & Wikipedia	Fin & Wikipedia	.929	.702	**.888**	.840
3	BERT	PT	Wikipedia	Fin & Wikipedia	.927	.696	.884	.835**
4	BERT	PT	Fin & Wikipedia	Wikipedia	.925	.701	.878	.835**
5	BERT	DAPT	Wikipedia→Fin	Wikipedia	.928	.702	.879	.836**
6	BERT	DAPT	Wikipedia→Fin	Fin & Wikipedia	**.932**	**.705**	**.888**	**.842**
7	ELECTRA	PT	Wikipedia	Wikipedia	.919	.697	.874	.830**
8	ELECTRA	PT	Fin & Wikipedia	Fin & Wikipedia	.917	.699	.874	.830**

Table 5
Results of base models. In the column of pre-training types, PT indicates pre-training and DAPT indicates domain-adaptive pre-training, respectively. The top scores for the tasks and the average between the tasks are displayed in bold. ** and * indicate significance at 1% and 5% compared to the average result of Model 5, respectively, which is the highest in the table, respectively.

#	Architecture	Pre-training Type	Corpus	Tokenizer Corpus	ABSA	Section Prediction	Causal Detection	Average
1	BERT[25]	PT	Wikipedia	Wikipedia	.936	.699	.880	.838**
2	BERT	PT	Fin & Wikipedia	Fin & Wikipedia	.942	.703	.882	.842
3	BERT	PT	Wikipedia	Fin & Wikipedia	.938	.700	.883	.840*
4	BERT	PT	Fin & Wikipedia	Wikipedia	.942	.703	.875	.840*
5	BERT	DAPT	Wikipedia→Fin	Wikipedia	**.946**	.704	.881	**.844**
6	BERT	DAPT	Wikipedia→Fin	Fin & Wikipedia	.945	**.705**	.881	.843
7	ELECTRA	PT	Wikipedia	Wikipedia	.935	.701	.876	.838**
8	ELECTRA	PT	Fin & Wikipedia	Fin & Wikipedia	.944	.703	.880	.842
-	RoBERTa[26]	PT	Wikipedia	Wikipedia	.942	.701	**.886**	.843
-	ALBERT[27]	PT	Wikipedia	Wikipedia	.910	.696	.882	.830**

Wikipedia corpus for pre-training and with the same tokenizers (Model 2). This might occur because the model can be better adapted by inputting the corpus of the target domain after acquiring knowledge of the general language with the pre-training process to some extent. Figure 3 illustrates the results of the principal component analysis (PCA) of the output of the PT and DAPT BERT models with a linear transformation. A linear transformation is performed to bring the latent space of the pre-trained model with the financial corpus (FinPT) and DAPT model closer to that of the pre-trained model using Wikipedia (GenPT) with a common vocabulary as an anchor. Here, we describe the transformation of the FinPT model output. Of the tokens that overlapped in the vocabulary of both the FinPT and GenPT model tokenizers, the d_{model} tokens are extracted from the longest to the shortest. d_{model} is the hidden size of the model in Table 3 with 256 for the small model and 768 for the base model. We add [CLS] and [SEP] tokens before and after these tokens and entered them into each model. Let $h_{\text{FinPT},1}, \cdots, h_{\text{FinPT},d_{\text{model}}}$ be the outputs at the position of the [CLS] token respectively, where $h_{\text{FinPT},i}$ satisfies $h_{\text{FinPT},i} \in \mathbb{R}^{d_{\text{model}}} (i \in \{1, 2, \cdots, d_{\text{model}}\})$. Similarly, the outputs from the GenPT model for d_{model} tokens are also referred to as $h_{\text{GenPT},1}, \cdots, h_{\text{GenPT},d_{\text{model}}}$. The linear transformation of the output from the FinPT model is expressed using $W_{\text{FinPT}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$ as follows:

$$\begin{aligned} h_{\text{GenPT},1} &= W_{\text{FinPT}} h_{\text{FinPT},1}, \\ &\vdots \\ h_{\text{GenPT},d_{\text{model}}} &= W_{\text{FinPT}} h_{\text{FinPT},d_{\text{model}}}. \end{aligned} \tag{17}$$



(a) Distance between GenPT and DAPT is 2302, and distance between GenPT and FinPT is 18230.

(b) Distance between GenPT and DAPT is 652, and distance between GenPT and FinPT is 12059.

(c) Distance between GenPT and DAPT is 1739, and distance between GenPT and FinPT is 19686.

(d) Distance between GenPT and DAPT is 29, and distance between GenPT and FinPT is 4138.

(e) Distance between GenPT and DAPT is 10, and distance between GenPT and FinPT is 7896.

(f) Distance between GenPT and DAPT is 69, and distance between GenPT and FinPT is 6525.

Figure 3: Mapping the small and base models for sentences. GenPT is pre-trained with both the Wikipedia corpus and the Wikipedia tokenizer. DAPT is pre-trained with the financial corpus from GenPT. FinPT is pre-trained with the Fin & Wikipedia corpora and Fin & Wikipedia tokenizer. (a), (d): Mapping the sentence "デリバティブ取引は、先物取引、オプション取引、スワップ取引などの総称。 (Derivatives trading is a general term for futures trading, options trading, swap trading, etc.)" with (a) the small models and (d) the base models. (b), (e): Mapping the sentence "資産運用とは、自分の持っているお金を預貯金や投資に配分することで効率的にふやしていくこと。 (Asset management is to efficiently increase the amount of funds you have by allocating capital to deposits, savings, and investments.)" with (b) the small models and (e) the base models. (c), (f): Mapping the sentence "投資とは利益を見込んでお金を出すこと。 (Investing is to put money in anticipation of profit.)" with (c) the small models and (f) the base models.

Equation (17) can be expressed as follows:

$$H_{\text{GenPT}} = W_{\text{FinPT}} H_{\text{FinPT}}, \tag{18}$$

where $H_{\text{GenPT}} = \left[h_{\text{GenPT},1}, \cdots, h_{\text{GenPT},d_{\text{model}}}\right] \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$, $H_{\text{FinPT}} = \left[h_{\text{FinPT},1}, \cdots, h_{\text{FinPT},d_{\text{model}}}\right] \in \mathbb{R}^{d_{\text{model}} \times d_{\text{model}}}$. In this transformation, $h_{\text{FinPT},i}$ are independent of one another. For the DAPT model, a weight matrix is defined, and the conversion is performed in the same manner as for FinPT. We perform this transformation only to observe the trends in the models. Therefore, the result may not be an accurate calibration but it is meaningful.

When we compare the distance between the PT models and the distance between the DAPT model and Wikipedia PT model in Figure 3, we observe that the latter is shorter in each sentence. This demonstrates that the DAPT model is closer to the Wikipedia PT model because the DAPT model is further pre-trained from the Wikipedia PT model. We consider that the DAPT model can improve its performance by further adaptation in the domain while inheriting the useful parts of the PT model.

DAPT also exhibits an advantage in terms of corpus scaling. Here, Japanese Wikipedia (approximately 2 GB) is used as the general language corpus. In the case of aiming to improve model performance by using a larger general corpus such as Japanese CC-100[28] (approximately 15 GB), the degree of domain adaptation in the PT model becomes smaller because the variety of the financial corpus in pre-training is relatively reduced, unless the larger financial corpus is adopted. By contrast, DAPT is expected to improve the accuracy of the pre-training stage. DAPT may be considered more suitable if the proportion of the target domain corpus is relatively low. The performance difference between PT

[28] https://data.statmt.org/cc-100/

and DAPT may also change depending on the ratio of the size of the corpus of the target domain to the general corpus. In future studies, the effects of varying this ratio should be investigated and discussed.

For the corpus used for the tokenizer and pre-training, adding the financial corpus improves performance in the downstream tasks. According to these results, the average accuracies in the pre-trained BERT models are improved when the pre-training corpus is changed from Wikipedia to Fin & Wikipedia and when the corpus of the tokenizer is changed from Wikipedia to Fin & Wikipedia. Note that the only difference between the Wikipedia tokenizer and Fin & Wikipedia tokenizer is the corpus used to build the tokenizer, and the other settings used to build the tokenizer are the same. There are 29,622 tokens in the two tokenizers (vocabularies) that are included in both. This implies that 90.4% of the tokens are duplicates. As only two financial corpora, financial results and securities reports, are used, we add Wikipedia to construct the vocabulary to ensure lexical diversity. However, a more specific tokenizer for the financial domain might be constructed by limiting only the financial corpora for the construction of the tokenizer. Additionally, we construct a tokenizer from the same corpus used for pre-training. Ensuring diversity in the financial corpus sources for the tokenizer is a future challenge given that the performance of the causal detection task using the newspaper article dataset decreased due to tokenizer adaptation.

Changing both the pre-training corpus and the corpus of the tokenizer from Wikipedia to Fin & Wikipedia results in a greater improvement in the average accuracy. Improvement of the performance by the adaptation of the pre-training corpus is consistent with previous studies (Araci, 2019; Liu et al., 2020; Peng et al., 2021). The pre-trained BERT models with the Fin & Wikipedia corpus in the tokenizer and pre-training achieve the highest average accuracy among the four pre-trained BERT models with different corpora. Between the domain-adaptive pre-trained BERT small models, the model using the Fin & Wikipedia corpora for the tokenizer also had higher average accuracy than the model using the Wikipedia corpus. However, Peng et al. (2021) reported that in English downstream tasks, using a financial corpus for pre-training was effective but not for the tokenizer vocabulary. This may be attributed to differences in the linguistic characteristics between Japanese and English. As displayed in Table 1, the Wikipedia tokenizer divides a word that is naturally recognized as a single token into multiple tokens, whereas the Fin & Wikipedia tokenizer recognizes a word as one token. Each of the words written in English is recognized as one token by the tokenizer trained on the general corpus.[29] Therefore, a significant effect on the adaptation of the tokenizer may not be observed in the English language (Peng et al., 2021). Owing to the difference between the tokenization by the Wikipedia tokenizer and that by the Fin & Wikipedia tokenizer (in Japanese), the tokenizer also has the effect of adapting the model to the domain as the pre-training corpus. Whether this difference, which was not observed in English, can be observed in other languages remains a topic for further study.

In the causal detection task, the effect of adapting the training corpus is considered small. This is guided by a comparison between Model 1, whose training corpus is Wikipedia, and Models 4 and 5, which use the financial corpus as the training corpus. In particular, except for Model 5 with the base size, the adaptation of the pre-training corpus degrades performance. This may be because the dataset used for the causal detection task is newspaper articles related to economics. Wikipedia is more suitable for the causal detection task as the pre-training corpus than the financial corpus specialized in financial terminology. The financial corpus used for pre-training in this study is only financial statements and securities reports. Therefore, constructing a tokenizer (vocabulary) using a wider range of text data in the financial domain is more suitable for such tasks. This could make FinPT's plot in Figure 3 closer to that of the DAPT and GenPT, and the model would demonstrate a wide range of high performance in the financial domain.

Additionally, when comparing the model sizes, in all the BERT models from Model 1 to Model 6, many of the base-size models slightly underperform the small-size models. In general, the larger the model size, the better the performance in downstream tasks (Tay, Dehghani, Rao, Fedus, Abnar, Chung, Narang, Yogatama, Vaswani and Metzler, 2022). The tendency of small-sized models to outperform base-sized models may have been caused by the differences in the maximum input length of the number of tokens in the model and the distribution of the number of tokens in the dataset. The maximum number of input tokens for the small- and base-size models used in this study is 128 and 512, respectively. The maximum number of tokens in the dataset used in the causal detection task is 512 or more. Therefore, the maximum input length of tokens in the experiment is set to 128 and 512 for small size and base sizes, respectively. However, 99.8% of the samples in the dataset have 128 tokens or less. This would make the small-sized models demonstrate higher performance in the causal detection task.

In the section prediction task, adapting both the pre-training corpus and vocabulary (tokenizer) is effective. From the comparison of Models 1, 3, and 4, it is considered that the adaptation of the pre-training corpus is more effective

[29] `https://huggingface.co/bert-base-uncased`

in terms of performance than the adaptation of the vocabulary. This finding suggests that understanding the financial context is more important than recognizing financial words in this task.

In the ABSA task, the performance difference between the small size and the base size is large compared to the other two tasks. It is considered that increasing the model size progresses the understanding of not only the financial words but also the financial context. Better performance in downstream tasks is generally achieved by a larger model size (Tay et al., 2022). Therefore, this task is considered the most suitable among the three evaluation tasks in this study to measure the performance of the models.

Compared to the pre-trained BERT and ELECTRA models using the Wikipedia or Fin & Wikipedia corpus, the BERT model exhibited the same or higher average accuracy than the ELECTRA model. However, Clark et al. (2020) reported that ELECTRA achieved higher performance than BERT with the same amount of computation. This may have occurred because the pre-training task (RTD) or architecture of ELECTRA does not fit Japanese compared to those of BERT, whereas it was designed to fit English. We leave this point as a problem of interest to be explored in future research.

With regard to the ALBERT and RoBERTa models as baseline models other than BERT and ELECTRA, ALBERT has lower performance than the BERT model (Model 1). This is consistent with the results of Lan et al. (2019), where ALBERT performs worse than BERT for the same base model size. By contrast, RoBERTa performs better than the BERT model (Model 1). This is also consistent with the results of Liu et al. (2019b), where RoBERTa outperforms BERT at the same base model size. These trends are also observed in each of the three tasks in the same manner. From this, it is considered that these three tasks are suitable not only for comparison of the corpus used for model construction but also for comparison of the model structure and training method, similar to NLP benchmark tasks such as GLUE (Wang et al., 2018). We use BERT as the base model architecture in this study, and the RoBERTa model, which is pre-trained using general corpora, outperforms the BERT model constructed solely from Wikipedia. We leave the construction of RoBERTa models (FinPT and DAPT) that adapt the tokenizer and pre-training corpus to the financial domain for the future.

8. Conclusion

In this study, we consider the learning processes of domain-specific language models in the finance domain, particularly in Japanese, and discussed a methodology to solve their incompatibility problems. We construct several models of various architectures using different pre-training methods, pre-training corpora, and tokenizer corpora for the financial domain. We observe that a word split into multiple tokens by the tokenizer constructed from the general corpus is recognized as a single token by the tokenizer constructed from the financial corpora. The models are evaluated in terms of several typical financial tasks. When we compare a pre-training model trained with the financial corpus and a DAPT model trained with the financial corpus after training with the general corpus, we find that the latter achieved a slightly higher fine-tuning performance. However, no significant differences are not observed between the models. We also observe that the performance of the model improved when both the pre-training corpus and the tokenizer corpus are adopted for finance.

In future work, we plan to investigate the performance of similar models for different size ratios of the financial corpus to the general corpus using a larger general corpus such as the CC-100. The findings of this study may be extended to RoBERTa and other post-BERT state-of-the-art (SOTA) models. We also plan to conduct an experiment to observe whether the effect of the domain-adaptive tokenizer can be observed in other languages. Because most of the vocabulary of the financial domain-adapted tokenizer overlapped with the vocabulary of the tokenizer based on the general corpus, a tokenizer with more financial domain vocabulary may achieve higher performance on financial domain tasks.

CRediT authorship contribution statement

Masahiro Suzuki: Conceptualization of this study, Methodology, Software, Formal analysis, Investigation, Resources, Writing, Visualization. **Hiroki Sakaji:** Validation, Supervision, Project administration, Writing. **Masanori Hirano:** Resources, Data Curation, Writing. **Kiyoshi Izumi:** Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Number JP21K12010, and the JST-Mirai Program Grant Number JPMJMI20B1, Japan.

References

Akita, R., Yoshihara, A., Matsubara, T., Uehara, K., 2016. Deep learning for stock prediction using numerical and textual information, in: 2016 IEEE/ACIS 15th International Conference on Computer and Information Science (ICIS), pp. 1–6. doi:10.1109/ICIS.2016.7550882.

Araci, D., 2019. FinBERT: Financial Sentiment Analysis with Pre-trained Language Models. doi:10.48550/arXiv.1908.10063.

Aroca-Ouellette, S., Rudzicz, F., 2020. On Losses for Modern Language Models, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics. pp. 4970–4981. doi:10.18653/v1/2020.emnlp-main.403.

Bar-Haim, R., Dinur, E., Feldman, R., Fresko, M., Goldstein, G., 2011. Identifying and following expert investors in stock microblogs, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics. pp. 1310–1319. URL: https://www.aclweb.org/anthology/D11-1121.

Beltagy, I., Lo, K., Cohan, A., 2019. SciBERT: A pretrained language model for scientific text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics. pp. 3615–3620. doi:10.18653/v1/D19-1371.

Bingler, J.A., Kraus, M., Leippold, M., Webersinke, N., 2022. Cheap talk and cherry-picking: What climatebert has to say on corporate climate risk disclosures. Finance Research Letters 47, 102776. URL: https://www.sciencedirect.com/science/article/pii/S1544612322000897, doi:https://doi.org/10.1016/j.frl.2022.102776.

Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics 5, 135–146. doi:10.1162/tacl_a_00051.

Bollen, J., Mao, H., Zeng, X., 2011. Twitter mood predicts the stock market. Journal of computational science 2, 1–8. doi:doi.org/10.1016/j.jocs.2010.12.007.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners 33, 1877–1901. URL: https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Chakrabarty, T., Hidey, C., McKeown, K., 2019. IMHO fine-tuning improves claim detection, 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), Association for Computational Linguistics. pp. 558–563. doi:10.18653/v1/N19-1054.

Chalkidis, I., Fergadiotis, M., Malakasiotis, P., Aletras, N., Androutsopoulos, I., 2020. LEGAL-BERT: The muppets straight out of law school, in: Findings of the Association for Computational Linguistics: EMNLP 2020, Association for Computational Linguistics. pp. 2898–2904. doi:10.18653/v1/2020.findings-emnlp.261.

Clark, K., Luong, M.T., Le, Q.V., Manning, C.D., 2020. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators, in: 8th International Conference on Learning Representations (ICLR). URL: https://openreview.net/forum?id=r1xMH1BtvB.

Cortis, K., Freitas, A., Daudert, T., Huerlimann, M., Zarrouk, M., Handschuh, S., Davis, B., 2017. SemEval-2017 Task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), Association for Computational Linguistics. pp. 519–535. doi:10.18653/v1/S17-2089.

Dai, A.M., Le, Q.V., 2015. Semi-supervised sequence learning, in: Advances in Neural Information Processing Systems (NeurIPS), Curran Associates, Inc.. pp. 3079–3087. URL: https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., Salakhutdinov, R., 2019. Transformer-XL: Attentive language models beyond a fixed-length context, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. pp. 2978–2988. doi:10.18653/v1/P19-1285.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. 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), Association for Computational Linguistics. pp. 4171–4186. doi:10.18653/v1/N19-1423.

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.A., Vincent, P., Bengio, S., 2010. Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research 11, 625–660. URL: http://jmlr.org/papers/v11/erhan10a.html.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets, in: Advances in Neural Information Processing Systems (NeurIPS), Curran Associates, Inc.. pp. 2672–2680. URL: https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

Hendrycks, D., Gimpel, K., 2016. Gaussian error linear units (gelus). doi:10.48550/ARXIV.1606.08415.

Hinton, G., Vinyals, O., Dean, J., 2014. Distilling the knowledge in a neural network, in: Deep Learning and Representation Learning at Twenty-eighth Conference on Neural Information Processing Systems (NeurIPS).

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics. pp. 328–339. doi:10.18653/v1/P18-1031.

Huang, K., Altosaar, J., Ranganath, R., 2020. Clinicalbert: Modeling clinical notes and predicting hospital readmission, in: Conference on Health, Inference, and Learning (CHIL) Workshop Track.

Ito, T., Sakaji, H., Izumi, K., Tsubouchi, K., Yamashita, T., 2018. Ginn: gradient interpretable neural networks for visualizing financial texts. International Journal of Data Science and Analytics doi:10.1007/s41060-018-0160-8.

Jang, E., Gu, S., Poole, B., 2017. Categorical reparameterization with gumbel-softmax, in: 5th International Conference on Learning Representations (ICLR). URL: https://openreview.net/forum?id=rkE3y85ee.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations (ICLR). URL: https://openreview.net/forum?id=8gmWwjFyLj.

Kölbel, J.F., Leippold, M., Rillaerts, J., Wang, Q., 2020. Ask bert: How regulatory disclosure of transition and physical climate risks affects the cds term structure. Swiss Finance Institute Research Paper .

Koppel, M., Shtrimberg, I., 2006. Good News or Bad News? Let the Market Decide. Springer Netherlands. pp. 297–301. doi:10.1007/1-4020-4102-0_22.

Kudo, T., Richardson, J., 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics. pp. 66–71. doi:10.18653/v1/D18-2012.

Kudo, T., Yamamoto, K., Matsumoto, Y., 2004. Applying conditional random fields to japanese morphological analysis, in: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), Association for Computational Linguistics. pp. 230—237.

Kumar, B.S., Ravi, V., 2016. A survey of the applications of text mining in financial domain. Knowledge-Based Systems 114, 128–147. doi:https://doi.org/10.1016/j.knosys.2016.10.003.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2019. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, in: 8th International Conference on Learning Representations (ICLR). URL: https://openreview.net/forum?id=H1eA7AEtvS.

Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents, in: Proceedings of the 31st International Conference on Machine Learning (ICML), PMLR. pp. 1188–1196. URL: https://proceedings.mlr.press/v32/le14.html.

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C.H., Kang, J., 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics 36, 1234–1240. doi:10.1093/bioinformatics/btz682.

Liu, X., He, P., Chen, W., Gao, J., 2019a. Multi-task deep neural networks for natural language understanding, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. pp. 4487–4496. doi:10.18653/v1/P19-1441.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., Allen, P.G., 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. doi:10.48550/arxiv.1907.11692.

Liu, Z., Huang, D., Huang, K., Li, Z., Zhao, J., 2020. FinBERT: A Pre-trained Financial Language Representation Model for Financial Text Mining, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, International Joint Conferences on Artificial Intelligence Organization. pp. 4513–4519. doi:10.24963/ijcai.2020/622.

Logeswaran, L., Chang, M.W., Lee, K., Toutanova, K., Devlin, J., Lee, H., 2019. Zero-shot entity linking by reading entity descriptions, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics. pp. 3449–3460. doi:10.18653/v1/P19-1335.

Maddison, C.J., Mnih, A., Teh, Y.W., 2017. The concrete distribution: A continuous relaxation of discrete random variables, in: 5th International Conference on Learning Representations (ICLR). URL: https://openreview.net/forum?id=S1jE5L5gl.

Malo, P., Sinha, A., Korhonen, P., Wallenius, J., Takala, P., 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. Journal of the Association for Information Science and Technology 65, 782–796. doi:10.1002/asi.23062.

Mariko, D., Abi-Akl, H., Labidurie, E., Durfort, S., De Mazancourt, H., El-Haj, M., 2020. The Financial Document Causality Detection Shared Task (FinCausal 2020). URL: https://aclanthology.org/2020.fnp-1.3.

McCann, B., Bradbury, J., Xiong, C., Socher, R., 2017. Learned in translation: Contextualized word vectors, in: Advances in Neural Information Processing Systems (NeurIPS), Curran Associates, Inc.. pp. 6294–6305. URL: https://proceedings.neurips.cc/paper/2017/file/20c86a628232a67e7bd46f76fba7ce12-Paper.pdf.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Distributed Representations of Words and Phrases and their Compositionality, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 3111–3119. URL: http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.

Milea, V., Sharef, N.M., Almeida, R.J., Kaymak, U., Frasincar, F., 2010. Prediction of the msci euro index based on fuzzy grammar fragments extracted from european central bank statements, in: 2010 International Conference of Soft Computing and Pattern Recognition, pp. 231–236. doi:10.1109/SOCPAR.2010.5686083.

Mittal, S., Chauhan, A., Nagpal, C.K., 2022. Stock Market Prediction by Incorporating News Sentiments Using Bert. Springer International Publishing, Cham. pp. 35–45. URL: https://doi.org/10.1007/978-3-030-96634-8_4, doi:10.1007/978-3-030-96634-8_4.

National Institute of Standards and Technology (U.S.), 2018. Reuters Corpora. doi:10.7910/DVN/IEJ2UX.

Oh, H.S., Thomas, R.J., 2008. Demand-side bidding agents: Modeling and simulation. IEEE Transactions on Power Systems 23, 1050–1056. doi:10.1109/TPWRS.2008.922537.

Peng, B., Chersoni, E., Hsu, Y.Y., Huang, C.R., 2021. Is Domain Adaptation Worth Your Investment? Comparing BERT and FinBERT on Financial Tasks, in: Proceedings of the Third Workshop on Economics and Natural Language Processing, Association for Computational Linguistics. pp. 37–44. doi:10.18653/v1/2021.econlp-1.5.

Pennington, J., Socher, R., Manning, C., 2014. GloVe: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics. pp. 1532–1543. doi:10.3115/v1/D14-1162.

Peters, M.E., Ammar, W., Bhagavatula, C., Power, R., 2017. Semi-supervised sequence tagging with bidirectional language models, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics. pp. 1756–1765. doi:10.18653/v1/P17-1161.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics. pp. 2227–2237. doi:10.18653/v1/N18-1202.

Phang, J., Févry, T., Bowman, S.R., 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. doi:10.48550/ARXIV.1811.01088.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., . Improving language understanding by generative pre-training URL: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Ranco, G., Aleksovski, D., Caldarelli, G., Grčar, M., Mozetič, I., 2015. The effects of twitter sentiment on stock price returns. PLOS ONE 10, 1–21. doi:10.1371/journal.pone.0138441.

Rasmy, L., Xiang, Y., Xie, Z., Tao, C., Zhi, D., 2021. Med-bert: pretrained contextualized embeddings on large-scale structured electronic health records for disease prediction. NPJ digital medicine 4, 1–13. doi:10.1038/s41746-021-00455-y.

Sakaji, H., Sakai, H., Masuyama, S., 2008. Automatic extraction of basis expressions that indicate economic trends, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 977–984. doi:10.1007/978-3-540-68125-0_102.

Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, in: 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing.

Schumaker, R.P., Chen, H., 2009. Textual analysis of stock market prediction using breaking financial news: The azfin text system. ACM Trans. Inf. Syst. 27, 12:1–12:19. doi:10.1145/1462198.1462204.

Schuster, M., Nakajima, K., 2012. Japanese and korean voice search, in: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5149–5152. doi:10.1109/ICASSP.2012.6289079.

Schuster, M., Paliwal, K., 1997. Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing 45, 2673–2681. doi:10.1109/78.650093.

Sennrich, R., Haddow, B., Birch, A., 2016. Neural machine translation of rare words with subword units, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics. pp. 1715–1725. URL: https://aclanthology.org/P16-1162, doi:10.18653/v1/P16-1162.

Smailović, J., Grčar, M., Lavrač, N., Žnidaršič, M., 2013. Predictive sentiment analysis of tweets: A stock market application, in: Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data, Springer Berlin Heidelberg. pp. 77–88.

Sonkiya, P., Bajpai, V., Bansal, A., 2021. Stock price prediction using bert and gan. URL: https://arxiv.org/abs/2107.09055, doi:10.48550/ARXIV.2107.09055.

Sun, C., Qiu, X., Xu, Y., Huang, X., 2019. How to fine-tune bert for text classification?, in: Chinese Computational Linguistics, Springer International Publishing. pp. 194–206. doi:10.1007/978-3-030-32381-3_16.

Tay, Y., Dehghani, M., Rao, J., Fedus, W., Abnar, S., Chung, H.W., Narang, S., Yogatama, D., Vaswani, A., Metzler, D., 2022. Scale efficiently: Insights from pretraining and finetuning transformers, in: The Tenth International Conference on Learning Representations (ICLR). URL: https://openreview.net/forum?id=f2OYVDyfIB.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 5999–6009. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S., 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, in: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics. pp. 353–355. doi:10.18653/v1/W18-5446.

Yamaguchi, A., Chrysostomou, G., Margatina, K., Aletras, N., 2021. Frustratingly simple pretraining alternatives to masked language modeling, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics. pp. 3116–3125. doi:10.18653/v1/2021.emnlp-main.249.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., Le, Q.V., 2019. XLNet: Generalized autoregressive pretraining for language understanding, in: Advances in Neural Information Processing Systems (NeurIPS), pp. 5753–5763. URL: https://proceedings.neurips.cc/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

Zhou, Z., Zhao, F., Wang, J., 2011. Agent-based electricity market simulation with demand response from commercial buildings. IEEE Transactions on Smart Grid 2, 580–588. doi:10.1109/TSG.2011.2168244.