Engineering Applications of Artificial Intelligence 97 (2021) 104100

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Transformers-based information extraction with limited data for domain-specific business documents[☆]

Minh-Tien Nguyen[a,b,*], Dung Tien Le[a], Linh Le[c]

[a] CINNAMON LAB, 10th floor, Geleximco building, 36 Hoang Cau, Dong Da district, Hanoi, Viet Nam
[b] Hung Yen University of Technology and Education, Hung Yen, Viet Nam
[c] The University of Queensland, Australia

ARTICLE INFO

Keywords:
Information extraction
Transfer learning
Transformers

ABSTRACT

Information extraction plays an important role for data transformation in business cases. However, building extraction systems in actual cases face two challenges: (i) the availability of labeled data is usually limited and (ii) highly detailed classification is required. This paper introduces a model for addressing the two challenges. Different from prior studies that usually require a large number of training samples, our extraction model is trained with a small number of data for extracting a large number of information types. To do that, the model takes into account the contextual aspect of pre-trained language models trained on a huge amount of data on general domains for word representation. To adapt to our downstream task, the model employs transfer learning by stacking Convolutional Neural Networks to learn hidden representation for classification. To confirm the efficiency of our method, we apply the model to two actual cases of document processing for bidding and sale documents of two Japanese companies. Experimental results on real testing sets show that, with a small number of training data, our model achieves high accuracy accepted by our clients.

1. Introduction

A large amount of unstructured data makes a big challenge to people in capturing important information. It leverages the growth of AI technologies, in which many applications have already been deployed to real business cases (Finkel and Manning, 2009; Manyika et al., 2017; Ju et al., 2018; Nguyen et al., 2019; Zhang et al., 2020). From the business side, the conversion of unstructured data into structured text makes a huge impact because it is stated as an entrance to the digital transformation (Inmon and Nesavich, 2007; Herbert, 2017; Lin et al., 2019). One technique of the conversion is information extraction, which extracts important information of a given input document. Over decades, the extraction has been studied by many methods (Angeli et al., 2015; Shimaoka et al., 2016; Deng and Liu, 2018; Andrew, 2018; Zhang et al., 2020). In business services, the extraction of specific information matters such as organization names, personal names, addresses, or dates plays an important role to facilitate document processing systems. The outputs of extraction can be used in many natural language processing (NLP) applications such as question answering, information retrieval (Lee et al., 2006; Shimaoka et al., 2016), or the automatic generation of ontology (Fleischman and Hovy, 2002). Therefore, information extraction (IE) is one of the key NLP technologies which help AI software contributing to commercial applications (Tur and De Mori, 2011).

Named entity recognition (NER) is an important task of information extraction. The NER task focuses on extracting several pre-defined entities, such as organization names, personal names, addresses, etc. Due to the huge impact of NER in NLP applications, it has been received attention from the research community (Watanabe et al., 2007; Finkel and Manning, 2009; Lample et al., 2016; Ju et al., 2018). The majority of the research has successfully done the simple tasks of NER, yet those have focused on relatively-easy tasks, e.g. extracting names of persons or organizations; hence it is not always straightforward to apply the proposed algorithms to real cases due to two gaps. The first gap is about the amount of training data. In real business scenarios, document processing typically focuses on narrow and specific topics rather than general and wide domains, e.g. news. In such scenarios, there is only a small amount of annotated data. Let us take our case as an example. We only received 100 bidding documents for both training and testing. It creates an obstacle to the machine learning-based approach, which usually requires thousand annotated training examples. For instance, CoNLL 2003 provided 20,000 annotated words for NER (Sang and De Meulder, 2003; Devlin et al., 2019). For actual

[☆] This manuscript is an improved and extended version of the paper: Transfer Learning for Information Extraction with Limited Data, presented at 16th International Conference of the Pacific Association for Computational Linguistics (PACLING) 2019, Hanoi, Vietnam.
[*] Corresponding author at: Hung Yen University of Technology and Education, Hung Yen, Viet Nam.
E-mail addresses: tiennm@utehy.edu.vn (M.-T. Nguyen), nathan@cinnamon.is (D.T. Le), linh.le@uq.edu.au (L. Le).

https://doi.org/10.1016/j.engappai.2020.104100
Received 26 May 2020; Received in revised form 3 October 2020; Accepted 9 November 2020
Available online 24 November 2020

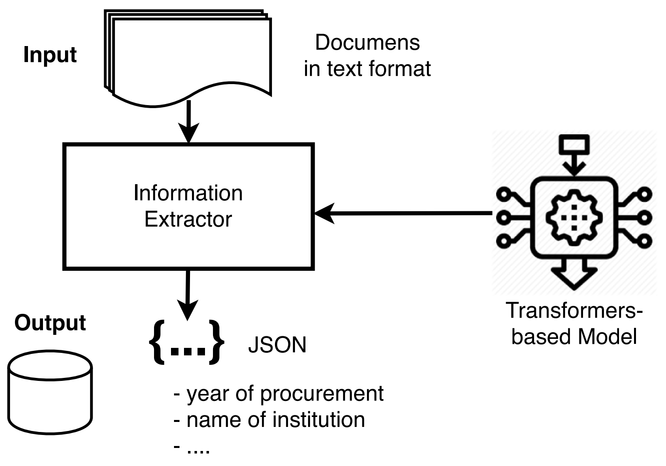


Fig. 1. The overview of the system. Given an input document, the information extractor employs pre-trained language models trained by transformers for the extraction. Outputs are stored in a database for document processing.

cases, in a narrow domain, the annotation of a large number of training data is a non-trivial, time-consuming, and labor-expensive task. For the second gap, the identification of entity types cannot be merely the categories (Fleischman and Hovy, 2002; Lee et al., 2006; Shimaoka et al., 2016), e.g. two types of organizations, such as payee and payer in invoices in bidding documents. We believe this detailed level of extracted information is the key to the practical use of NER.

In this research, we aim to overcome the two gaps of information extraction in real scenarios by introducing a general extraction system in Fig. 1. The system receives input documents and then the information extractor distills nugget information to store in a database for other tasks of document processing in our system. More precisely, for the first gap of the lack of training data, the idea of transfer learning is utilized to the fine-grained NER task. The intuition is that the power of transformers trained on a huge amount of data is exploited to fine-tune the transformer-based model into our downstream task. This is because we receive small training examples, i.e. 100 business documents, which challenge common machine learning algorithms. For the second gap, we focus on identifying a particular type of named entities from business documents. Compared to the traditional NER task, which usually extracts four or seven entity types, we deal with a larger number of types, e.g. 24 types of entities. This makes our extraction to be a non-trivial task. This paper makes three main contributions:

- We propose a practical model of fine-grained NER which employs transformers, e.g. BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2020) as an element of transfer learning (Pan and Yang, 2009; Shin et al., 2016; Weiss et al., 2016). By employing and fine-tuning transformers into the downstream task with a small amount of data, our model achieves significant improvements compared to strong methods in extracting fine-grained entities. To the best of our knowledge, we are the first study for the extraction of fine-grained NER for Japanese business documents.
- We present two actual cases of applications that have been demanded by two Japanese companies and confirm that the accuracy of the proposed model is acceptably high enough for practical use. Statistical analyses show that our model with small training data achieves improvements in terms of F-score compared to strong baselines.
- We release an online system[1] which can process real bidding and sale Japanese business documents. Our introduction video is also available.[2] We believe that our system will facilitate companies as well as the research community in converting unstructured documents to structured data.

The rest of the paper is organized as follows. We give a review of the related works in Section 2. We next introduce the task and datasets in Section 3. We detail our framework in Section 4 and show the statistical analysis in Section 5. After extracting summaries, we show experimental results along with discussion and analyses in Section 6. The paper makes conclusions in Section 7.

2. Related work

Information extraction (IE) is an important task of natural language processing (NLP). The task is to extract important information of an input document. The IE task can be addressed by using two approaches: unsupervised learning and supervised learning. For the first approach, conventional methods usually utilize dictionaries for extracting information. The dictionary-based method usually uses a pre-defined dictionary of entities to match tokens in documents. For example, Watanabe et al. (2007) introduced a method for categorizing named entities in Wikipedia. To do that, the authors assumed an anchor text can be considered as an entity. For categorizing, the authors introduced a graph structure in which an anchor text is a node and used Conditional Random Fields (CRF) (Lafferty et al., 2001) for entity recognition. In this work, the authors utilized anchor texts in Wikipedia as a large dictionary. Del Corro and Gemulla (2013) introduced a ClauseIE system for the problem of clause extraction for open domains. ClauseIE utilizes linguistic knowledge of English grammar to detect clauses and then identifies the type of each clause. By using dependency parsing and a small set of domain-independent lexica, ClauseIE achieves high accuracy on various real-world datasets. The unsupervised method, e.g. dictionary-based (Watanabe et al., 2007) can achieve high accuracy, but it is time-consuming and labor-expensive to prepare the dictionary. In contrast, the machine learning method exploits features to train a classifier that can distinguish extracted information. Angeli et al. (2015) exploited linguistic structure for open-domain information extraction. The authors trained a classifier with a set of features of canonically structured sentences. The classifier learns to split a sentence into shorter utterances and then appeals to natural logic. The proposed model showed promising results on the end-to-end TAC-KBP 2013 Slot Filling task. Recently, the success of deep learning attracts researchers to apply this technique to information extraction. A recent study employs Long-Short Term Memory (LSTM) with a Conditional Random Field (CRF) to classify the contextual expressions (Lample et al., 2016). More precisely, the authors used LSTM to learn the hidden representation of data and then stacked CRF for classification. This method showed promising results. In practice, several research projects focus on the nested named entities and have great progress so far (Finkel and Manning, 2009; Ju et al., 2018).

For NER, high-level concepts such as people, places, organizations usually need to extract. For example, Sang and De Meulder (2003) introduced the shared task of NER, which extracts four types of entities: location (LOC), mixed entities (MISC), organization (ORG), and person (PER). However, for practical applications, categories must be at a more detailed level (Del Corro et al., 2015; Andrew, 2018; Nguyen et al., 2019). Here, fine-grained entity type classification was proposed, especially in the field of question answering, information retrieval (Lee et al., 2006; Shimaoka et al., 2016), or the automatic generation of ontology (Fleischman and Hovy, 2002). For example, the work of Del Corro et al. (2015) can be considered as the most fine-grained entity type classification system to-date, in which the proposed method classifies more than 16,000 types on the entire WordNet hierarchy. However, the main challenge of fine-grained NER is the amount of

[1] https://aurora-demo.cinnamon.is. The guideline is shown in Section 6.5. The username is `cinnamon` and password is `cinnamon`.

[2] http://y2u.be/xHQpYE41Tqw.

training data required to train the classifier. To tackle this problem, transfer learning (Weiss et al., 2016; Nguyen et al., 2019) is an appropriate solution. It leverages pre-trained models trained by a large amount of out-domain data to build a new model with a small number of training data in a new domain. Transfer learning is efficient because we daily have faced with limited training data. Let us take our scenario as an example, we need to extract values for 24 tags in long Japanese documents while the number of the training documents is only 78. Thus, transfer learning is one of the most efficient techniques in such scenarios (Pan and Yang, 2009; Shin et al., 2016; Weiss et al., 2016). Two recent successful transformers models: BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2020) are the recent open-source of NER. BERT (Bidirectional Encoder Representation from Transformers) (Devlin et al., 2019) was trained on a huge amount of data by using the transformer architecture for language understanding. After training, the pre-trained BERT model can be used to downstream tasks. ALBERT (Lan et al., 2020) is a lighter version of BERT, in which the model compresses the number of parameters. Both models have achieved state-of-the-art results on many NLP tasks, including the very competitive Stanford Question Answering Dataset (SQuAD). In this work, we develop a model based on BERT and ALBERT for our business task, which extracts information in long Japanese documents.

The method of Zhang et al. (2020) is perhaps the most closet to our short version (Nguyen et al., 2019) of this paper. The authors employed BERT for extracting important content from regulatory filings and property lease agreement documents. To do that, the extraction was formulated as a sequence labeling task, in which BERT was used to directly infer extracted information. Our study shares the idea of using BERT for information extraction with limited data; however, our model makes three significant differences. Firstly, instead of directly using BERT, we used BERT in a transfer learning fashion. We fine-tuned BERT into downstream tasks by adding additional layers for extracting important information. The additional layer allows our model to capture data representation of specific domains. As a result, our model achieves better results compared to the original BERT and ALBERT models (Table 5). Secondly, we formulated the extraction as a question and answering task, which is different from sequence labeling of Zhang et al. (2020). Finally, we investigated transformers, e.g. BERT and ALBERT, which give comprehensive investigation compared to Zhang et al. (2020).

3. The task and data

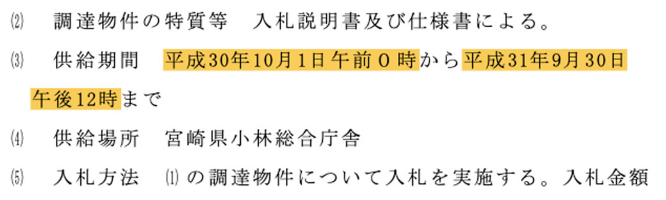
This section first introduces the problem and then describes data preparation for our IE in domain-specific business documents.

3.1. The task

Business documents. This research focuses on an actual case of document processing of two Japanese companies, using bidding and sale documents. The documents are long and complicated, making a challenge for IE models. Also note that most of IE approaches, e.g. NER have limitation in applying to specific domains. Let take a bidding document as an example. A bidding is a long document (20–50 pages) which has 3 sections: (i) specifications, (ii) invitation to bid, and (iii) instructions to bid. Fig. 2 shows a part of a bidding document, in which IE models need to extract supply period highlighted in yellow.

The task. Our IE model needs to extract specific information from business documents. Given a set of tags defined by our clients and an input document, the model extracts information corresponding to tags. We can also consider the task as key–value extraction. In this paper, we use tag-value and key–value simultaneously with the same meaning.

Table 1 shows target information to be extracted of bidding documents. As observed, there are 24 types of extracted information. It is different from common NER, which includes a small number of entity types (4–7 types). Also, many tags may have the same value or data


⑵　調達物件の特質等　入札説明書及び仕様書による。

⑶　供給期間　平成30年10月1日午前０時から平成31年9月30日午後12時まで

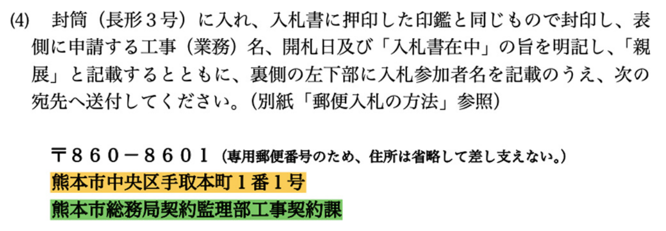
⑷　供給場所　宮崎県小林総合庁舎

⑸　入札方法　⑴の調達物件について入札を実施する。入札金額

Fig. 2. A part of bidding document. Clause 3 (two first lines) means "*supply period from 10/01/2018 to 09/30/2019* (MM/dd/YYYY)".

Table 1
Information need to be extracted from bidding contract documents. Japanese tags were translated to English for easy use. Content type of name can be the name of an entity such as a company, department, or person.

No.	Tags need to extract	Type
1	Year of procurement	Year
2	Prefecture	Text
3	Title of bidding	Text
4	Name of institution	Text
5	Address for demand	Address
6	Start date of procurement	Date
7	End date of procurement	Date
8	Contracted electric energy	Number
9	Amount of electric energy	Number
10	Classification of reserved electric energy	Number
11	Contracted reserved electric energy	Number
12	Public announcement data	Date
13	Deadline for delivery the specification	Date
14	Deadline of questionnaire	Date
15	Deadline for applying qualification	Date
16	Deadline for bidding	Date
17	Opening application date	Date
18	PIC of inquiry of questions	Name
19	TEL/FAX of inquiry of question	Tel/fax
20	Address for submitting application of qualification	Address
21	Department & PIC for submitting application of qualification	Name
22	Address of submitting of bid	Address
23	Department & PIC for submitting of bid	Name
24	Place of opening bid	Address


(4)　封筒（長形３号）に入れ、入札書に押印した印鑑と同じもので封印し、表側に申請する工事（業務）名、開札日及び「入札書在中」の旨を明記し、「親展」と記載するとともに、裏側の左下部に入札参加者名を記載のうえ、次の宛先へ送付してください。(別紙「郵便入札の方法」参照)

〒８６０－８６０１（専用郵便番号のため、住所は省略して差し支えない。）
熊本市中央区手取本町１番１号
熊本市総務局契約監理部工事契約課

Fig. 3. Example of a paragraph in a document. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

types, e.g. deadlines for the questionnaire (tag 14) and bidding (tag 16). Hence, locating the value of a specific tag is challenging and needs to understand the context in the document.

A part of a bidding document with three clauses 3, 4, 5 of the specification section is showed in Fig. 2. The yellow highlighted texts are the values for tag 6 (start date of procurement), and tag 7 (end date of procurement). Another difficulty is that a tag is not represented by specific terms. To retrieve the values correctly, the learning model needs to understand the document structures and meaning of the text. For example, given a paragraph in Fig. 3, the yellow line is the value of tags 20, 22, and the green line is the value of tags 18, 21, 23, 24, the model needs to decide which is the tag of the yellow and green lines.

The tags of extracted information of sale documents are showed in Table 2. We can observe that the number of tags is smaller than that of bidding documents. There are three tags with mixed data and others are date–time.

Table 2
Extracted information of sale documents.

No.	Tags need to extract	Type
1	Model code	Mixed
2	Model name	Mixed
3	Start of sales	Date
4	End of sales (planed)	Date
5	End of sales (fixed)	Date
6	End of sales (special)	Date
7	End of support	Date
8	Revision	Mixed

Table 3
Data observation on two datasets.

Statistics	Bidding documents		Sale documents	
	Mean	Std.	Mean	Std.
#training samples	78	–	300	–
#testing samples	22	–	165	–
#characters per sample	22,537	17,191	2083	2089
#sentences per sample	616	368	56	46
# of tags	24	–	8	–

3.2. Data preparation

Data. As mentioned, we focus on extracting information from long documents in specific domains. The extraction is different from common IE or NER, in which our model needs to extract a lot of information. Therefore, common datasets of IE or NER (e.g. CoNLL) may not applicable to our real scenarios. To evaluate IE models, we prepared two datasets taken from our clients.

The first dataset contains 100 bidding documents of a Japanese company, in which 78 documents were used for training and 22 documents for testing. From Tables 1 and 2, we can observe that the extraction extracts many similar types of short values in a very long bidding document, which has an average of 616 sentences. The second dataset includes sale documnets, in which the number of training examples is three times bigger (300) than that (78) of the bidding dataset. The number of testing examples is much larger. The length of sale documents is much shorter than that of bidding documents. It explains Tables 1 and 2, in which the number of tags of sale documents is three times smaller than that of bidding documents. By doing this, we want to confirm the efficiency of our model on two identical datasets: long documents and short documents. We can also observe that a small number of training examples and long documents challenge IE models in order to achieve high accuracy.

Data annotation. Raw data was sent from our clients; therefore, our QAs (quality assurance) were asked to annotate the data. Our QAs are people who have Japanese language skills, in which they have at least the N3 certificate (Japanese-Language Proficiency Test — JLPT, with N1 is the highest level).

The annotation was internally conducted with two annotators in two steps. In the first step, each annotator was assigned a set of documents. They read tags and segments in each document. For each tag, the starting position and ending position were assigned to a corresponding segment in the document. The starting and ending positions are an indicator showing that which segments belong to which tags. The second step is cross-validation, in which annotated data of an annotator was cross-checked by the other annotator. The agreement computed by Cohen Kappa[3] of two annotators is 0.682, showing that the annotators have a high agreement in annotating data.

[3] http://graphpad.com/quickcalcs/kappa1.cfm.

4. Information extraction with transfer learning

As mentioned, one of the most challenges of our problem comes from the limitation of training data. In actual business cases, we usually receive a small number of training samples. For example, the bidding document dataset only includes 100 examples. Therefore, building a high-quality IE model is a non-trivial task. After investigation, we come up with the idea of employing transfer learning for dealing with the limited training data (Pan and Yang, 2009; Weiss et al., 2016). The transfer learning process includes two steps: training on general and well-known domains and fine-tuning the trained model on new domains. For the first step, based on the nature of having a representation model that can encode the contextual aspect of words, we take advantage of pre-trained language models, thanks to the recent success of transformers such as BERT (Devlin et al., 2019) or ALBERT (Lan et al., 2020) in NLP tasks. BERT and ALBERT allow our model to well representing the meaning of tokens that is crucial for the learning process. For the second step, the transformers need to be adapted to new domain by using transfer learning. To do that, a solution is to stack additional layers on the top of transformers. The final vector learnt from the model is used for the prediction of extraction information.

The overall architecture of our model is showed in Fig. 4. The model has four main components: (i) the input vector representations of input tokens, (ii) transformers for learning hidden vectors for every token from the input tag and the document, (iii) transfer learning for learning the data representation of inputs and (iv) classification to predict the value location used for IE.

Our information extraction problem was formulated as a question answering (QA) task, as the suggestion of BERT and ALBERT on the QA task (Devlin et al., 2019; Lan et al., 2020). The value is pulled from the document by querying the tag. In the model, transformers learn the context of the document given the tag (question) and produce hidden vectors for every token; the hidden representation layer adjusts the vectors towards to our domain. The rest of this section is described all parts of the model.

4.1. Input representation

As mentioned, we formulate our IE task as a question and answering problem. For QA, given a question, the QA model needs to output a corresponding answer. For our task, a tag can be considered as a question, and extracted information from a document can be considered as an answer.

From our formulation, each input includes the tag and a document (a bidding or sale document). The tag was treated as a single sequence, and the document was split into segments with a length of 384 tokens. Each token vector was represented based on three embeddings: token embedding, segment embedding, and position embedding. To differentiate among the input sequences, a special token ([SEP]) was inserted between them. For example, as showed in Fig. 4, the tag i has N_i tokens, in which $token_1$ corresponds to embedding E_1^i. The document has k sequences separated by token [SEP]. The detailed specification was described in the original BERT paper (Devlin et al., 2019).

4.2. Transformers

The transformer layer receives input tokens and transforms the tokens into vectors. This section investigates two recent success transformers, namely BERT (Devlin et al., 2019) and ALBERT (Lan et al., 2020).

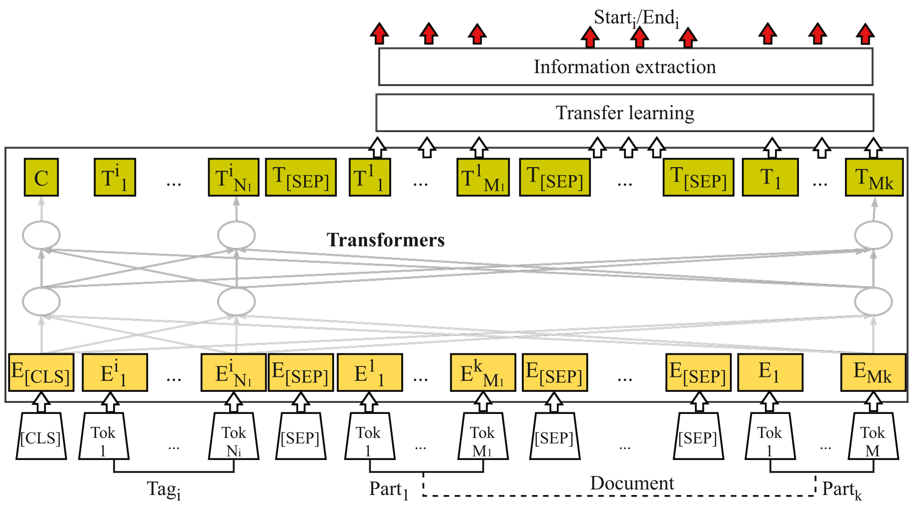


Fig. 4. The transformer-based model for extracting the value of a tag. The model receives a set of tags (each tag contains a set of tokens) and a document (including several parts). For training, with each tag (including a set of tokens), the model concatenates the tag with the relevant value (answer) to form a sequence. The sequence is fed into transformers for data representation. The output vector is used for training learning for domain adaptation. Finally, the information extraction layer predicts the start and end positions, which indicate the extracted information. For testing, given an input tag, the trained model pulls the corresponding answer from the document based on patterns learned from the training process.

BERT. BERT is a multi-layered bidirectional Transformer encoder, which allows our model to represent the context of a word by considering its neighbors (Devlin et al., 2019). This is unlike unidirectional models that learn the contextual representation for each word using the words on one side (left or right). For example, considering the word "bank" in two sentences "I went to the bank to deposit some money" and "I went to the bank of the river", the representations of "bank" are identical for a left-context unidirectional model, but they are distinguished with the use of BERT. This characteristic is compatible with our problem where many tags have homogeneous values. Let us take a look to Table 1, given an address, it may be the address for demand (tag 5), submitting the application of qualification (tag 20), submitting for of bid (tag 22), or place of the opening bid (tag 24). Therefore, in our problem, the context aspect is critical to determine the tag that the address belongs to.

In our research, a pre-trained BERT was employed for two reasons. Firstly, BERT has shown state-of-the-art performance on many NLP tasks, ranging over single/pair sentence classification, question answering, and sentence tagging. We, therefore, take advantage of BERT for our task of information extraction. Secondly, a pre-trained model is an appropriate solution to tackle our scenario of lacking training data. For example, we received only 100 documents for training and testing from our client. This is impractical to train the whole network from scratch. To tackle this problem, we decided to use BERT as a type of transfer learning to fine-tune our model on domain-specific business documents.

ALBERT. ALBERT is a lighter version of the original BERT, which incorporates two important techniques to reduce the number of parameters used in the model (Lan et al., 2020). The first one is a factorized embedding parameterization. Instead of projecting the one-hot vectors into a high-dimensional hidden space of size H, the model decomposes this step into two smaller steps. It first projects these vectors into a lower-dimensional embedding space size E, and then projects it into the hidden space. This reduces the embedding parameters significantly when E << H. The second technique is cross-layer parameter sharing, where all parameters are shared across multiple layers. This prevents the number of parameters from growing as the number of layers increases. For the aforementioned techniques, ALBERT also employed an inter-sentence coherent loss of the next sentence prediction task during the pre-training process.

In our study, we employed a pre-trained ALBERT model. The reason is similar to BERT, in which we adapt ALBERT to our IE task with a small number of training examples.

4.3. Transfer learning for new domains

As mentioned, we used transformers as transfer learning for our new domains, which lack annotated data. To do that, we followed the guideline of Devlin et al. (2019) and Lan et al. (2020), in which transformers were used to initialize parameters of our model and then the model was fine-tuned in downstream tasks. To fine-tune our model, neural network architectures were stacked on the top of transformers. Our intuition is that applying transformers is to take advantage of transformer models trained on a huge amount of data in different domains. By stacking neural networks, we adapted transformers to deal with domain-specific business documents by using transfer learning. This section investigates three well-known neural architectures for our transfer learning purpose: Convolutional Neural Network (CNN), Long-short Term Memory (LSTM), and bidirectional LSTM (BiLSTM).

4.3.1. CNN

CNN was originally designed for computer vision (LeCun and Bengio, 1998) and recently has achieved promising in NLP tasks (Kim, 2014; Shin et al., 2016; Nguyen et al., 2018; Yao et al., 2019). We take into account the efficiency of CNN for capturing the local context of a tag and an extracted sequence. CNN has two operations: convolution and pooling.

Convolution. The convolutional operation is to transform input data by using transformation functions. Given a sequence of tokens $X = (x_1, x_2, \dots, x_n)$, $x_i \in \mathbb{R}^k$ represents the embedding of this token learnt from transformers. Let $\boldsymbol{X}$ is the concatenation of n tokens of a tag and h is the kernel size (window size). The convolution operation applies the kernel size h to each position ith of the embedding matrix to produce the non-linear transformation of the input.

$$c_i^h = f(\boldsymbol{W}^h \times v_{i:i+h-1} + b) \tag{1}$$

where $f()$ is a non-linear transformation function, e.g. $tanh()$, $\boldsymbol{W}^h$ is the kernel weight matrix, b is a bias vector, and $v_{i:i+h-1}$ is the concatenation of vectors in the kernel size h. After convolution, each kernel size has a vector c_i. The feature map of a kernel size h is $\boldsymbol{C}^h = [c_1^h, \dots, c_{N-h+1}^h]$ (N is the maximal sequence length). We used multiple kernels to enrich representation (Kim, 2014).

Pooling. The pooling operation receives feature maps from convolution for removing unnecessary information. It includes a down-sampling operation to modify the input to retain important features of each feature map. It is possible to apply several types of pooling, we used max pooling due to its efficiency (Kim, 2014; Nguyen et al., 2018).

$$\hat{c}^h = \max(\boldsymbol{C}^h) \tag{2}$$

The $max()$ function retains most important values of $\boldsymbol{C}^h$ to forming final vectors. These vectors of pooling were used for the final layer for classification.

4.3.2. LSTM

LSTM is an extension of recurrent neural networks (RNN) and was invented to tackle the limitation of RNN for learning long sequences (Hochreiter and Schmidhuber, 1997). It uses the gate mechanism to control remember and forget information. LSTM has showed promising results in many NLP tasks (Hochreiter and Schmidhuber, 1997; Wang et al., 2017; Zhao et al., 2018). We used LSTM to take into account the dependency of tokens in a segment.

RNN includes a graphical model of inputs and states for learning hidden representation. Given an input sequence $X = (x_1, x_2, \dots, x_n)$, LSTM outputs an intermediate representation based on the hidden state of the current step t.

$$\boldsymbol{y}_t = \sigma(\boldsymbol{W}_y \boldsymbol{h}_t + \boldsymbol{b}_t) \tag{3}$$

where σ is the element-wise Sofmax function; $\boldsymbol{W}_t$ is a weight matrix and $\boldsymbol{h}_t$ is a bias vector learnt from the training; $\boldsymbol{h}_t$ is the current state updated by using a non-linear function from the previous state $\boldsymbol{h}_{t-1}$ as follows.

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t) \tag{4}$$

The hidden state h_t in Eq. (4) is computed by using the current input x_t and the previous hidden state h_{t-1} by using a non-linear function $f()$. The output of y_t in Eq. (3) by using the softmax function with the weight matrix $\boldsymbol{W}$ and a bias vector b_t.

For extension, a LSTM cell uses gates to control information, including: an input gate $\boldsymbol{i}_t$; a forget gate $\boldsymbol{f}_t$, an output gate $\boldsymbol{o}_t$, and a memory cell $\boldsymbol{c}_t$ to update the hidden state $\boldsymbol{h}_t$ as follows.

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{V}_{it-1} + \boldsymbol{b}_i),\\
\boldsymbol{f}_t &= \sigma(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{V}_{ft-1} + \boldsymbol{b}_f),\\
\boldsymbol{o}_t &= \sigma(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{V}_{ot-1} + \boldsymbol{b}_o),\\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{V}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c),\\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)
\end{aligned}$$

where $\odot$ is a multiplication function; $\boldsymbol{W}$ and $\boldsymbol{V}$ are weight matrices learnt from training; and $\boldsymbol{b}$ is a bias vector. The final hidden state s_t is the computation of the output gate o_t and the memory cell c_t. The output gate uses the current input x_t to compute the output representation by using $\boldsymbol{W}$ and $\boldsymbol{V}$ are weight matrices. The memory cell uses information from the input gate i_t and the forget gate f_t for its computation. The final hidden state encodes information of the whole sequence, that was used for classification.

4.3.3. Bi-LSTM

We also investigated an extension of LSTM, namely Bi-LSTM. Different from LSTM which learns data representation from left-to-right, Bi-LSTM combines forward and backward operations to represent the hidden state of the current step t. By combining two directions, Bi-LSTM has shown better performance than LSTM in several NLP tasks (Ju et al., 2018; Devlin et al., 2019).

A hidden state h_t was computed by concatenating a forward hidden state $\overrightarrow{h_t}$ and a backward hidden state $\overleftarrow{h_t}$. Its computation is described as follows.

$$\begin{aligned}
\overrightarrow{\boldsymbol{h}_t} &= LSTM(\overrightarrow{\boldsymbol{h}}_{t-1}, x_t),\\
\overleftarrow{\boldsymbol{h}_t} &= LSTM(\overleftarrow{\boldsymbol{h}}_{t+1}, x_t),\\
\boldsymbol{h}_t &= [\overrightarrow{\boldsymbol{h}_t}; \overleftarrow{\boldsymbol{h}_t}]
\end{aligned}$$

The backward and forward use a LSTM cell for encoding information. The concatenation of $\boldsymbol{h}_t$ was fed into the classification layer.

4.4. Information extraction

Information extraction is the last layer of our model. The extraction was formulated as classification that receives vectors from the representation layer for prediction. In our study, we investigated two classification methods: multi-layer perceptron with Softmax and Conditional Random Fields (CRFs).

Classification with softmax. After learning hidden representation, output vectors were fed to a fully-connected layer (multi-layer perceptron — MLP). On the top, classification was used by using softmax. The softmax predicts the position of the value. Each token is predicted to one of three outcomes including the start/end positions, and irrelevant to the tag. The extracted value was gathered based on the start to end positions with the highest probabilities.

CRFs. CRFs were utilized for predicting extracted sequences corresponding to tags. The reason is that CRFs were designed for sequence labeling (Lafferty et al., 2001) and stacking CRFs on LSTM or Bi-LSTM has achieved promising results for NER (Lample et al., 2016; Ju et al., 2018).

CRFs were designed to globally predict label sequences of any given sequences. Given an input sequence $X = (x_1, x_2, \dots, x_n)$ which is the output from the hidden representation layer, CRFs learn to make the prediction of x_i by maximizing the log probability during training. More precisely, given a sentence sequence $X = (x_1, x_2, \dots, x_n)$ and the corresponding state of sequence (labels) $Y = (y_1, y_2, \dots, y_n)$, the probability of Y conditioned on X was defined as $P(Y|X)$. When making prediction of y_i, CRFs considers the current input x_i (the current word) and previous states of previous words as $P(y_i = 1|X)$.

Prediction with CRFs can be considered as a sequence labeling task. Given an input document including segments, CRFs classifies each segment is an extracted value, corresponding to a tag or not.

4.5. Training

The training process of our model was done in two stages: (i) pre-training and (ii) fine-tuning. For the first stage, the pre-trained weights of BERT and ALBERT were reused, while the weights of the rest layers were generated with a truncated normal distribution. In training, our model used parameters from pre-trained transformers and fine-tuned these parameters on new domains. To consider our work's impact, our architecture was applied to the most notorious pre-trained models as BERT and ALBERT. We used a multilingual BERT-Base model trained for 102 languages on a huge amount of texts from Wikipedia. The BERT was trained with a large text corpus of Japanese collected from Wikipedia. The training task is to predict whether a sentence is the next or just a random of other sentences (Devlin et al., 2019). The setting is the same as the original paper (Devlin et al., 2019). The BERT model has 12 layers, a hidden layer of 768 neurons, 12 heads, and 110M parameters. For ALBERT, we used the pre-trained Japanese model with 12 layers, the hidden size of 768, the embedding size of 128 with 12M parameters (Lan et al., 2020).

For training, the whole network was fine-tuned in 20 epochs with our training data by using the cross-entropy loss function. We used the same hyperparameters of our model for both BERT and ALBERT as follows.

Table 4
The parameters for training our model.

Parameter	Value
Number of training epochs	20
Convolutional kernel size	3
Convolutional output size	768
LSTM output size	128
Training batch size	24
Number of GPUs for BERT	1
Number of GPUs for ALBERT	1

5. Statistical analysis

5.1. Settings

We focus our impact on Japanese data sets, which were provided by our clients. For the bidding dataset, we used 78 documents for training and 22 documents for testing. For the sale dataset, we used 300 documents for training and 165 documents for testing. The parameters are shown in Table 4.

5.2. Baselines

To verify the efficiency of our approach, we compared our model to seven baselines. The baselines are in two groups: transformers-based methods and non-transformers-based methods as follows.

- **BERT**: is the basic model which obtains state-of-the-art performance on many natural language processing tasks, including QA (Devlin et al., 2019). We directly applied BERT for QA on our testing data, without any additional training as Zhang et al. (2020).
- **BERT+CRF**: stacks CRF on BERT for sequence labeling. This is because CRF is a conventional method for information extraction and NER (Lample et al., 2016; Andrew, 2018). This method was trained on training data and then applied to test data.
- **BERT+LSTM+CRF**: This model uses LSTM to capture the hidden representation of sequences and employs CRF for classification. This is a variation of LSTM–CRF for NER (Lample et al., 2016). We also tried with BiLSTM. Its results are shown in Table 6.
- **ALBERT**: This baseline directly uses the pre-trained model of ALBERT (Lan et al., 2020). Input sequences were fed into ALBERT to get embeddings as inputs for MLP for classification.
- **ALBERT+CNN+MLP**: This is a variation of ALBERT. We used the same architecture with BERT+CNN+MLP for classification.
- ***n*-grams+MLP+regex**: was trained with n-gram features (n in $[1, 4]$). The MLP was used to predict the paragraph containing the values of tags. The regular expression was finally applied to extract the values.
- **CNN+BiLSTM+CRF**: We used a deep neural network including layers of convolution, bidirectional Long Short Term Memory (BiLSTM), and conditional random fields (CRFs) to automatically extract tags of input texts (Lample et al., 2016). For token embedding, we used Glove due to its efficiency in representing the meaning of tokens (Pennington et al., 2014).

5.3. Evaluation metrics

We used F-score to evaluate the performance of our model as well as the baselines. Extracted outputs were matched to ground-truth data to compute precision, recall, and F-score. The final F-score was computed on all tags.

Table 5
Comparison of methods according the average of F1-score. **Bold** is the best and *italic* is the second best.

Method	Bidding documents	Sale documents
BERT (QA)	0.8607	*0.8456*
BERT+CRF	0.1773*	0.1254*
BERT+LSTM+CRF	0.3817*	0.5572*
ALBERT	*0.8655*	0.7734*
ALBERT+CNN+MLP	0.8587	0.7878*
n-grams+MLP+Regex	0.8523	0.7177*
CNN+BiLSTM+CRF+MLP	0.6766*	0.7513*
BERT+CNN+MLP (Our model)	**0.9062**	**0.8614**

*Shows that our model is significantly better with p-value ≤ 0.05.

6. Results and discussion

This section shows the comparison of our model to the baselines. We first report F-scores and then discuss the contribution of transfer learning and transformers in our model. Next we observe the relationship between the number of training data with the performance of our model. We final show an error analysis of the outputs of our model.

6.1. F-score comparison

Table 5 shows the comparison according to the F1-score of our model with the baselines. We can observe that our model consistently achieves better F-scores than baselines. In many cases, our model significantly outperforms the baselines with large margins (the significant test was conducted by using pairwise t-test[4] with the p-value ≤ 0.05). The improvement comes from two factors. Firstly, our model exploits the efficiency of BERT trained on a large amount of data; therefore, it can potentially capture the hidden representation of data. The hidden representation facilitates the learning process by providing high-quality data representation for CNN. By stacking CNN and fine-tuning the model with a small number of training data in downstream tasks, our model has the ability to adapt to new domains. For example, by only using 78 training bidding documents, our model can adapt to a new domain. As a result, the model can correctly extract information on bidding documents and can improve the performance over BERT-QA of 4.55%. Secondly, we take advantage of CNN for fine-tuning to capture the local context of tag-value pairs. This confirms our assumption in Section 1 that we can utilize transfer learning for information extraction in narrow business domains with limited data.

The BERT-QA model is competitive on the two datasets. It is understandable that this model also uses BERT as a pre-trained model to extract hidden representation. However, it needs to be adapted to new domains. It shows the efficiency of retraining the model on new domains. Even with a small amount of data (e.g. 78 bidding documents), our model can improve the F-score. However, the gap between BERT-QA and our model on sale documents is small (4.55% of F-score), showing that the original BERT is a strong baseline of information extraction in business domains. The variations of BERT do not show improvements. For BERT–CRF, it may lack the patterns learned from training data because it directly uses outputs from BERT for CRF to do classification. For BERT-LSTM-CRF, the possible reason is bidding documents are very long (Table 3); therefore, long-term dependencies may affect these models. By contrast, BERT+LSTM+CRF improves the F-score on sale documents. A possible reason is that sale documents are much shorter than bidding documents, then this model can learn the dependency between tags and values for prediction. The model using ALBERT is the second best on bidding documents but outputs low results on sale documents. It is understandable that

[4] https://docs.scipy.org/doc/scipy-0.19.0/reference/generated/scipy.stats.ttest_ind.html.

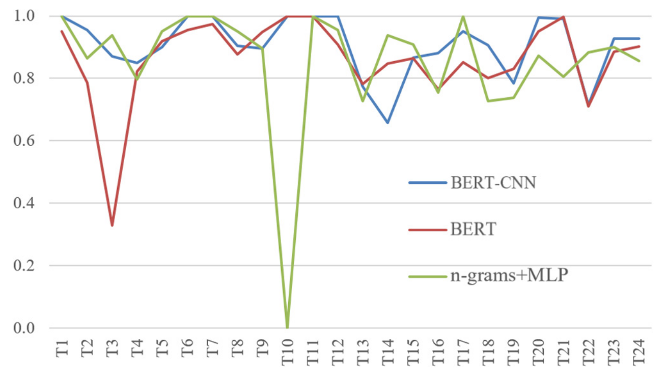


Fig. 5. The F1-score per tag.

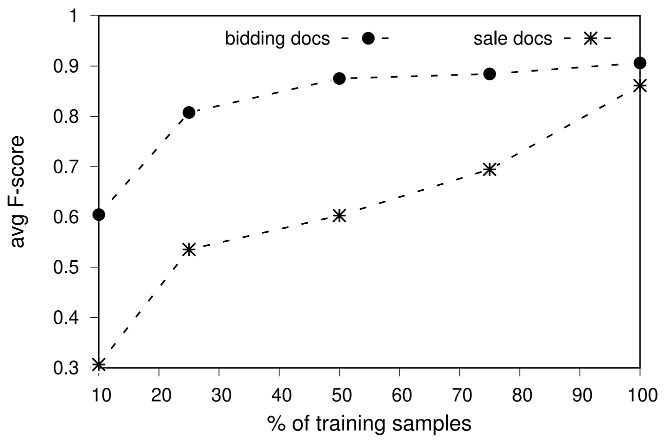


Fig. 6. The average F-score of our model on each data segment on two datasets.

ALBERT is the lighter version of BERT with much smaller number of parameters (12M vs. 110M). However, the gap is not so much different, especially on bidding documents. It confirms the efficiency of ALBERT in retaining high accuracy with a compressed model (Lan et al., 2020). The combination of ALBERT with CNN gives better results than the original ALBERT model. It supports our idea in stacking CNN for domain adaptation.

The model using *n*-grams features achieves competitive results. This is because this model uses two steps of candidate extraction. In the first step, *n*-grams features were fed into MLP to detect whether a sentence contains extracted information or not. The second step uses regular expressions to extract candidates. However, in some cases, the expressions cannot cover the patterns of text (please refer to Section 6.5). An interesting point is that the model CNN+BiLSTM+CRF outputs a low score on bidding documents but gives a promising result on sale documents. This trend is similar to BERT+LSTM+CRF. As pointed out, the model using CRF for classification may have challenges on long documents (bidding documents). Also, training deep neural networks from scratch with limited data is difficult to convergence because we need to optimize a large number of parameters. In contrast, the feature-based method using *n*-grams features requires less training data but it easily suffers from the definition of regular expressions and needs to adapt to new domains.

We observed the F-score on each tag to analyze how each model works on bidding documents. To do that, we matched extracted text to reference and compute scores, which are plotted in Fig. 5. We did not plot the results of BERT+CRF, BERT+LSTM+CRF, and CNN+CNN+CRF+MLP due to its low scores. Also, we only report on bidding documents due to space limitations.

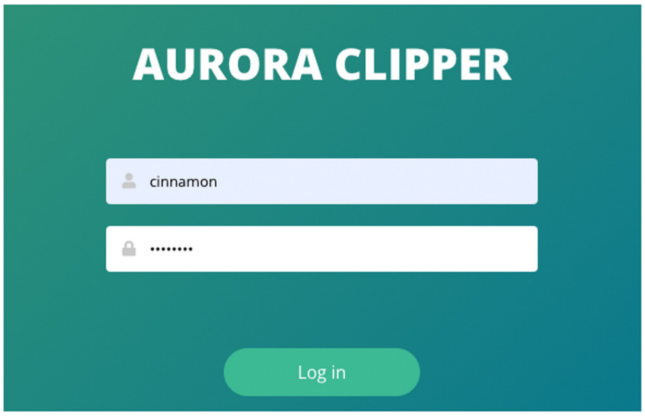


Fig. 7. The interface of the login function.

Table 6
The F1-score of variations of BERT.

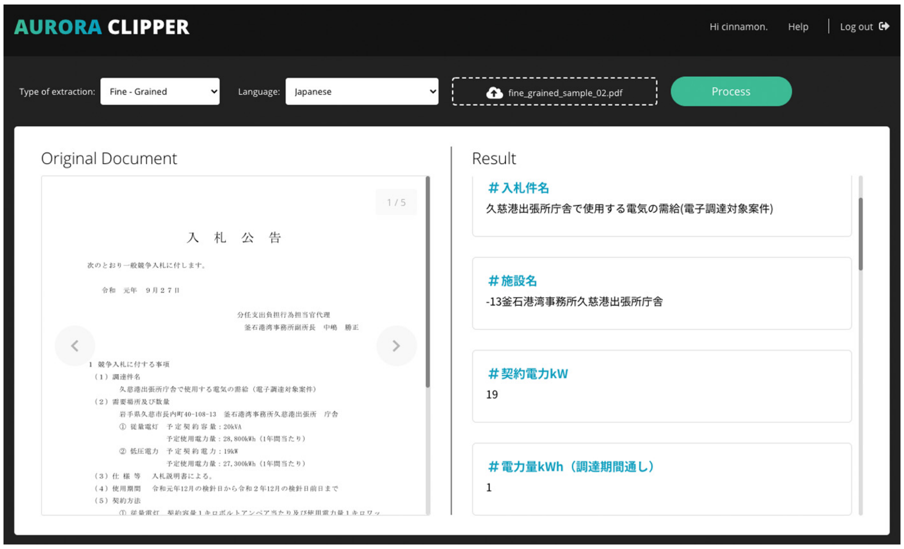
Method	Bidding documents	Sale documents
BERT+CNN + MLP	**0.9062**	**0.8614**
BERT+LSTM + MLP	0.8546	0.7829
BERT+BiLSTM + MLP	0.8737	0.8185
BERT+CNN + CRF	0.4250	0.4882
BERT+LSTM + CRF	0.3817	0.4680
BERT+BiLSTM + CRF	0.4621	0.5572

As can be seen, our model works stably on all tags and usually achieves the best F1 scores in comparison with BERT-QA and *n*-grams-MLP. This again supports results in Table 5 where our model is the best. It should be noted that the feature-based method obtains 0% of F-score for tag 10 (classification of reserved electric energy). The reason is that *n*-grams features and the regular expressions may not capture well the patterns of reserved electric energy (tag 10). This shows that the *n*-grams+MLP+regex method can be affected by the definition of regular expressions. As a result, it could not find any values for tag 10 on the test set. By contrast, our model trained on BERT can exploit rich hidden representation from large pre-trained data, and with 78 training documents of bids, the model can correctly extract tag 10.

6.2. *Transfer learning observation*

This section observes the contribution of the transfer learning and classification layers. To do that, we fixed the transformer layer and relaxed the transfer and classification layers. For the transfer learning layer, we tried the combination of BERT with LSTM and BiLSTM. This is because these architectures have achieved promising results for many NLP problems. The classification layer was tested with CRF. After combining, these combined models were re-trained and tested on test data. Our observation is showed in Table 6.

We can observe two interesting points. Firstly, the model using CNN+MLP with softmax classification outputs the best F-scores. As mentioned, CNN can exploit the local context of tag-value pairs in documents, then this model can learn data representation well. In other cases, models using BiLSTM (BERT+BiLSTM+MLP and BERT+BiLSTM+CRF) achieve better F-scores than those using LSTM. This is because BiLSTM encodes information in forward and backward, so hidden representation may be better than only using forward (LSTM). Secondly, the classification layer using MLP and softmax outputs better F-scores than that using CRF. As explained, tag-value pairs are in different positions in a document; therefore, the lack of dependency among pairs may challenge CRF for prediction.



Type of extraction: Fine - Grained

Language: Japanese

fine_grained_sample_02.pdf

入　札　公　告

次のとおり一般競争入札に付します。

令和　元年　９月２７日

分任支出負担行為担当官代理
釜石港湾事務所副所長　中嶋　勝正

１　競争入札に付する事項
（１）調達件名
久慈港出張所庁舎で使用する電気の需給（電子調達対象案件）
（２）需要場所及び数量
岩手県久慈市長内町40-108-13　釜石港湾事務所久慈港出張所　庁舎
①　従量電灯　予定契約容量：20kVA
予定使用電力量：28,800kWh（1年間当たり）
②　低圧電力　予定契約電力：19kW
予定使用電力量：27,300kWh（1年間当たり）
（３）仕　様　等　　入札説明書による。
（４）使用期間　　令和元年12月の検針日から令和２年12月の検針日前日まで
（５）契約方法
①　従量電灯　契約容量１キロボルトアンペア当たり及び使用電力量１キロワッ

＃入札件名
久慈港出張所庁舎で使用する電気の需給(電子調達対象案件)

＃施設名
-13釜石港湾事務所久慈港出張所庁舎

＃契約電力kW
19

＃電力量kWh（調達期間通し）
1

Fig. 8. The interface of our IE system for Japanese business documents.

Table 7
The F1-score of variations of transformers.

Method	Bidding documents	Sale documents
BERT+CNN+MLP	**0.9062**	**0.8614**
BERT+LSTM+MLP	0.8546	0.7829
BERT+BiLSTM+MLP	0.8737	0.8185
ALBERT+CNN+MLP	0.8587	**0.7878**
ALBERT+LSTM+MLP	0.8671	0.7036
ALBERT+BiLST+MLP	**0.8753**	0.7119

6.3. Transformer contribution

This section reports our observation of the contribution of transformers. To do that, BERT and ALBERT were used as the embedding layer, combined with CNN, LSTM, and BiLSTM. The classification is MLP with softmax because it gives the best results (Tables 5 and 6). The comparison is showed in Table 7.

We can observe that the BERT-based methods achieve promising results, in which its performance is better than that of the ALBERT-based methods. The trend is similar to Table 5 in which two ALBERT-based methods do not achieve the best F-scores on the two datasets. Among those methods, the models using CNN+MLP obtain better F-scores than those using LSTM+MLP and BiLSTM+MLP. It supports our hypothesis in using CNN to capture the local context of tag-value pairs. Also, business documents are quite long. It may challenge the models using LSTM and BiLSTM.

6.4. The efficiency of transformers with training data

For better understanding the behavior of our model on training data, we observed the relationship between F-scores and the number of training samples. To do that, training data was segmented into different parts. More precisely, training data was divided into 10%, 25%, 50%, 75%, and 100%. For segmentation, we suffered the training data and kept the samples which can cover the tags as much as we can. This is because the number of tags in each document is different. After that, our model was trained on each segment and its average F-score is plotted in Fig. 6.

We can observe that changing the number of training examples affects our model. Generally, increasing training samples improves the quality of the extraction. The model is dramatically improved from 10% to 25%. This is because using only 10% of training data is not enough for training. By adding more 15%, the model can partly learn hidden patterns from data. However, there is a difference after 25%. For bidding documents, the performance slightly increases even adding more data. It seems that the BERT pre-trained model can cover the patterns of bidding documents. In contrast, for sale documents, the performance still raises when adding more training samples. Even using 75% of training data, there exists a large gap between the two models, using 75% and 100% training data (0.6946 vs. 0.8614). It shows that we should not cut-off the number of training data for sale documents. Also, it is interesting to test our model with more training examples for sale documents.

6.5. Error analysis

In this section, we investigate how our model works on real data of bids. We observed the output of our model and the baselines. Table 8 shows some predicted values for three tags 21, 22, 23 (Table 1) in documents of two contracts. The first part shows a case that our model does not work and the second part presents a case that our model extracts correct sequences. We did not show the outputs of sale documents due to space limitations.

After observing, we found that for the feature-based method, it may suffer from the out of vocabulary problem because regular expressions were created by on keywords defined by humans. Although the classifier can predict correctly the sentence containing the value, the regular expression fails to locate the information. Taking the contract 0553 as an example, since the address of the company is not in the training data, the regular expression only captures the number part. Unlikely, the deep models can predict both the text and number parts in this case. However, the model trained from scratch usually produces uncompleted text, especially with the long-text values. This comes from the lack of training data that is one of the essential keys to allow the convergence of a deep model. In contrast, the pre-trained model can predict the whole content.

We also found that the feature-based methods can predict short values like date times, and numbers efficiently. In this case, the classifier finds the sentence using features and the regular expression can locate the information easily. For these types of fields, the performance of the feature-based and the deep models is similar. This is shown in Fig. 5.

6.6. Our IE system

We public our system in which users can use our system to extract information from real bidding and sale documents. Please connect

Table 8
The correct answers and correct predictions are highlighted in blue. We did not show outputs of BERT, BERT+CRF and BERT+LSTM+CRF due to space limitation.

File	**中国0041**
Tag	**Correct answer**
21	岡山市東区役所瀬戸支所1階総務民生課 (Okayama City Higashi Ward Office Seto Branch 1F General Affairs and Welfare Division).
22	〒7000899備前瀬戸郵便局留 (Bizen Seto Post Office Station, postal code: 7000899).
23	岡山市役所東区役所瀬戸支社 (Okayama City Hall Higashi Ward Office Seto Branch Office).
	n-gram MLP
21	岡山市東区役所瀬戸支所1階総務民生課
22	〒7000899備前瀬戸郵便局留
23	岡山市役所東区役所瀬戸支所宛
	CNN+BiLSTM
21	岡山市東区瀬戸町瀬戸45番地岡山市東区役所瀬戸支所宛
22	〒700-0899備前瀬戸郵便局留岡山市役所東区役所瀬戸支所宛
23	〒700-0899備前瀬戸郵便局留岡山市役所東区役所瀬戸支所宛
	Our model
21	瀬戸支所総務民生課
22	岡山市東区瀬戸町瀬戸45番地
23	瀬戸支所総務民生課
File	**九州0553**
Tag	**Correct answer**
21	宮崎県会計管理局物品管理調達課物品調達担当 (Miyazaki Prefecture Accounting Administration Department Goods Management Procurement Section Goods Procurement).
22	〒882-0835延岡市新小路2丁目1番地10 (postal code 882-0835 Miyazaki, Nobeoka, Shinkoji, 2 Chome—1—10).
23	県立延岡病院総務課整備担当 (Prefectural Nobeoka Hospital General Affairs Division Maintenance Section).
	n-gram MLP
21	(232)6118
22	目1番地10郵便番号882-0853電話番号098
23	(232)6181
	CNN+BiLSTM
21	目1番地10郵便番号882
22	延岡市新小路2丁
23	延岡市新小路2丁
	Our model
21	宮崎県会計管理局物品管理調達課物品調達担当
22	〒882-0835延岡市新小路2丁目1番地10
23	宮崎県会計管理局物品管理調達課物品調達担当

this site via a server address at https://aurora-demo.cinnamon.is. Due to the security of our company, please login with the username as `cinnamon` and password as `cinnamon`. Fig. 7 shows our online system.

After logging with the username and password, users can use the extraction functions of our system shown in Fig. 8. Users can read the help and download the sample data by clicking on the menu help, on the top right corner. In the context of this paper, pleas try with fine-grained IE. Of course, you can also try with whole paragraph extraction and cause–effect analysis with our samples.

After uploading a sample, the left side of Fig. 8 is the original document, the right side is extracted information. The interface shows some tags in Table 1 such as prefecture (tag 2), the title of bidding (tag 3), name of institution (tag 4), contracted electric energy (tag 8), amount of electric energy (tag 9), public announcement date (tag 12), the deadline for delivery the specification (tag 13).

7. Conclusion

This paper presents a practical model for information extraction from Japanese business documents. The model provides a way for dealing with the challenging problem of limited training data when building extraction systems in actual cases. To do that, the extraction adapts pre-trained general language models to new domains by using transfer learning, that uses an additional CNN layer for data representation. We conduct experiments and analyses on two actual datasets. Promising results confirm that our model can achieve high accuracy that satisfies the need of clients in their businesses with a small number of training samples. The model can be used to reduce paper-work and extract nugget information for data transformation in actual business cases. The analysis indicates that the model works well in two domains with the different number of tags, showing that it can be viable alternative to extraction-based systems.

The future work will confirm the efficiency of our model on other business text genres with limited training data. Also, more sophisticated models should be considered, e.g. using attention. Finally, we would like to test our model with other transformers such as ELECTRA for Japanese.

CRediT authorship contribution statement

Minh-Tien Nguyen: Conceptualization, Formal analysis, Investigation, Resources, Writing - original draft, Supervision, Funding acquisition. **Dung Tien Le:** Software, Validation, Data curation, Project administration. **Linh Le:** Methodology, Software, Validation, Investigation.

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.

Acknowledgments

We would like to thank to Kiet T. Nguyen and Tung H. Phan for building the system; Truong N. V. Nguyen for creating the video; Akira Shojiguchi for giving useful comments and discussion; and comments of reviewers and anonymous readers for improving our paper. This research is funded by Hung Yen University of Technology and Education under the grant number UTEHY.L.2020.04.

Appendix. The main code of our BERT-based model

```
def create_model(bert_config, is_training, input_ids, input_mask,
                                  segment_ids, use_one_hot_embeddings):
    """Creates a classification model."""
    model = modeling.BertModel(config=bert_config,
           is_training=is_training,
           input_ids=input_ids,
           input_mask=input_mask,
           token_type_ids=segment_ids,
           use_one_hot_embeddings=use_one_hot_embeddings)
    final_hidden = model.get_sequence_output()
    final_hidden_shape = modeling.get_shape_list(final_hidden,
         expected_rank=3)
    batch_size = final_hidden_shape[0]
    seq_length = final_hidden_shape[1]
    hidden_size = final_hidden_shape[2]
    output_weights = tf.get_variable(
        "cls/squad/output_weights", [2, hidden_size],
        initializer=tf.truncated_normal_initializer(stddev=0.02))
    output_bias = tf.get_variable(
        "cls/squad/output_bias", [2],
              initializer=tf.zeros_initializer())
    conv1d = tf.layers.conv1d(inputs=final_hidden,
         filters=hidden_size, kernel_size=3, padding='same',
         activation=tf.nn.elu)
    conv1d = tf.reshape(conv1d, [batch_size * seq_length,
         hidden_size])
    logits = tf.matmul(conv1d, output_weights, transpose_b=True)
    logits = tf.nn.bias_add(logits, output_bias)
    logits = tf.reshape(logits, [batch_size, seq_length, 2])
    logits = tf.transpose(logits, [2, 0, 1])
    unstacked_logits = tf.unstack(logits, axis=0)
    (start_logits, end_logits) = (unstacked_logits[0],
         unstacked_logits[1])
    return start_logits, end_logits
```

References

Andrew, J.J., 2018. Automatic extraction of entities and relation from legal documents. In: Proceedings of the Seventh Named Entities Workshop, pp. 1-8.

Angeli, G., Premkumar, M.J.J., Manning, C.D., 2015. Leveraging linguistic structure for open domain information extraction. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). ACL, pp. 344–354.

Del Corro, L., Abujabal, A., Gemulla, R., Weikum, G., 2015. Finet: context-aware fine-grained named entity typing. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, pp. 868–878.

Del Corro, L., Gemulla, R., 2013. Clausie: clause-based open information extraction. In: Proceedings of the 22nd International Conference on World Wide Web. ACM, pp. 355–366.

Deng, L., Liu, Y., 2018. Deep Learning in Natural Language Processing. Springer.

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 (NAACL), Volume 1 (Long and Short Papers). ACL, pp. 4171–4186.

Finkel, J.R., Manning, C.D., 2009. Nested named entity recognition. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP), Volume 1-Volume 1. ACL, pp. 141–150.

Fleischman, M., Hovy, E., 2002. Fine grained classification of named entities. In: Proceedings of the 19th International Conference on Computational Linguistic (COLING), Volume 1. ACL, pp. 1–7.

Herbert, L., 2017. Digital Transformation: Build your Organization's Future for the Innovation Age. Bloomsbury Publishing.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Inmon, W.H., Nesavich, A., 2007. Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence. Pearson Education.

Ju, M., Miwa, M., Ananiadou, S., 2018. A neural layered model for nested named entity recognition. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL), Volume 1 (Long Papers). ACL, pp. 1446–1459.

Kim, Y., 2014. Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, pp. 1746–1751.

Lafferty, J., McCallum, A., Pereira, F.C., 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of the 18th International Conference on Machine Learning (ICML), pp. 282–289.

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 (NAACL). ACL, pp. 260–270.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R., 2020. Albert: a lite bert for self-supervised learning of language representations. In: International Conference on Learning Representations.

LeCun, Y., Bengio, Y., 1998. Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks, Cambridge, MA, USA, pp. 255–258.

Lee, C., Hwang, Y.G., Oh, H.J.L.S., Heo, J., Lee, C.H., Kim, H., Wang, J., Jang, M.G., 2006. Fine-grained named entity recognition using conditional random fields for question answering. In: Asia Information Retrieval Symposium (AIRS). Springer, pp. 581–587.

Lin, J.C.W., Shao, Y., Zhou, Y., Pirouz, M., Chen, H.C., 2019. A bi-lstm mention hypergraph model with encoding schema for mention extraction. Eng. Appl. Artif. Intell. 85, 175–181.

Manyika, J., Chui, M., Miremadi, M., 2017. A future that works: ai, automation, employment, and productivity. Technical Report, McKinsey Global Institute Research, Tech. Rep, 60.

Nguyen, M.T., Phan, V.A., Linh, L.T., Son, N.H., Dung, L.T., Hirano, M., Hotta, H., 2019. Transfer learning for information extraction with limited data. In: Proceedings of 16th International Conference of the Pacific Association for Computational Linguistics. Springer, pp. 469–482.

Nguyen, M.T., Tran, D.V., Phan, V.A., Nguyen, L.M., 2018. Towards social context summarization with convolutional neural networks. In: Proceedings of the 19th International Conference on Computational Linguistics and Intelligent Text Processing (CICLING).

Pan, S.J., Yang, Q., 2009. A survey on transfer learning. IEEE Trans. Knowl. Data Eng. 22 (10), 1345–1359.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). ACL, pp. 1532–1543.

Sang, E.F., De Meulder, F., 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. arXiv Preprint Cs/0306050.

Shimaoka, S., Stenetorp, P., Inui, K., Riedel, S., 2016. An attentive neural architecture for fine-grained entity type classification. In: Proceedings of the 5th Workshop on Automated Knowledge Base Construction, pp. 69-74.

Shin, H.C., Roth, H.R., Gao, M., Lu, L., Xu, Z., Nogues, I., Yao, J., Mollura, D., Summers, R.M., 2016. Deep convolutional neural networks for computer-aided detection: cnn architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imag. 35 (5), 1285–1298.

Tur, G., De Mori, R., 2011. Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. John Wiley and Sons.

Wang, L., Jiang, J., Chieu, H.L., Ong, C.H., Song, D., Liao, L., 2017. Can syntax help? improving an lstm-based sentence compression model for new domains. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). ACL, pp. 1385–1393.

Watanabe, Y., Asahara, M., Matsumoto, Y., 2007. A graph-based approach to named entity categorization in wikipedia using conditional random fields. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). ACL, pp. 649–657.

Weiss, K., Khoshgoftaar, T.M., Wang, D., 2016. A survey of transfer learning. J. Big Data 3 (1), 9.

Yao, L., Mao, C., Luo, Y., 2019. Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33), pp. 7370-7377.

Zhang, R., Yang, W., Lin, L., Tu, Z., Xie, Y., Fu, Z., Tan, L., Xiong, K., Lin, J., 2020. Rapid adaptation of bert for information extraction on domain-specific business documents. arXiv Preprint arXiv:2002.01861.

Zhao, Y., Luo, Z., Aizawa, A., 2018. A language model based evaluator for sentence compression. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). ACL, pp. 170–175.