Understanding Transformers for Information Extraction with Limited Data

Minh-Tien Nguyen
CINNAMON LAB, 10th floor,
Geleximco building, 36 Hoang Cau,
Dong Da, Hanoi, Vietnam.
Hung Yen University of Technology and
Education Hung Yen, Vietnam.
tiennm@utehy.edu.vn

Dung Tien Le, Nguyen Hong Son,
Bui Cong Minh, Do Hoang Thai Duong,
Le Thai Linh
CINNAMON LAB, 10th floor,
Geleximco building, 36 Hoang Cau,
Dong Da, Hanoi, Vietnam.
{nathan,levi,matt,howard}@cinnamon.is
linh.le@uq.edu.au

Abstract

Transformers have recently achieved promising results in many natural language processing tasks; however, the understanding of transformers for information extraction in business scenarios is still an open question. This paper bridges the gap by introducing an investigation to understand the behavior of transformers in extracting information from domain-specific business documents. To do that, we employ transformers for taking advantage of these architectures trained on a huge amount of general data and fine-tune transformers to our down-stream IE task by using transfer learning. Experimental results on three Japanese datasets show that there are small margins among transformers in terms of F-scores but some models can achieve high accuracy with a small number of training data.

1 Introduction

The significant growth of data provides a chance for humans to approach information from many sources. Yet, it also makes an obstacle for distilling useful knowledge. To address this issue, information extraction (IE) can be considered as an appropriate solution for converting unstructured to structured data. From the research side, due to its large impact, IE has received attention from the research community with many studies (Corro and Gemulla, 2013; Angeli et al., 2015; Nguyen et al., 2019). From the business site, IE is a crucial step for digital transformation (Inmon and Nesavich, 2007; Herbert, 2017; Lin et al., 2019). The outputs of IE systems can be used in many natural language processing (NLP) applications, e.g. question answering, information retrieval (Shimaoka et al., 2016), or the automatic generation of ontology (Fleischman and Hovy, 2002).

The recent success of transformers draws a new direction for many NLP tasks. For example, BERT (Devlin et al., 2019) pioneers to creating a contextual language model for language understanding. As a result, BERT has achieved promising results on many NLP tasks, including IE. Following the success of BERT, a lot of transformer architecture has developed such as ALBERT (Lan et al., 2020), DistillBERT (Sanh et al., 2019), or ELECTRA (Clark et al., 2019). It leverages the adaptation of transformers for IE. For example, (Nguyen et al., 2019) adapted BERT to extract information from business documents. These studies achieved promising results; however, we argue that there exist gaps that limit the understanding of transformers for IE from domain-specific business documents. The first gap is that previous work only investigates the IE task with one transformer model, e.g. BERT. The second gap is that several important aspects of transformers were not studied well, e.g. the relationship between the number of training samples and performance.

This paper bridges the two gaps by investigating the behavior of transformers for extracting information from business documents, in actual scenarios. To do that, we empower IE models by using transformers in the form of transfer learning. Precisely, transformers are used to utilize the power of these models trained on the huge amount of general data. Then the transformer-based IE models are fine-tuned in the downstream IE task. By using transform-

ers for transfer learning, we simulate actual business cases that have a small number of training data. This paper makes three main contributions:

- It analyzes the behavior of transformers for IE in the context of business scenarios. The analysis examines the transformers in three aspects: performance comparison, the relationship between performance and the number of training samples, and training time. To the best of our knowledge, we are the first conducting the comprehensive investigation for IE from domain-specific business documents in a low-resource language, i.e. Japanese.
- It introduces a public dataset[1] for the IE task of business documents. The dataset mimics actual business cases in which IE models are trained with a small number of training data.
- It releases a pre-trained model[2] based on ELECTRA (Clark et al., 2019), which facilitates studies of NLP tasks on Japanese.

2 Related Work

Information extraction is an important task of NLP and has investigated in a long time with many studies. There are two main approaches for IE, using dictionaries (Watanabe et al., 2007) and machine learning (Corro and Gemulla, 2013; Angeli et al., 2015; Lample et al., 2016). The first approach usually defines a dictionary for extracting information. Input documents are parsed to tokens which are matched to each item in the dictionary for extraction. The second approach usually uses training data to train a classifier that can distinguish extracted or non-extracted information (Corro and Gemulla, 2013; Angeli et al., 2015; Lample et al., 2016). Using a dictionary-based method can achieve high accuracy, but it is time-consuming and labor-expensive for dictionary preparation. In contrast, machine learning models exploit linguistic features (Angeli et al., 2015) or hidden features learned from LSTM for classification (Lample et al., 2016). As a result, it can reduce the cost of dictionary maintenance and easy to adapt to other domains. In practice, several research projects focus on the nested named entities and have great progress so far (Finkel and Manning, 2009; Lample et al., 2016).

NER is a specific task of IE in which high-level concepts such as people, places, organizations usually need to extract. For example, CoNLL 2003 defined four types of entities, including locations, mixed entities, organizations, and persons (Sang et al., 2003). However, for document analysis in practical business cases, entity types should be at a more detailed level (Corro et al., 2015; Nguyen et al., 2019). To address this problem, fine-grained entity extraction was introduced and applied to several 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). The recent success of transformers draws a new method for NER. BERT (Devlin et al., 2019), ALBERT (Lan et al., 2020), DistillBERT (Sanh et al., 2019), and ELECTRA (Clark et al., 2019) are four pre-trained transformers which achieve promising results many NLP tasks. This paper employs the power of those transformers as transfer learning for our IE problem. This employment allows us to simulate our business cases which only have a small number of training samples.

The work of (Nguyen et al., 2019) is perhaps the most relevant to our task. In this paper, the authors adapted BERT for extracting information from domain-specific documents. However, understanding the behavior of BERT in terms of extracting from business documents is still an open question. We dig a deeper level to observe IE models by comparing four transformers. We believe that this comparison provides a comprehensive analysis of transformers for such IE task in actual business cases.

3 Task Definition and Data Preparation

3.1 Task definition

As mentioned, we deal with the task of IE with limited data for business documents. Given a document and pre-defined tags (keywords), IE models need to extract corresponding information to the tags. Formally, the task can be formally defined as follows.

- **Input**: a document and a set of tags.
- **Output**: extracted information corresponding to the tags.

[1] https://github.com/DungLe13/bidding-dataset

[2] https://github.com/thaiduongx26/electra_japanese

Our IE task is quite different from the common NER task in which we need to extract a large number of entity types, e.g. 24 (Table 1) while the common NER task extracts a small number of entities, e.g. four types of CoNLL. Also, due to the restriction of actual business cases, we use a small number of training samples instead of using a large number of training examples e.g. around 15,000 samples in CoNLL (Sang et al., 2003).

3.2 Data preparation

It is hard to use published datasets, e.g. CoNLL (Sang et al., 2003) for comparison due to our different purpose with common NER tasks. We, therefore, prepared three datasets, for testing IE models.

3.2.1 CinData

Because there are gaps in using common IE datasets to our task, we created a new corpus named CinData. To do that, we collected 124 public Japanese bidding documents from the Japan Oil, Gas and Metals National Corporation (JOGMEC).[3] Each document is a public notice, which outlines the information about the bidding process, including the dates of the contract, the deadlines for submission, and the contacts of the department or person in charge. These documents are raw texts, so we need to define a set of tags for the annotation process. To do that, we consulted our legal team for the definition. The discussion and definition were internally conducted. Finally, we defined 19 names that represent the categories of extracted information, which we formally refer to as ``tags". The list of tags covers common important information of a bidding document. The list is unique and remains unchanged in all three train/dev/test sets. Please refer to the Appendix for the description of tags.

The collected documents are `PDF` files, so they were converted to the text format for easy use. To do that, we used `pdfplumber`,[4] as a parser, combined with heuristic rules: bullets, numberings, indentication, title, table for keeping the structure of documents. After parsing, our QAs (quality assurance - people who have at least the N3 Japanese-Language Proficiency Test certificate, with N1 is the highest level) checked and corrected errors of outputs.

[3]http://www.jogmec.go.jp/news/bid/search.php

[4]https://github.com/jsvine/pdfplumber

The annotation was internally conducted with two annotators in two steps. In the first step, each annotator was assigned a set of documents. With each document, the annotator read predefined tags and assigned start and end positions for corresponding segments. The second step is cross-validation, in which documents were cross-checked and corrected based on the negotiation of the annotators. The agreement computed by Cohen Kappa[5] of two annotators is 0.8275 (before correction), showing that the annotators have a high agreement in annotating data.

3.2.2 Bidding and sale documents

To have a better assessment of IE models, we prepared two other datasets used internally in our company. The first contains bidding documents in different domains compared to the CinData. The second includes sale documents of hardware devices. Due to the policy, we can not disclose these datasets.

3.2.3 Data observation

Table 1 shows statistics of the three datasets. As observed, the number of training samples is small. It supports the point that in business cases, having a large number of training data is a big obstacle. In this sense, we also simulated our dataset with limited training samples. The documents are quite long, with a quite large number of sentences and characters per sample. A large number of entity types, e.g. 19 or 24 also challenges IE models.

Statistics	CinData	Bidding docs	Sale docs
#training docs	82	78	300
#dev docs	22	-	-
#testing	20	22	165
#chars/doc	3,030	22,537	2,083
#sentss/doc	120	616	56
# of tags	19	24	8

Table 1: Data observation on three datasets.

4 Extraction with Transformers

This section introduces the IE models based on transformers. We first describe transformers and then show transfer learning, information extraction, and the training process of the models.

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

4.1 Transformers

The Transformer is the first transduction model relying entirely on self-attention to compute representations of its input and output without using sequence-aligned RNNs or convolution (quoted from (Vaswani et al., 2017)). The Transformer complies with the overall architecture of encoder-decoder using stacked self-attention and point-wise, fully connected layers. The attention function of the transformer is computed by mapping a query and a set of key-value pairs to an output. Then, the output is computed as a weighted sum of the values, where the weight of each value is computed by a compatibility function of the query with the correlated key.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

where dimension d_k of keys, and dimension d_v of values Moreover, Transformer performs the attention function in parallel, resulting d_v-dimensional output values using "multi-head attention" as following: $MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$ where $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$

This paper investigates the IE task with four transformer-based models: BERT, ALBERT, DistilBERT, and ELECTRA. We selected BERT because it pioneers the transformer direction (Devlin et al., 2019), after that, its variation also achieves promising results. For ELECTRA, it is up-to-date architecture that obtains improvements compared to the BERT family (Clark et al., 2019).

BERT BERT, introduced by (Devlin et al., 2019), was the state-of-the-art model for many benchmark datasets in multiple NLP tasks. It utilized the bidirectional pre-training to represent a language as dense and low-dimensional vectors. The model is pre-trained using two unsupervised tasks, namely masked language modeling and next sentence prediction. In Masked Language Modeling, 15% of all WorkPiece tokens in a sequence are either (i) replaced with a [MASK] token, or (ii) replaced with a random token, or (iii) remained the same. By learning to predict the masked tokens, the model learns the representation of tokens in association with the context surrounding it. In Next Sentence Prediction, the model learns the relationships between two sentences by predicting whether sentence B follows sentence A in a sequence.

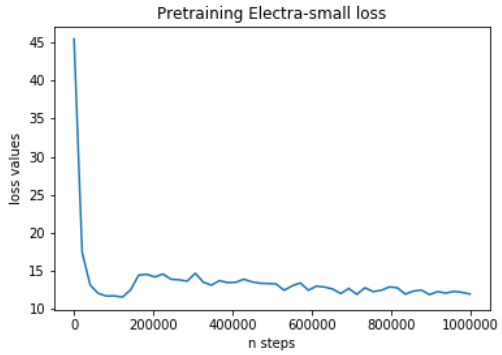
DistilBERT DistilBERT leverages knowledge distillation, in which a compact model - the student - is trained to reproduce the performance of a large model - the teacher (Sanh et al., 2019). Following this setting, the student - DistilBERT, which has the same architecture as BERT but fewer layers learn to perform pre-trained tasks by mimicking the output distribution of the teacher network - the original BERT model. The model uses the triple loss, which combines the losses of the masked language model, distillation, and cosine-distance. It also follows the practice of previous variations of BERT-based models by using dynamic masking and omitting the next sentence prediction objective.

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. In addition to the aforementioned techniques, ALBERT also employs an inter-sentence coherent loss in the replacement of the next sentence prediction task during the pre-training process.

ELECTRA ELECTRA is a replaced token detection method that trains a discriminative model predicting whether each token in the corrupted input could be replaced by a generator sample (Clark et al., 2019). Compared to BERT and its variations, ELECTRA makes two important differences. First, instead of training a `[MASK]` language model trained on the small subset that was masked, ELECTRA trains a language model on all input tokens. Second, ELECTRA was trained in a discriminative fashion to predict whether each token in the cor-

rupted input was replaced by a generator sample or not rather than predicting the original identities of the corrupted tokens. In addition, ELECTRA makes an important consideration for pre-training methods that should be efficiently computed without large amounts of data.

We employed the success of ELECTRA (Clark et al., 2019) to our IE task. Since ELECTRA is only for non-Japanese languages, we trained the Japanese ELECTRA model for our purpose. To do that, we collected Japanese Wiki-data then used the code of ELECTRA[6] for training an ELECTRA-small model. The difference compared to the original model is that we used SentencePiece instead of WordPiece because it is hard to apply word segmentation to Japanese. The idea of SentencePiece[7] bases on subword units and unigram language model, which help us to train our ELECTRA without any language-specific pre- and post-processing. More importantly, SentencePiece allows our ELECTRA to extend the vocabulary which is benificial for the training process. The size of our vocabulary for Japanese-wiki is 32,000. The pre-training task of Electra-small took 6 days with 1M steps by using a single GPU Radeon VII 16GB. The following figure shows the loss during the training process.



After training, we applied the model to the datasets. The idea is similar to BERT-QA, in which we fed hidden representation from ELECTRA to an MLP for classification.

4.2 Transfer learning

Transformers provide an appropriate solution for data representation by using contextual embeddings learned from a large amount of data. However, they should be adapted to downstream tasks by using training data in specific domains. To do that, we fine-tuned the models to the downstream IE task by using the samples data of each dataset. The pre-trained weights of transformers were first reused and then adjusted in the fine-tuning process.

4.3 Information extraction

Output vectors from the transfer learning layer were put into the extraction layer for extracting information. To do that, the extraction was formulated as a question answering (QA) task, thanks to the suggestion of BERT (Devlin et al., 2019). A question (tag) and corresponding segment were fed into transformers to learn hidden representation. The extraction predicts start and positions based on the probability of the word i in this span. The final score of a potential answer spanned from position i to position j defined as $\max_{i,j}(S\Delta T_i + E\Delta T_j)$ with $j \geq i$.

$$P_{start_i} = \frac{e^{S.T_i}}{\sum e^{S.T_j}}; \qquad P_{end_i} = \frac{e^{E.T_i}}{\sum e^{E.T_j}} \quad (2)$$

The extraction uses the positions $start$ and end to extract information corresponding to input tags.

4.4 Training

We used a multilingual BERT-base model trained for 102 languages (including Japanese) on a huge amount of texts from Wikipedia (Devlin et al., 2019). The BERT model has 12 layers, a hidden layer of 430, 768 neurons, 12 heads. For DistillBERT, we used a multilingual model pretrained with the supervision of BERT-base-multilingual-cased on the concatenation of Wikipedia in 104 different languages. The model has 6 layers, 768 dimensions, and 12 heads. For ALBERT,[8] we used the pre-trained Japanese model with 12 layers, the hidden size of 768, and the embedding size of 128. For ELECTRA, we used our pre-trained model trained on Japanese Wiki data. The ELECTRA-small has 12 layers, with a hidden size of 256.

Thanks to the suggestion of BERT, we formulated the training process as a QA task. Tags and corresponding segments were fed into the models for

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

[7]https://github.com/google/sentencepiece#comparisons-with-other-implementations

[8]https://huggingface.co/ALINEAR/albert-japanese-v2

learning. The training was done in two steps: pre-training and fine-tuning. For the first step, the pre-trained weights of transformers were reused, while the weights of the rest layers were generated with a truncated normal distribution. All models were fine-tuned in 20 epochs by using the cross-entropy loss function between predicted and correct information. The training process was done with a single GPU.

5 Settings and Evaluation Metrics

Settings We used training samples in Table 1 for training IE models and applied the model on the test sets. Due to our investigation purpose, we did not fine-tune IE models by using the development set of CinData. Instead of doing that, we report the performance on this set. For transformers, Table 2 summarizes its information. All models were trained by using the same data segmentation, settings, and GPU.

Model	Layers	Parameters
BERT-base	12	110M
DistillBERT	6	66M
ALBERT	12	12M
ELECTRA	12	14M

Table 2: Information of transformers.

As observed, BERT and DistillBERT have a large number of parameters while ALBERT and ELECTRA are significantly compressed.

Evaluation metrics Extracted information was matched with correct answers for computing F-scores based on precision and recall metrics. The F-score of a model on a dataset is the average of F-scores on all tags computed by fields.

6 Results and Discussion

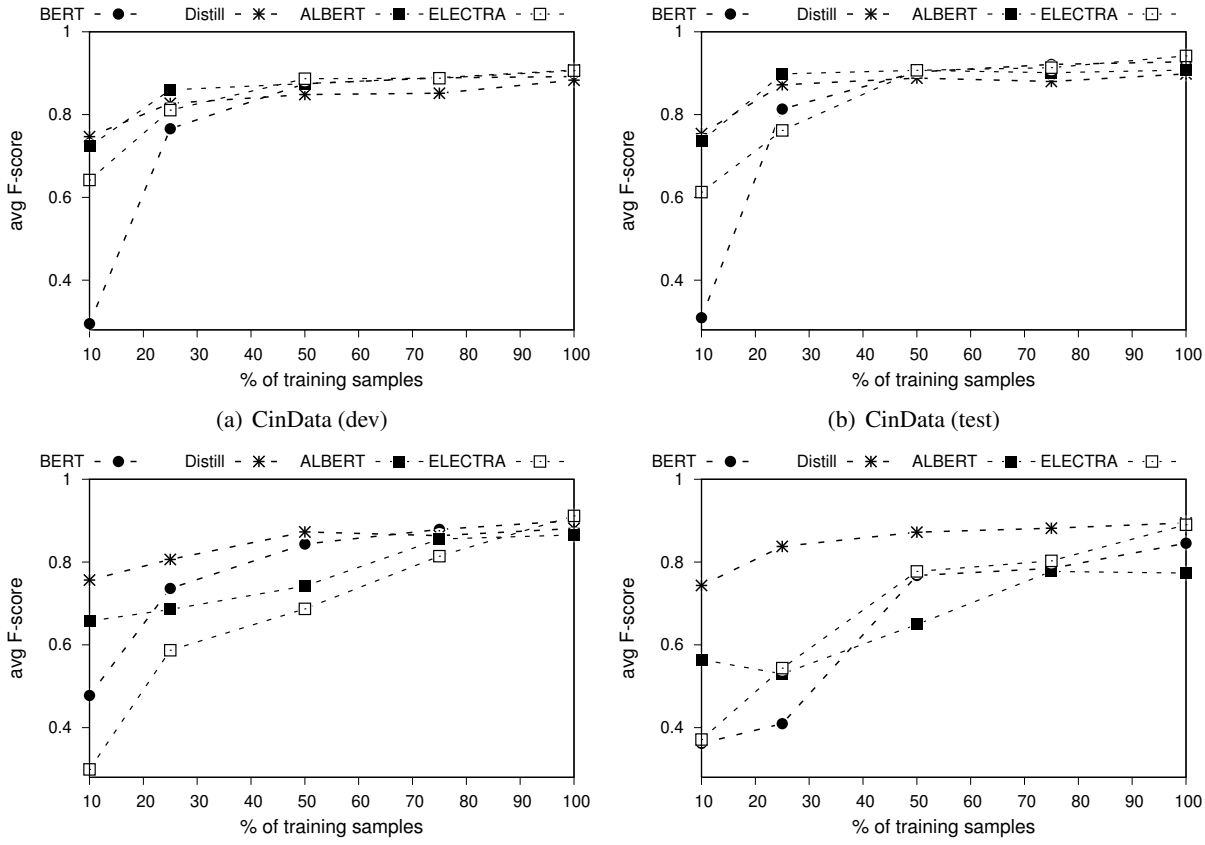
6.1 F-scores Comparison

Table 3 summarizes the comparison of transformer-based IE models on four datasets. As we can observe that the IE models based on BERT and ELECTRA achieve promising results. For example, the model of BERT is the best in two cases (CinData (dev) and CinData(test)) and ELECTRA obtains the highest F-score on the bidding dataset. For BERT, it is understandable that it has the largest model which includes 110M parameters. This enables BERT to capture the context of words from the input (the relationship between a tag-segment pair). As a result, the IE model using BERT achieves promising results. An interesting point comes from ELECTRA. It is a small model with 14M parameters, compared to BERT (110M) and DistilBERT (66M); however, the ELECTRA-based IE model outputs competitive F-scores on four datasets. For example, the IE model using ELECTRA is better than BERT of 1.15 F-score on bidding documents (0.9115 vs. 0.9000), which is the most challenging dataset with very long documents. The possible reason comes from the training process of ELECTRA that can contribute to the ELECTRA-based IE model. As mentioned in Section 4.1, we used SentencePiece instead of WordPiece due to the word segmentation of Japanese. This is different from BERT, DistilBERT, and ALBERT which used WordPiece for Japanese. The promising F-scores of ELECTRA with a small pre-trained model draw a new direction for adapting transformers to our IE task and confirm the results of ELECTRA (Clark et al., 2019).

Method	CinData (dev)	CinData (test)
BERT (QA)	**0.8887**	**0.9175**
DistilBERT	0.8831	0.8983
ALBERT	0.8585	0.8926
ELECTRA	*0.8879*	*0.9133*
Method	**Bidding docs**	**Sale docs**
BERT (QA)	0.9000	0.8456
DistilBERT	0.8811	**0.8944**
ALBERT	0.8655	0.7734
ELECTRA	**0.9115**	*0.8901*

Table 3: Comparison of methods according the average of F1-score. **Bold** is the best and *italic* is the second best.

The extension of BERT does not show the best performance on four datasets. For example, DistilBERT is only the best on sale documents with tiny margins compared to other models, even it is the second larger model (66M parameters). It is understandable that DistilBERT tries to compress the model size while approximating the performance with BERT. In other cases, DistilBERT and ALBERT output lower F-scores than BERT and ELECTRA. A possible reason comes from the size of the model. For instance, ALBERT obtains the lowest F-



(a) CinData (dev)

(b) CinData (test)

(c) Bidding documents

(d) Sale documents

Figure 1: F-scores and the number of training examples.

scores in all cases, e.g. 0.7734 on sale documents because it only has 12M parameters, which are hard to cover all the semantic aspects of the datasets.

6.2 F-scores and Training Samples

We observed the behavior of transformers regarding the number of training examples. This is because we would like to understand when the transformers can achieve good results. To do that, we randomly segmented data into several parts, ranging from 10%, 20%, 50%, 75%, and 100% and observed the F-scores at each data segment. Figure 1 visualizes the observation of BERT, ALBERT, DistillBERT, and ELECTRA wit different data segments.

As we can observe, the number of training samples affects the quality of transformer-based IE models. The general trend shows that adding more training examples increases F-scores. However, the behavior of transformers is different. For example, on CinData, F-scores significantly raise from 10% to 25% of training data and reach the top at 50% of training data. After that, the F-scores slightly grow. This indicates that for CinData, transformers only need 50% of data to obtain stable performance. For biddings and sales, the trend is quite different. For biddings in Figure 1(c), two strong models (BERT and ELECTRA) share the similar behavior, in which its F-scores dramatically increase from 10% to 75%. After that, the F-scores are stable. It is explainable that adding more data helps to improve the quality of BERT and ELECTRA-based IE models. In contrast, DistilBERT and ALBERT have the same trend, in which these models obtain quite high results at 10% and steadily raise until 75%. The trend on sale documents in Figure 1(d) is quite diverse, in which the behavior of DistilBERT is the same on bidding and sale documents. BERT and ELECTRA have significant improvements from 10% to 50% while ALBERT reaches the top at 75%. It is interesting to observe that by using a small number of data, Distil-

BERT seems to be better than others on bidding and sale documents in Figures 1(c) and 1(d). This suggests two use cases: (i) if we only have some dozens of data, e.g. 50-100 samples, DistilBERT can be a good option and (ii) otherwise, BERT and ELECTRA are appropriate the selection.

6.3 Training Time

We observed the training time of transformers with the same data segmentation of Section 6.2. Figures 2, 3, and 4 plots the observation.

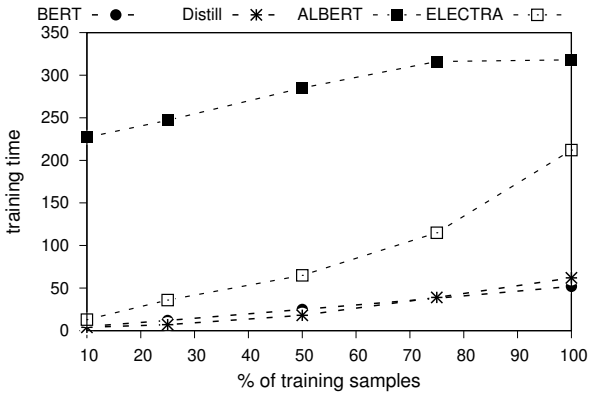


Figure 2: Training time (minutes) on CinData.

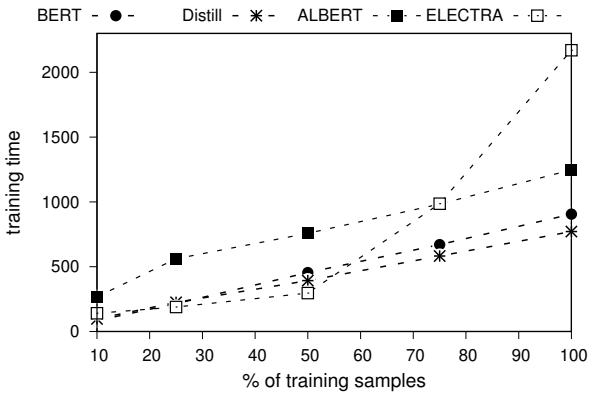


Figure 3: Training time (minutes) on bidding documents.

It is interesting to observe that BERT and DistilBERT are the fastest even they have the largest models with a huge of parameters. A possible reason is that with a large number of parameters, these models do not need to learn so much from the data of new domains. As a result, they are quick to be covered in the training process. In contrast, ALBERT and ELECTRA take a long time to complete the training process. For example, ALBERT needs 300 minutes

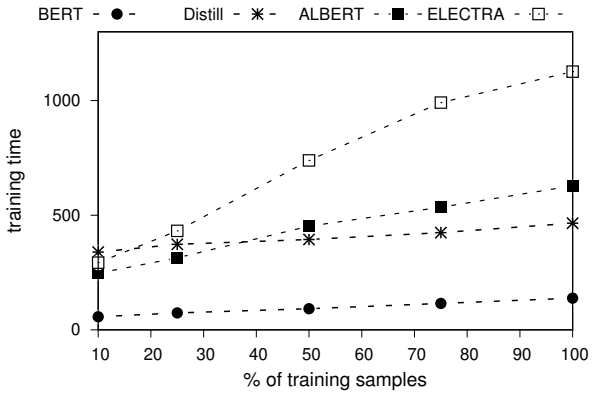


Figure 4: Training time (minutes) on sale documents.

for 100% data on CinData. Also, the small number of parameters seems to be efficient for inference only. For training, the computation operation is not so much different among the four transformers. As a result, ALBERT and ELECTRA took a longer time than BERT and DistilBERT for training.

7 Conclusion

This paper introduces an investigation of transformers for information extraction with limited data. The investigation simulates business scenarios that have small numbers of training data to build IE models. To do that, we employ four well-known transformers for taking advantage of the contextual aspect learned on huge data and fine-tune to our down-stream IE tasks by using transfer learning. Experimental results on three domain-specific business datasets confirm the efficiency of BERT and ELECTRA, that can be applied to actual business cases. The observation of training samples indicates that in some cases, transformers can achieve good results with 50% of training data. The training time shows that BERT is potential while ALBERT and ELECTRA take a long time when training with all data.

For future direction, we encourage to deeply investigate sophisticated models for the IE task, e.g. stacking transformers with refined architecture.

Acknowledgments

We would like to thank Gaku Fujii, Shahab Sabahi, Akira Shojiguchi, and reviewers for useful comments and discussion. This research is funded by Hung Yen University of Technology and Education under the grant number UTEHY.L.2020.04.

References

Gabor Angeli, Melvin Johnson Premkumar, and Christopher D. Manning. 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), pp. 344-354*.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *Proceedings of International Conference on Learning Representations*.

Luciano Del Corro and Rainer Gemulla. 2013. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web, pp. 355-366*.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 868-878*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186*.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, Volume 1-Volume 1, pp. 141-150. Association for Computational Linguistics.*

Michael Fleischman and Eduard Hovy. 2002. Fine grained classification of named entities. In *Proceedings of the 19th International Conference on Computational Linguistic, Volume 1, pp. 1-7. Association for Computational Linguistics.*

Lindsay Herbert. 2017. Digital transformation: Build your organization's future for the innovation age. Technical report, Bloomsbury Publishing.

Bill Inmon and Anthony Nesavich. 2007. *Tapping into Unstructured Data: Integrating Unstructured Data and Textual Analytics into Business Intelligence*. Pearson Education.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 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*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of The International Conference on Learning Representations*.

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium, pp. 581-587. Springer, Berlin, Heidelberg*.

Jerry Chun-Wei Lin, Yinan Shao, Yujie Zhou, Matin Pirouz, and Hsing-Chung Chen. 2019. A bi-lstm mention hypergraph model with encoding schema for mention extraction. *Engineering Applications of Artificial Intelligence 85: 175-181*.

Minh-Tien Nguyen, Viet-Anh Phan, Le Thai Linh, Nguyen Hong Son, Le Tien Dung, Miku Hirano, and Hajime Hotta. 2019. Transfer learning for information extraction with limited data. In *Proceedings of 16th International Conference of the Pacific Association for Computational Linguistics, pp. 469-482*.

Erik Sang, Tjong Kim, and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. In *arXiv preprint arXiv:1910.01108*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 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*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems, pp. 6000–6010*.

Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 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, pp. 649-657*.

Appendix A. The Name of Tags

English Tag	Type
Year of Procurement	datetime (year only)
Prefecture	text
Bid Subject	text
Facility Name	text
Address for Demand	text
Start Date of Procurement	datetime
End Date of Procurement	datetime
Public Announcement Date	datetime
Deadline for Questionnaire	datetime
Deadline for Applying Qualification	datetime
Deadline for Bidding	datetime
Opening Application Date	datetime
PIC for Inquiry of Questions	text
TEL/FAX for Inquiry of Questions	text
Address for Submitting Application	text
Department/PIC for Submitting Application	text
Address for Submitting Bid	text
Department/PIC for Submitting Bid	text
Place of Opening Bid	text

Table 4: Extracted information of CinData.

English Tag	Type
Model code	mixed
Model name	mixed
Start of sales	date
End of sales (planed)	date
End of sales (fixed)	date
End of sales (special)	date
End of support	date
Revision	mixed

Table 5: Extracted information of sale documents.

English Tag	Type
Year of procurement	datetime (year only)
Prefecture	text
Title of bidding	text
Name of institution	text
Address for demand	text
Start date of procurement	datetime
End date of procurement	datetime
Contract value	number
Amount of value	number
Class of reserved value	number
Amount of reserved value	number
Public Announcement Date	datetime
Deadline for delivery specification	date
Deadline for questionnaire	datetime
Deadline for applying qualification	datetime
Deadline for bidding	datetime
Opening application date	datetime
PIC for inquiry of questions	text
TEL/FAX foriInquiry of questions	tel/fax
Address for submitting application of qualification	address
Address of submitting of bidding applications	address
Department/PIC for submitting application	name
Place of Opening Bid	text

Table 6: Extracted information of bidding documents.