FinBERT–MRC: Financial Named Entity Recognition Using BERT Under the Machine Reading Comprehension Paradigm

Yuzhe Zhang · Hong Zhang

Accepted: 19 March 2023 / Published online: 7 April 2023
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract
Financial named entity recognition (FinNER) is a challenging task in the field of financial text information extraction, which aims to extract a large amount of financial knowledge from unstructured texts. It is widely accepted to use the sequence tagging framework to implement the FinNER tasks. However, such sequence tagging models cannot fully take advantage of the semantic information in the texts. Instead, we formulate the FinNER task as a machine reading comprehension (MRC) problem and propose a new model termed FinBERT–MRC. This formulation introduces significant prior information by utilizing well-designed queries, and extracts the start index and end index of the target entities without decoding modules such as conditional random fields (CRFs). We conduct experiments on a publicly available Chinese financial dataset ChFinAnn and a real-world business dataset AdminPunish. FinBERT–MRC achieves average $F_1$ scores of 92.78% and 96.80% on two datasets, respectively, with average $F_1$ gains $+3.75\%$ and $+0.68\%$ over some sequence tagging models including BiLSTM–CRF, BiGRU–CRF, BiLSTM–CNN–CRF, FinBERT–Tagger, and FinBERT–CRF. The source code is available at https://github.com/zyz0000/FinBERT-MRC.

Keywords Text mining · Named entity recognition · Machine reading comprehension

1 Introduction

In natural language processing (NLP) field, financial named entity recognition (FinNER) aims to automatically retrieve financial entities (e.g., financial institutions, currencies, and companies) in given texts. It is a prerequisite to precisely and effectively recognize financial entities in order to transform them into structured knowledge. Therefore, the FinNER task has received extensive research attention in recent years. The rule-based symbolic methods has been widely applied in the NER tasks for decades [1]. About twenty years ago, several approaches based on feature engineering and specific domain knowledge began to appear. Typical representatives of such models used in the FinNER task include Hidden
Markov Model (HMM) [2], Maximum Entropy (ME) [3], Support Vector Machine (SVM) [4], Conditional Random Fields (CRFs) [5], and Decision Tree (DT) [6]. Feature engineering not only relies heavily on domain-specific knowledge and handcrafted features, but is also model- and entity-specific. In recent years, neural-network-based deep learning methods have been proved to be more effective for the NER tasks [7, 8]. Among them, LSTM–CRF-based methods have gained great popularity. The vector representations of each word (token) in a sentence are first extracted by LSTM, and are then fed into CRF model for sequence tagging. Chiu and Nichols [9] used a hybrid network structure by integrating both character-level and word-level features. Their model decoded each tag independently based on a BiLSTM layer followed with a log-softmax layer. Wang et al. [10] combined CRFs with information entropy to recognize the abbreviation of financial named entity candidates. Miwa and Bansal [11] proposed a BiLSTM encoder and an incrementally-decoded neural network structure to decode tags jointly. These methods generally encode texts based on recurrent neural network (RNN), but differ in decoding phase. Very recently, language models (e.g., Embeddings from Language Models (ELMo) [12], Generative Pre-trained Transformer 3 (GPT3) [13], and Bidirectional Encoder Representations from Transformers (BERT) [14]) obtained state-of-the-art (SOTA) performance in many NLP tasks and have gradually become the mainstream in NLP domain. Compared with the methods depending on feature engineering, deep neural networks are able to automatically extract features and thus can achieve more competitive performance.

The NER task is commonly formalized as a sequence labeling task. A sequence labeling model [9, 14, 15] is trained to assign a tagging class to each unit within a sequence of tokens. However, the models mentioned above, i.e., BiLSTM–CRF and BERT-Tagger, cannot effectively learn the semantic information under the framework of sequence labeling. Inspired by the current trend of formalizing NLP problems under the machine reading comprehension (MRC) framework [16–19], we use FinBERT (namely BERT pretrained on financial corpora) as the backbone to perform the FinNER task under the MRC framework, which we believe can enhance the capability of learning the semantic information in financial texts. Under the MRC framework, each financial entity type is encoded by a language query and identified by answering the query. For example, the task of assigning the \textit{LOC (LOCATION)} label to some token spans in sentence “So far U.S. soldiers have discovered nearly 600 million dollars hidden around [Baghdad]” is formalized as answering the query “Which location is mentioned in the text?”. Compared with the sequence labeling framework, the MRC framework has the advantage of introducing prior knowledge, which can contribute to improving the model performance.

The main contributions of this article are as follows:

- We use FinBERT to accomplish the Chinese FinNER task under the machine reading comprehension paradigm, and realizes the aim of extracting various types of financial entities from both public dataset and real-world business corpus.
- This paper evaluates the effect of query design strategy and training sample size on the performance of the Chinese FinNER task.
- This study obtains a particular real-world business corpus of financial announcement from a Fintech company, and performs text pre-processing, then constructed a Chinese corpus in the financial field which can be used for the Chinese FinNER task.

To the best of our knowledge, we are pioneers to successfully tackle the Chinese FinNER task under the MRC framework and showcase the effectiveness of the proposed model in industrial applications. The remainder of this paper is organized as follows. A brief review of related works is provided in Sect. 2. Our modeling procedure is described in detail in
Sect. 3. Some quantitative and qualitative experimental results are presented in Sect. 4. Some concluding remarks are presented in Sect. 5.

2 Related Works

2.1 Named Entity Recognition (NER)

As one of the basic NLP tasks, NER has been studied extensively ever since it was presented as a subtask at MUC-6 (Message Understanding Conference) in 1995. In order to complete information extraction from unstructured texts, it is necessary to recognize information units such as Date, Person and Location. Identifying references to these entities in text is recognized as one of the important subtasks of information extraction and is referred to as Named-Entity Recognition and Classification (NERC). The earliest implementation of the NER tasks was based on lexical features, which are mostly summarized by linguists through the study on large corpus and named entity libraries. Rule-based methods apply predefined patterns based on certain features, including keywords, punctuation, and statistical information, to perform string matching. However, these methods are time consuming, expensive, and inflexible.

To resolve such problems, researchers proposed to treat NER as a sequence labeling task, i.e., to find the best label sequence for a given input sequence. Typical methods, such as HMM, MEMM, SVM, and CRFs, depend on manually-crafted discrete features [20, 21]. One needs to define handcrafted features that describe the syntactic and semantic essence of the entities to be extracted in the text, before training the model on sufficiently large amount of annotated data.

In recent years, neural networks have been increasingly used in the NER tasks [7]. Neural networks are able to learn contextual features automatically, thereby reducing the reliance on handcrafted features compared with existing machine learning models. Hammerton [22] first applied unidirectional LSTMs to the NER tasks. Collobert et al. [23] introduced convolutional neural network (CNN) to the NER tasks and presented the CNN–CRF model. Lample et al. [7] combined bidirectional LSTMs with CRFs using features based on character-based word embeddings and unsupervised word embeddings. Chiu and Nichols [9] and Ma and Hovy [15] used a character-level CNN to extract features from original texts. Thanks to the advent of pre-trained language models such as ELMo [12] and BERT [14], the performance of NER is significantly improved by the powerful feature extraction capability of pre-trained language models. Kenton and Toutanova [14] solved the NER task using BERT for token classification model. Gao et al. [24] proposed a BERT–BiLSTM–CRF model and validated its feature extraction capability on the downstream NER task.

2.2 Language Models

The acquisition of word embeddings is a crucial step in NLP tasks. Word embeddings mean learning the latent semantic information of words (tokens) from a large amount of unlabeled data, and mapping the words (tokens) into dense low-dimensional vectors, which is also called the distributed representation of words [25]. In the past decade, several word-embedding techniques have been proposed such as Word2Vec [26, 27] and GloVe [28]. Word2Vec utilizes the Skip-Gram model to predict the context words given a target word, or employs the Continuous Bag-Of-Words (CBOW) model to optimize the embeddings so that they can predict the current word based on the contextual information. GloVe uses a weighted least squares
model trained on global word-word co-occurrence counts. The limitation of Word2Vec and GloVe is also obvious, i.e., word embeddings trained by these methods can only model the context-independent representations due to the existence of slicing windows.

The advent of Transformer [29], a sequence-to-sequence model based solely on attention mechanisms, brought significant progress in NLP field. The core component of Transformer is the self-attention mechanism, which aims at modeling the relevance between representation pairs, such that a representation can build a direct relation with another representation. Instead of performing single attention function, Vaswani et al. [29] proposed multi-head attention to capture different contexts with multiple individual attention functions. Subfigures (a) and (b) of Fig. 1 show the mechanism of single head attention and multi-head attention, respectively. The core operations of single-head attention (denoted as Attention) and multi-head attention (denoted as MH) can be formulated as follows:

\[ \text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V, \]
\[ \text{MH}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_H) W^O, \]

where \( d_k \) is the dimension of \( K \), \( \text{head}_h = \text{Attention}(QW^Q_h, KW^K_h, VW^V_h) \), \( 1 \leq h \leq H \), where \( W^Q_h, W^K_h, W^V_h, \) and \( W^O \) are linear transformation matrices with algebraically legal dimensions.

In contemporary NLP, pretrained large language models have demonstrated their success in many NLP tasks. BERT [14], which was trained on Wikipedia\(^1\) and BookCorpus [30], has become a standard building block for training task-specific NLP models. It stacks Transformer encoder layers to pretrain representations by jointly conditioning on both left and right contexts in all layers. Because of the great success of BERT, it has gradually become

---

\(^1\) http://wikipedia.org.
a mainstream method to utilize pretrained BERT as the feature extractor for input texts and finetune it on the target dataset.

Pretrained language models using large-scale domain-specific corpora has gradually been studied in recent years, due to the bottleneck that language models pretrained on universal corpora may not suit the domain-specific downstream tasks well. To this end, many domain-specific BERT models have been trained and released. BioBERT [31] pretrains a biomedical domain-specific language representation model using large-scale biomedical corpora. SciBERT [32] pretrains a scientific domain-specific BERT model using a large-scale scientific publications to improve the performance on downstream scientific NLP tasks. Wang et al. [33] proposed a semi-supervised model named Smiles-BERT to solve different molecular property prediction tasks. Yang et al. [34] are the first to pretrain BERT on financial domain-specific corpora.

2.3 Machine Reading Comprehension (MRC)

The aim of MRC models [16–19, 35] is to extract the answer spans from a context through a query. The task can be formulated as two classification problems: predicting the start and end positions of the answer spans. Over the past one or two years, there has been a trend of solving NLP tasks using MRC models. Levy et al. [16] formulated the task of relation extraction as a question-answering (QA) task, i.e., each relation type \( R(x, y) \) can be parameterized as a query \( q(x) \) whose answer is \( y \). For example, the relation \( \text{ESTABLISHED-IN} \) can be linked to \( \text{When was x established?} \) Given a query \( q(x) \), if a non-null answer \( y \) can be returned from a sentence, it means that the relation label for the current sentence is \( R(x, y) \). McCann et al. [17] transformed ten different NLP tasks into the QA task, and all achieved competitive performance. For the NER task, Li et al. [19] utilized BERT as the backbone to recognize flat and nested entities from texts under the MRC framework. Their work focuses on the general domain NER task. However, to the best of our knowledge, there is currently no specific research for BERT on the Chinese FinNER task under the MRC framework. Our work focuses specifically on financial entities, which is different from Li’s work.

3 Methodology

3.1 Task Definition

Given an input sequence \( X = \{x_1, x_2, \ldots, x_n\} \), where \( n \) is the length of the sequence and \( x_i \) is the \( i \)-th word (token) of the sequence. We need to find all entities in \( X \), and then assign a label \( y \in \mathcal{Y} \) to each entity, where \( \mathcal{Y} \) is the predefined set of all possible label types (e.g., \( = \{\text{LOC}, \text{ORG}, \text{PER}\} \)). We formulate the NER task as an MRC task and thus convert the labeling-style NER dataset into a set of \((\text{Question}, \text{Context}, \text{Answer})\) triplets. Among them, Context is the given input sequence \( X \), and Question is the query sentence designed based on \( X \), and Answer is the target entity span. For each label type \( y \in \mathcal{Y} \), we first construct a question \( q_y = \{q_1, q_2, \ldots, q_m\} \) for each sentence, where \( m \) equals to the length of the question. An annotated entity \( x_{\text{start,end}} = \{x_{\text{start}}, x_{\text{start}+1}, \ldots, x_{\text{end}−1}, x_{\text{end}}\} \) is a substring of \( X \) satisfying \( \text{start} \leq \text{end} \). For example, labels corresponding to a sentence from the ChFinAnn dataset “证券代码: 300142, 证券简称: 沃森生物” (Security code: 300142, Security abbreviation: Walvax) are “O O O O B I I I I I O O O O O B I I I I”. According to the labels, we can obtain the entities and their corresponding spans: \{300142: \{5,10\}, 沃
Table 1 Queries constructed from Wikipedia

| Entity name (in Chinese) | Query (in Chinese) | Query (in English) |
|-------------------------|--------------------|--------------------|
| 价格 (Price)            | 找出价格: 以货币为表现形式, 为商品、服务及资产所订的价值数字 | Find price: A price is the quantity of payment or compensation given by one party to another in return for goods or services |
| 股份 (Share)             | 找出股份: 是投资市场的股本的一个单位, 一般是以股票的形式存在, 但有时亦会以互惠基金、有限合伙、房地产投资信托等形式出现 | Find shares: In financial markets, a share is a unit used as mutual funds, limited partnerships, and real estate investment trusts |
| 日期 (Date)              | 找出日期: 是一个在历法中特定的日子 | Find date: A calendar date is a reference to a particular day represented within a calendar system |
| 机构 (Institution)      | 找出机构: 是指拥有独立架构的社会组织 | Find institution: Institutions can refer to mechanisms which govern the behavior of a set of individuals within a given community, and are identified with a social purpose |

森生物: {17,20} (300142: {5, 10}, Walvax: {17, 20}). By generating a natural language question \( q_y \) based on the label \( y \), we can obtain the triplet \((q_y, X, x_{start, end})\), which is exactly the triplet form \((Question, Context, Answer)\) that we need.

### 3.2 Query Construction

In the work of Li et al. [19], seven various ways were proposed to construct queries for the entities. In our work, we adopt the scheme which uses the explanation from Chinese Wikipedia\(^2\) to construct queries. Table 1 lists some examples of the queries we constructed. In addition, we also evaluate the effect of different query construction methods on the model performance in Sect. 4.

### 3.3 Model Formulation

In this research, we need to recognize entities in the financial domain, so we employed FinBERT as the model backbone. Fig. 2 shows the FinNER task implemented under the MRC framework using FinBERT.

In the first step, the question \( q_y \) and the context \( X \) are concatenated, forming the combined input string

\[
[[CLS], q_1, q_2, \ldots, q_m, [SEP], x_1, x_2, \ldots, x_n],
\]

where [CLS] and [SEP] are special tokens for marking the start and separation of tokens. Then the combined sequence is fed into FinBERT, which can be depicted by the following formulas:

\[
e_i^0 = Wt_i + b,
\]

\(^2\) https://zh.wikipedia.org/wiki.
The pipeline of FinBERT-MRC consisting of three steps. First, the concatenated question and context is encoded by BERT; Second, the probability of each token to be a start or end index is predicted; Third, the entity spans are selected by applying the nearest matching principle.

\[ e_i^l = \text{Transformer}(e_i^{l-1}). \]
\[ E = \left[ e_1^L, e_2^L, \ldots, e_n^L \right] \in \mathbb{R}^{n \times d}, \]

where \( e_i \) is the \( i \)th token embedding, \( W \) and \( b \) are parameters of linear transformation, \( L \) is the number of BERT layers, \( l(1 \leq l \leq L) \) is the number of the \( l \)th BERT layer, and\( \text{Transformer} \) denotes the Transformer encoder layer, including multi-head attention module, fully connected module, and layer normalization. Finally a context representation matrix \( E \in \mathbb{R}^{n \times d} \), where \( d \) is the vector dimension of the last Transformer block of BERT, is acquired.

The second and third steps aim to select entity spans. Generally, there are two choices for selecting entity spans in the MRC framework. The first choice is to train two \( n \)-class classifiers separately to predict the start index and the end index, where \( n \) represents the length of the context. Because the Softmax function is calculated on all tokens within the context, this choice has deficiency that each input sentence can only output one span. The second choice is to construct two binary classifiers, one aims to predict whether the token is a start index, and the other serves to predict whether the token is an end index. This strategy allows for outputting multiple start indexes and multiple end indexes for a given context and a specific query, and therefore has the potential to identify all target entities based on \( q_y \). In our research, we adopt the second strategy. The descriptions of the second and third steps are detailed as follows.
In the second step, given the representation matrix $E$, the model predicts the probability of each token to be a start index. This procedure is formulated as:

$$P_{\text{start}} = \text{softmax}(E W_{\text{start}}) \in \mathbb{R}^{n \times 2},$$

where $W_{\text{start}}$ is a weight matrix to be learned. Softmax is employed for each row of $P_{\text{start}}$. Similarly, the model predicts the probability of each token to be an end index. The formula is

$$P_{\text{end}} = \text{softmax}(E W_{\text{end}}) \in \mathbb{R}^{n \times 2},$$

where $W_{\text{end}}$ is a weight matrix to be learned. Softmax is employed for each row of $P_{\text{end}}$.

In the third step, by applying the argmax function to each row of $P_{\text{start}}$ and $P_{\text{end}}$, we predict indexes that might be start or end positions, i.e.,

$$I_{\text{start}} = \{i | \text{argmax}(P_{\text{start}}^i) = 1, 1 \leq i \leq n\}, \quad I_{\text{end}} = \{i | \text{argmax}(P_{\text{end}}^j) = 1, 1 \leq j \leq n\},$$

where the superscripts $i$ and $j$ denote the $i$th and $j$th rows of a matrix, respectively. Because the datasets we conduct experiments on do not contain overlapped entities, we use the nearest matching principle to match the start and end indexes to obtain the predicted entity span. To be specific, if one start (end) index corresponds to multiple end (start) indexes, then only the nearest end (start) index is selected to form the predicted entity span.

### 3.4 Training and Test

At the training phase, each context $X$ is paired with two label sequences $Y_{\text{start}}, Y_{\text{end}} \in \{0, 1\}^n$, where $Y_{\text{start}}$ is the ground-truth label of each token $x_i$ being the start index of an entity, and $Y_{\text{end}}$ is the ground-truth label of each token $x_i$ being the end index of an entity. We therefore split the loss into two parts:

$$L_{\text{start}} = CE(P_{\text{start}}, Y_{\text{start}}), \quad L_{\text{end}} = CE(P_{\text{end}}, Y_{\text{end}}),$$

where CE$(\cdot, \cdot)$ denotes the cross-entropy loss function, which can be expressed in the form

$$CE(P, Y) = -\frac{1}{n} \sum_{i=1}^{n} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)],$$

where $y_i$ denotes the ground truth label with $y_i = 1$ for positive instance and $y_i = 0$ for negative instance, $p_i$ is the Softmax probability for the $i$th data point. The training objective function is

$$L_{\text{train}} = L_{\text{start}} + L_{\text{end}}.$$

At the test phase, the start and end indexes are first selected based on $I_{\text{start}}$ and $I_{\text{end}}$, respectively. Then, the nearest matching principle is applied to match the start and end indexes, leading to the final extracted answers.
Table 2 Summary statistics of two datasets

|                  | ChFinAnn |                | AdminPunish |                |
|------------------|----------|----------------|-------------|----------------|
|                  | Train    | Validation     | Test        |                |
| No. Docs         | 25,632   | 3,204          | 3,204       | No. Sents      | 2,596          |
| Avg. Sent/Doc    | 19.1     | 22.5           | 21.0        | Avg. SentLength| 234.9          |
| Avg. Tokens/Doc  | 878.7    | 1077.7         | 1008.3      | Avg. EntityNum/Sent | 6.2 |
| Avg. Entities/Doc| 10.6     | 11.3           | 10.9        | Avg. EntityLength | 12.5 |
| Avg. EntityLength| 8.4      | 8.3            | 8.4         |                |                |

No. Docs: Number of documents; Avg. Sent/Doc: Average number of sentences per document; Avg. Tokens/Doc: Average number of tokens per document; Avg. Entities/Doc: Average number of entities per document; Avg. EntityLength: Average entity length; No. Sents: Number of sentences; Avg. SentLength: Average sentence length; Avg. EntityNum/Sent: Average number of entities per sentence; Avg. EntityLength: Average entity length.

4 Experiments

4.1 Experiments on Two Chinese FinNER Datasets

4.1.1 Dataset Description

The first dataset we utilized contains 10-year (2008–2018) ChFinAnn documents. This dataset is originally aimed at developing algorithms for document-level event extraction (DEE) tasks [36]. This dataset contains 32,040 Chinese financial announcement documents with 24 different types of entities, and it has already been split into training, validation, and test sets. This dataset is available at https://github.com/dolphin-zs/Doc2EDAG/blob/master/Data.zip. To the best of our knowledge, we are the first to utilize ChFinAnn dataset in the FinNER research.

The second dataset is a real business dataset concerning administrative punishment from the mobile application program “Enterprise Early Warning” developed by Shanghai Financial China Information & Technology Co., Ltd, which is referred to as AdminPunish in the remainder of this paper. This dataset contains the punishment announcements released by local authorities. The content of the announcements involves punishing a citizen, legal representative or organization that violates the administrative order by reducing rights and interests or increasing obligations. It contains 2,596 annotated sentences with 7 types of entities. The sentences are annotated in BIOE format.

Some summary statistics of the above two datasets are provided in Table 2.

4.1.2 Data Construction

Due to limited computational resources, we cannot use the whole ChFinAnn dataset for model training. Instead, we adopt a resampling scheme to use only a fraction of ChFinAnn in our experiment. In each sampling replication, we randomly extract 1200, 300, and 300 documents from the original training, validation, and test sets, respectively. We delete those sentences that do not contain any entities. In order to minimize the variability due to random sampling, we repeat the aforementioned sampling procedure on the ChFinAnn dataset five times.

3 Crawling from http://www.cninfo.com.cn/new/index.
times to form five resampled datasets. For the AdminPunish dataset, we randomly split it into
training, validation, and test sets according to the proportion of 6:2:2. Again, this splitting
scheme is repeated five times to form five re-splitted datasets.

For ChFinAnn dataset, we properly merge entities according to their actual meanings in
order to simplify the problem to some extent. For example, both of the entities “HighestTrad-
ingPrice” and “LowestTradingPrice” represent price, so we integrate them into a common
entity “Price”. This merging strategy produces 10 types of entities in total for the ChFinAnn
dataset. For the AdminPunish dataset, all seven entities remain unchanged. The names of
the entities, and their proportions in corresponding datasets are listed in Table 3. It can be
known that, compared with AdminPunish dataset, the phenomenon of entity sparsity exists
in ChFinAnn dataset, i.e., “Institution”, “Price”, and “Ratio” are the top three least frequent
entities, which account for very small proportions in the whole ChFinAnn dataset.

### 4.1.3 Baselines

We use the following models as baselines:

- BiLSTM–CRF from [15]. BiLSTM–CRF model can be divided into the BiLSTM layer
  and the CRF layer. The function of the BiLSTM layer is to extract contextual information
  through input words or word vectors, and determine the probability of any type of label
  for making prediction. The CRF layer is used to consider the correlation between tags in
  order to filter the label sequences that are not grammatically reasonable.
- BiGRU–CRF from [37]. This model is similar to BiLSTM–CRF by replacing BiLSTM
  with BiGRU.
- BiLSTM–CNN–CRF from [15]. BiLSTM–CNN–CRF combines Bi-LSTM, CNN and
  CRF.
- BERT–Tagger from [14]. BERT–Tagger treats NER as a tagging task. The model uses
  BERT as the backbone to extract the representation of input texts, and a followed linear
  layer transforms the embedding vectors into a lower dimension, the number of tags in the
  training data. In this way, the NER task is solved in a token classification way.

|               | ChFinAnn       | AdminPunish    |
|---------------|----------------|----------------|
| Entry name    | Proportion (%) | Entry name     | Proportion (%) |
| Price         | 3.43           | Client         | 27.51          |
| Shares        | 27.76          | FileNumber     | 18.33          |
| Institution   | 0.51           | Punishing Authority | 18.65    |
| Company       | 8.49           | Unified Social Credit Codes | 7.82 |
| StockAbbr     | 8.45           | Legal Representative | 11.38    |
| StockCode     | 9.66           | Id Card Number | 3.41           |
| Date          | 19.77          | Address        | 12.90          |
| Ratio         | 5.50           |                |                |
| EquityHolder  | 10.42          |                |                |
| Pledgee       | 6.00           |                |                |
| Model          | seqlen | lr    | bs  | epochs | loss |
|----------------|--------|-------|-----|--------|------|
| BiLSTM–CRF    | 256    | 1e−3  | 64  | 50     | NLL  |
| BiGRU–CRF     | 256    | 1e−3  | 64  | 50     | NLL  |
| BiLSTM–CNN–CRF| 256    | 1e−3  | 64  | 50     | NLL  |
| FinBERT–Tagger| 512    | 5e−5  | 4   | 10     | CE   |
| FinBERT–CRF   | 512    | 5e−5, 5e−3 | 32 | 10     | NLL  |
| FinBERT–MRC   | 512    | 5e−5  | 8   | 10     | CE   |

Table 4 Hyperparameters for model training

- **seqlen**: The maximum sequence length;
- **bs**: The batch size;
- **lr**: The learning rate;
- **NLL**: The negative log likelihood loss;
- **CE**: The cross-entropy loss;
- **epochs**: The minimum number of training epochs when both BERT–MRC and FinBERT–MRC reaches convergence

In the FinBERT–CRF model, the learning rate of CRF layer is set to be $5 \times 10^{-3}$ and those in the other parts of the model are set to be $5 \times 10^{-5}$. This trick aims to accelerate the convergence of CRF layer.

- BERT–CRF from [24]. BERT–CRF improves the feature extraction capability using BERT as the feature extractor, followed by a CRF layer for decoding.

### 4.1.4 Settings

In order to make fair comparison, for BERT–Tagger and BERT–CRF, we utilize the pre-trained FinBERT as the model backbone, denoted as FinBERT–Tagger and FinBERT–CRF. Pretrained FinBERT model checkpoint we used in our experiment can be downloaded from https://github.com/valuesimplex/FinBERT. The hyperparameters used for model training are listed in Table 4.

The experiments are conducted on a Linux server with processor Intel(R) Core(TM) i9-7900X CPU @ 3.30 GHz and GPU GeForce RTX 2080 Ti.

### 4.1.5 Results

The performance of each model is evaluated with the $F_1$-score ($F_1$) defined as $F_1 = 2PR/(P + R)$, where $P$ and $R$ are precision and recall, respectively. We report the mean value of each measure ($P$, $R$, $F_1$) as well as the standard deviation on the test sets in mean ± std format. The test results on two datasets are listed in Table 5 and Table 6. Detailed experimental results can be found in Tables 9 and 10 in Appendices.

Evidently, FinBERT–MRC achieves the best $F_1$ scores on both ChFinAnn and AdminPunish datasets, with average $F_1$ scores of 92.78% and 96.80%, respectively. FinBERT–MRC has $F_1$ score gains of $+2.75\%$ and $+0.27\%$ over FinBERT–CRF on the ChFinAnn and AdminPunish datasets, respectively. This performance boost could be due to the fact that some entities of ChFinAnn are sparse since MRC formulation is more immune to the tag sparsity issue.
Table 5 Prediction accuracy measures (mean ± std) on test sets of the ChFinAnn dataset

| Model               | Precision | Recall    | F1 score    |
|---------------------|-----------|-----------|-------------|
| BiLSTM–CRF          | 87.89 ± 1.80 | 90.19 ± 0.97 | 89.00 ± 0.73 |
| BiGRU–CRF           | 87.63 ± 1.00 | 90.01 ± 0.35 | 88.80 ± 0.53 |
| BiLSTM–CNN–CRF      | 88.74 ± 1.27 | 90.41 ± 0.64 | 89.55 ± 0.43 |
| FinBERT–Tagger      | 85.54 ± 0.61 | 90.23 ± 1.09 | 87.75 ± 0.39 |
| FinBERT–CRF         | 90.80 ± 1.50 | 89.35 ± 2.68 | 90.03 ± 0.89 |
| FinBERT–MRC         | 92.31 ± 0.46 | 93.33 ± 0.79 | 92.78 ± 0.56 |

Table 6 Prediction accuracy measures (mean ± std) on test sets of the AdminPunish dataset

| Model               | Precision | Recall    | F1 score    |
|---------------------|-----------|-----------|-------------|
| BiLSTM–CRF          | 96.36 ± 0.39 | 96.62 ± 0.57 | 96.49 ± 0.43 |
| BiGRU–CRF           | 96.34 ± 0.61 | 96.63 ± 0.58 | 96.48 ± 0.54 |
| BiLSTM–CNN–CRF      | 96.28 ± 0.33 | 96.54 ± 0.51 | 96.41 ± 0.35 |
| FinBERT–Tagger      | 93.19 ± 1.33 | 96.34 ± 1.09 | 94.71 ± 0.88 |
| FinBERT–CRF         | 96.58 ± 0.37 | 96.48 ± 0.77 | 96.53 ± 0.46 |
| FinBERT–MRC         | 96.45 ± 0.53 | 97.16 ± 0.29 | 96.80 ± 0.38 |

4.2 Ablation Studies

4.2.1 The Effect of Query Design Strategy

The construction of query is known to have some impact on the model performance under the MRC framework. Intuitively, the more information a query contains, the better performance a model can achieve. In this subsection, we explore the effect of the following three different query construction methods:

- **Keyword** A query is the keyword describing the tag. For example, the query for tag “机构” (institution) is “机构” (institution). In our experiment, the query is the entity tag itself.
- **Rule-based query template** This generates queries using templates. For example, the query for tag “机构” (institution) is “句子中提到了哪种机构” (Which institution is mentioned in the sentence).
- **Wikipedia** A query is constructed using its Wikipedia definition. The query for tag “机构” (institution) is “一个机构是由众多组成的一个集体, 例如组织或协会” (An institution is an entity comprising multiple people, such as an organization or an association).

It is worth mentioning that there are other query construction methods, such as synonyms, keyword + synonyms, and so on. Nevertheless, the reason why they are not studied in our experiment is that they might not be appropriate for the ChFinAnn and AdminPunish datasets. For example, it is difficult to find synonyms for financial entity “法定代表人” (legal representative). Synonyms may be more suitable for entities of general domain such as organization and location.
Table 7 $F_1$ scores with various types of queries

| Model (query)       | $F_1$         | Model (query)       | $F_1$         |
|---------------------|---------------|---------------------|---------------|
| ChFinAnn            |               | AdminPunish         |               |
| FinBERT–Tagger      | 87.75 ± 0.39  | FinBERT–Tagger      | 94.71 ± 0.88  |
| MRC-Keyword         | 89.44 ± 0.47  | MRC-Keyword         | 95.96 ± 0.40  |
| MRC-Rule-based      | 92.47 ± 0.31  | MRC-Rule-based      | 95.97 ± 0.41  |
| MRC-Wiki            | 92.78 ± 0.56  | MRC-Wiki            | 96.80 ± 0.38  |

In this way, we can comprehensively analyze the effect of different query construction methods on FinBERT–MRC. We follow the same hyperparameter setting in Table 4 and change the queries to train FinBERT–MRC on the ChFinAnn and AdminPunish datasets. Table 7 summaries the $F_1$ scores of three query construction methods: Keyword (MRC-Keyword), Rule-based query template (MRC-rule-based) and Wikipedia (MRC-Wiki), as well as the baseline method FinBERT–Tagger. Evidently, all query construction methods outperform the baseline method FinBERT–Tagger on the two datasets. This conforms with the intuition that incorporating prior knowledge can boost prediction performance. Figure 3 showcases the $F_1$ scores of FinBERT–Tagger and three query construction methods. MRC-Wiki is shown to performs the best in all replications, while MRC-Keyword and MRC-Rule-based outperform FinBERT–Tagger except in one replication for the AdminPunish dataset.

4.2.2 The Effect of Training Sample Size

We also explore the impact of the size of training samples on the performance of FinBERT–MRC. Figure 4 shows the $F_1$ scores on test samples of the FinBERT–MRC models trained with various sizes of training samples. For both FinBERT–Tagger and FinBERT–MRC, the $F_1$ score declines in sync with the training sample size, but FinBERT–MRC appears to be more robust, further demonstrating the superiority of FinBERT–MRC.

4.2.3 Several Representative Examples

In Table 8, we present several representative examples with different prediction results using FinBERT–Tagger and FinBERT–MRC. In the first example, FinBERT–MRC correctly identifies the entity (an institution) while FinBERT–Tagger misses it. This example shows that FinBERT–MRC has advantage over FinBERT–Tagger in terms of learning syntactic information under the context of financial text. In the second and third examples, FinBERT–Tagger does not recognize all entities of the same type in one sentence, while FinBERT–MRC can help alleviate this problem by sufficiently learning the whole sentence. In the fourth example, FinBERT–Tagger only identifies a fragment of the true entity span, while FinBERT–MRC correctly obtains the complete entity span. This phenomenon reflects that FinBERT–MRC can better distinguish the boundary of entities.

The above four examples verify the effectiveness of FinBERT–MRC in syntactic learning, showing that using domain-specific pretrained language model can achieve better generalization. Through this case study, we can infer that FinBERT–MRC outperforms FinBERT–Tagger because of its superiority in syntactic and semantic learning. To be specific,
Fig. 3 $F_1$ scores for different query construction methods evaluated on the ChFinAnn dataset (a) and the AdminPunish dataset (b). Each query construction method is evaluated on five replicates of experiments.
FinBERT-MRC can correct some false negative instances in the presence of multiple entities in one sentence, and can alleviate the boundary recognition problem to some extent.

5 Discussion and Conclusion

In this work, we use BERT to perform FinNER task under the MRC framework. Compared with the conventional methods which solve the NER task under the sequence labeling framework, using FinBERT under the MRC framework can enhance the ability for identifying target entities of the model. We think that the improvement is due to two aspects. First, the application of pretrained language model trained on domain-specific corpora learns better semantic representation for input texts, which is more beneficial for domain-specific downstream tasks. Second, the appropriately designed queries encode important prior knowledge about the entity to extract and the prior knowledge is learned jointly with the original texts. This mechanism instructs the model to find the entity span accurately. The proposed model demonstrates its effectiveness in real-world industrial business scenario. The limitation of this work is that one should conduct some trials on the design of queries according to the experimental results presented in SubSect. 4.2.1. Sometimes it may be not easy to construct appropriate queries, especially for domain-specific corpus. There are two lines of future work. First, we did not apply FinBERT–MRC to nested financial NER datasets because of their unavailability, and we hope to find such datasets in the future so that we can examine the performance of FinBERT–MRC on such datasets. Second, we hope to extend FinBERT–MRC under the low-resource FinNER scenario since FinBERT–MRC shows its advantage under the limited training sample size.
| No | Dataset | Ground truth entities (in Chinese) | Ground truth entities (in English) | Model          | Prediction result (in Chinese) | Prediction result (in English) |
|----|---------|-----------------------------------|-----------------------------------|----------------|--------------------------------|--------------------------------|
| 1  | ChFinAnn| 武汉市汉阳区人民政府(机构)       | Wuhan Hanyang District People’s Government (Institution) | FinBERT–Tagger | 合同纠纷一案 | 合同纠纷一案 |
|    |         |                                   |                                   |                | 武汉市汉阳区人民政府(机构) | 武汉市汉阳区人民政府(机构) |
|    |         |                                   |                                   |                | 合同纠纷一案 | 合同纠纷一案 |
| 2  | ChFinAnn| 2018年9月26日(日期)               | September 26, 2018 (Date)        | FinBERT–Tagger | 2018年9月26日延迟至2019年9月26日 | Delayed from September 26, 2018 to September 26, 2019 |
|    |         |                                   |                                   |                | 2018年9月26日延迟至2019年9月26日 | Delayed from September 26, 2018 to September 26, 2019 |
| 3  | AdminPunish | 国家统计局(处罚机关) | National Bureau of Statistics (Punishing Authority) | FinBERT–Tagger | 合法检查和执法检查国家统计局 | 合法检查和执法检查国家统计局 |
|    |         |                                   |                                   |                | 国家统计局执法检查和 | 国家统计局执法检查和 |
|    |         |                                   |                                   |                | 河北省统计局立案调查查实 | 河北省统计局立案调查查实 |
|    |         |                                   |                                   |                | 河北省统计局立案调查查实 | 河北省统计局立案调查查实 |
|    |         |                                   |                                   |                | By the confirmation of NBS and HBS | By the confirmation of NBS and HBS |
Table 8 (continued)

| No | Dataset     | Ground truth entities (in Chinese) | Ground truth entities (in English) | Model                  | Prediction result (in Chinese) | Prediction result (in English) |
|----|-------------|-----------------------------------|-----------------------------------|------------------------|-------------------------------|-------------------------------|
| 4  | AdminPunish | 市环保局(处罚机关)           | Municipal Environmental Protection Bureau (Punishing Authority) | FinBERT–Tagger         | 信息来源: 市环保局           | Information source: Municipal Environmental Protection Bureau         |
|    |             |                                   |                                   | FinBERT–MRC            | 信息来源: 市环保局           |                                |

For simplicity, National Bureau of Statistics and Hebei provincial Bureau of Statistics are abbreviated as NBS and HBS in Column “Prediction Result (in English)”

Acknowledgements We thank Shanghai Financial China Information & Technology Co., Ltd for providing the AdminPunish dataset for this research.

Author contributions YZ: Conceptualization of this study, Methodology, Software, Writing—Original Draft. HZ: Conceptualization of this study, Methodology, Data curation, Funding acquisition, Writing—Review Editing, Supervision.

Funding This study was funded by the National Natural Science Foundation of China (No. 72091212) and the Anhui Center for Applied Mathematics.

Declarations

Competing interests The authors declare no competing interests.

A Appendix

A.1. Detailed experiment results on the ChFinAnn and AdminPunish datasets

See Tables 9 and 10.
### Table 9 Prediction accuracy measures of six models on the ChFinAnn dataset based on five replications of experiments

| Metric | Model                  | Rep1   | Rep2   | Rep3   | Rep4   | Rep5   |
|--------|------------------------|--------|--------|--------|--------|--------|
| Precision | BiLSTM–CRF            | 89.75  | 89.81  | 86.25  | 86.17  | 87.49  |
|         | BiGRU–CRF             | 88.93  | 88.01  | 86.78  | 86.84  | 87.95  |
|         | BiLSTM–CNN–CRF        | 90.02  | 90.13  | 88.38  | 87.62  | 87.53  |
|         | FinBERT–Tagger        | 86.00  | 85.88  | 86.01  | 85.14  | 84.67  |
|         | FinBERT–CRF           | 91.53  | 90.75  | 91.95  | 88.32  | 91.43  |
|         | FinBERT–MRC           | 92.83  | 91.75  | 91.91  | 92.61  | 92.45  |
| Recall  | BiLSTM–CRF            | 90.51  | 88.69  | 90.15  | 91.38  | 90.22  |
|         | BiGRU–CRF             | 90.23  | 89.44  | 89.92  | 90.27  | 90.18  |
|         | BiLSTM–CNN–CRF        | 90.15  | 89.78  | 89.92  | 91.16  | 91.02  |
|         | FinBERT–Tagger        | 89.02  | 90.98  | 89.77  | 89.66  | 91.71  |
|         | FinBERT–CRF           | 90.73  | 88.73  | 85.83  | 92.99  | 88.49  |
|         | FinBERT–MRC           | 94.09  | 93.69  | 92.02  | 93.55  | 93.28  |
| $F_1$   | BiLSTM–CRF            | 90.12  | 89.24  | 88.15  | 88.70  | 88.83  |
|         | BiGRU–CRF             | 89.58  | 88.72  | 88.32  | 88.33  | 89.05  |
|         | BiLSTM–CNN–CRF        | 90.08  | 89.95  | 89.14  | 89.35  | 89.24  |
|         | FinBERT–Tagger        | 87.43  | 88.27  | 87.79  | 87.30  | 87.96  |
|         | FinBERT–CRF           | 91.13  | 89.73  | 88.78  | 90.60  | 89.93  |
|         | FinBERT–MRC           | 93.44  | 92.65  | 91.92  | 93.06  | 92.81  |

### Table 10 Prediction accuracy measures of six models on the AdminPunish dataset based on five replications of experiments

| Metric | Model                  | Rep1   | Rep2   | Rep3   | Rep4   | Rep5   |
|--------|------------------------|--------|--------|--------|--------|--------|
| Precision | BiLSTM–CRF            | 96.24  | 96.74  | 96.23  | 96.77  | 95.84  |
|         | BiGRU–CRF             | 96.38  | 96.82  | 96.05  | 96.98  | 95.46  |
|         | BiLSTM–CNN–CRF        | 96.40  | 96.77  | 96.24  | 96.12  | 95.88  |
|         | FinBERT–Tagger        | 94.99  | 92.84  | 91.78  | 92.24  | 94.08  |
|         | FinBERT–CRF           | 95.97  | 96.98  | 96.58  | 96.73  | 96.65  |
|         | FinBERT–MRC           | 97.09  | 96.33  | 96.37  | 95.69  | 96.76  |
| Recall  | BiLSTM–CRF            | 96.10  | 97.10  | 95.95  | 97.19  | 96.74  |
|         | BiGRU–CRF             | 96.22  | 97.04  | 95.97  | 97.38  | 96.52  |
|         | BiLSTM–CNN–CRF        | 96.34  | 97.02  | 96.02  | 97.14  | 96.18  |
|         | FinBERT–Tagger        | 96.97  | 95.83  | 96.80  | 95.58  | 96.53  |
|         | FinBERT–CRF           | 96.57  | 97.29  | 96.70  | 96.66  | 95.20  |
|         | FinBERT–MRC           | 97.29  | 97.07  | 96.86  | 97.00  | 97.60  |
Table 10 (continued)

| Metric | Model               | Rep1 | Rep2 | Rep3 | Rep4 | Rep5 |
|--------|---------------------|------|------|------|------|------|
| F₁     | BiLSTM–CRF          | 96.17| 96.92| 96.09| 96.98| 96.29|
|        | BiGRU–CRF           | 96.30| 96.93| 96.01| 97.18| 95.99|
|        | BiLSTM–CNN–CRF      | 96.37| 96.89| 96.13| 96.63| 96.03|
|        | FinBERT–Tagger      | 95.96| 94.28| 94.19| 93.85| 95.28|
|        | FinBERT–CRF         | 96.27| 97.13| 96.64| 96.69| 95.92|
|        | FinBERT–MRC         | 97.19| 96.70| 96.96| 96.32| 97.17|

References

1. Spasić I, Livsey J, Keane JA, Nenadić G (2014) Text mining of cancer-related information: review of current status and future directions. Int J Med Inform 83(9):605–623. https://doi.org/10.1016/j.ijmedinf.2014.06.009
2. Eddy SR (1996) Hidden markov models. Curr Opin Struct Biol 6(3):361–365. https://doi.org/10.1016/S0959-440X(96)80056-X
3. Kapur JN (1989) Maximum-entropy models in science and engineering. John Wiley & Sons
4. Hearst MA, Dumais ST, Osuna E, Platt J, Scholkopf B (1998) Support vector machines. IEEE Intell Syst Appl 13(4):18–28. https://doi.org/10.1109/5254.708428
5. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th international conference on machine learning. https://doi.org/10.1109/ICML.2012.6466940
6. Quinlan JR (1986) Induction of decision trees. Mach Learn 1(1):81–106. https://doi.org/10.1007/BF00116251
7. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 260–270. https://doi.org/10.18653/v1/N16-1030
8. Jaganmatha AN, Yu H (2016) Structured prediction models for RNN based sequence labeling in clinical text. In: Proceedings of 2016 the conference on empirical methods in natural language processing, pp 856–865. https://doi.org/10.18653/v1/d16-1082
9. Chiu JP, Nichols E (2016) Named entity recognition with bidirectional LSTM–CNNs. Trans Assoc Comput Linguist 4:357–370. https://doi.org/10.1162/tacl_a_00104
10. Wang S, Xu R, Liu B, Gui L, Zhou Y (2014) Financial named entity recognition based on conditional random fields and information entropy. In: 2014 international conference on machine learning and cybernetics, IEEE, pp 838–843. https://doi.org/10.1109/ICMLC.2014.7009718
11. Miwa M, Bansal M (2016) End-to-end relation extraction using LSTMs on sequences and tree structures. In: Proceedings of the 54th annual meeting of the association for computational linguistics, pp 1105–1116
12. Peters ME, 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, pp 2227–2237. https://doi.org/10.18653/v1/N18-1202
13. Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A et al (2020) Language models are few-shot learners. Adv Neural Inf Process Syst 33:1877–1901. https://doi.org/10.48550/arXiv.2005.14165
14. Kenton JDMWCL, 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, pp 4171–4186
15. Ma X, Hovy E (2016) End-to-end sequence labeling via Bi-directional LSTM–CNNs–CRF. In: Proceedings of the 54th annual meeting of the association for computational linguistics, pp 1064–1074. https://doi.org/10.13140/RG.2.1.2182.5685
16. Levy O, Seo M, Choi E, Zettlemoyer L (2017) Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st conference on computational natural language learning (CoNLL 2017), pp 333–342. https://doi.org/10.18653/v1/K17-1034
17. McCann B, Keskar NS, Xiong C, Socher R (2018) The natural language decathlon: multitask learning as question answering, arXiv preprint arXiv:1806.08730. https://doi.org/10.48550/arXiv.1806.08730
18. Li X, Yin F, Sun Z, Li X, Yuan A, Chai D, Zhou M, Li J (2019) Entity-relation extraction as multi-turn question answering. In: Proceedings of the 57th annual meeting of the association for computational linguistics, pp 1340–1350. https://doi.org/10.18653/v1/P19-1129
19. Li X, Feng J, Meng Y, Han Q, Wu F, Li J (2020) A unified MRC framework for named entity recognition. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 5849–5859. https://doi.org/10.18653/v1/2020.acl-main.519
20. Yao L, Sun C, Li S, Wang X, Wang X (2009) CRF-based active learning for Chinese named entity recognition. In: 2009 IEEE international conference on systems, man and cybernetics, pp 1557–1561. https://doi.org/10.1109/ICSMC.2009.5346315
21. Han X, Ruoran R (2011) The method of medical named entity recognition based on semantic model and improved SVM–KNN algorithm. In: 2011 seventh international conference on semantics, knowledge and grids, pp 21–27. https://doi.org/10.1109/SKG.2011.24
22. Hammerton J (2003) Named entity recognition with long short-term memory. Proc Seventh Conf Nat Lang Learn HLT-NAACL 2003:172–175
23. Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P (2011) Natural language processing (almost) from scratch. J Mach Learn Res 12:2493–2537. https://doi.org/10.1016/j.chemolab.2011.03.009
24. W. Gao, X. Zheng, S. Zhao (2021) Named entity recognition method of Chinese EMR based on BERT—BiLSTM–CRF. In: Journal of physics: conference series, p 012083. https://doi.org/10.1088/1742-6596/1848/012083
25. Liu Z, Lin Y, Sun M (2020) Representation learning for natural language processing. Springer Nat. https://doi.org/10.1007/978-981-15-5573-2
26. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst. https://doi.org/10.5555/2999792.2999959
27. Mikolov T, Corrado G, Chen K, Dean J (2013) Efficient estimation of word representations in vector space. In: Proceedings of the international conference on learning representations (ICLR 2013). https://doi.org/10.48550/arXiv.1301.3781
28. Pennington J, Socher R, Manning CD (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543. https://doi.org/10.3115/v1/D14-1162
29. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008. https://doi.org/10.48550/arXiv.1706.03762
30. Zhu Y, Kiros R, Zemel R, Salakhutdinov R, Urtasun R, Torralba A, Fidler S (2015) Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In: Proceedings of the IEEE international conference on computer vision, pp 19–27. https://doi.org/10.1109/ICCV.2015.11
31. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, Kang J (2020) BioBERT: a pre-trained biomedical representation model for biomedical text mining. Bioinformatics 36(4):1234–1240. https://doi.org/10.1093/bioinformatics/btz682
32. 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), pp 3615–3620. https://doi.org/10.18653/v1/D19-1371
33. Wang S, Guo Y, Wang Y, Sun H, Huang J (2019) SMILES-BERT: large scale unsupervised pre-training for molecular property prediction. In: Proceedings of the 10th ACM international conference on bioinformatics, computational biology and health informatics, pp 429–436. https://doi.org/10.1145/3307339.3342186
34. Yang Y, UY MCS, Huang A (2020) FinBERT: a pretrained language model for financial communications, arXiv e-prints arXiv:2006.08097. https://doi.org/10.48550/arXiv.2006.08097
35. Shen Y, Huang PS, Gao J, Chen W (2017) ReasoNet: learning to stop reading in machine comprehension. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1047–1055. https://doi.org/10.1145/3097983.3098177
36. Zheng S, Cao W, Xu W, Bian J (2019) Doc2EDAG: an end-to-end document-level framework for Chinese financial event extraction. 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), pp 337–346. https://doi.org/10.18653/v1/D19-1032

37. Ma S, Cheng L, Huang S, Cui B (2021) Event extraction of Chinese electronic medical records based on BiGRU-CRF. In: 2021 4th international conference on artificial intelligence and pattern recognition, pp 592–598. https://doi.org/10.1145/3488933.3488981

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.