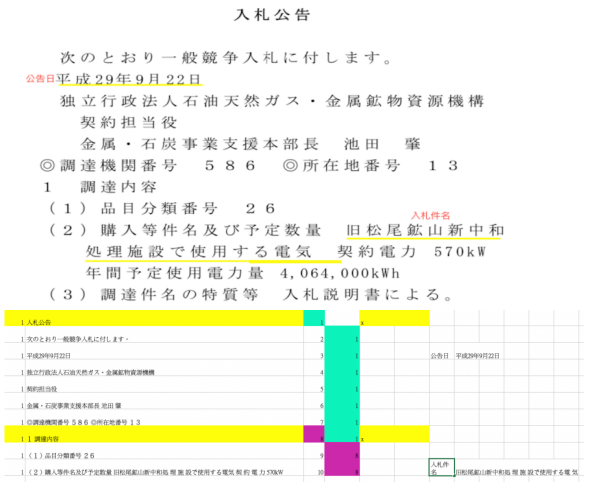
Document Information Extraction

Kuan-Lin Wu*†
R08922115
National Taiwan University
Taipei, Taiwan

Shung-Cyuan Hong
R08944034
National Taiwan University
Taipei, Taiwan

Yi-Kai Lu
R08944035
National Taiwan University
Taipei, Taiwan



入札公告

次のとおり一般競争入札に付します。

公告日 平成29年9月22日

独立行政法人石油天然ガス・金属鉱物資源機構
契約担当役
金属・石炭事業支援本部長　池田　肇

◎調達機関番号　５８６　◎所在地番号　１３

１　調達内容

（１）品目分類番号　２６

（２）購入等件名及び予定数量　旧松尾鉱山新中和処理施設で使用する電気　契約電力　570kW
年間予定使用電力量　4,064,000kWh

（３）調達件名の特質等　入札説明書による。

入札件名

Figure 1. Original document, including PDF file and excel file

No.	Tag Name (JP)	Tag Name (EN)	Value Type	Description
1	調達年度	Year of Procurement	datetime (year only)	
2	都道府県	Prefecture	text	
3	入札件名	Bid Subject	text	
4	施設名	Facility Name	text	
5	需要場所(住所)	Address for Demand	text	
6	調達開始日	Start Date of Procurement	datetime	
7	調達終了日	End Date of Procurement	datetime	
8	公告日	Public Announcement Date	datetime	
9	仕様書交付期限	Deadline for Delivery of the Specification	datetime	
10	質問票締切日時	Deadline for Questionnaire	datetime	
11	資格申請締切日時	Deadline for Applying Qualification	datetime	
12	入札書締切日時	Deadline for Bidding	datetime	
13	開札日時	Opening Application Date	datetime	
14	質問箇所 所属/担当者	PIC for Inquiry of Questions	text	
15	質問箇所 TEL/FAX	TEL/FAX for Inquiry of Questions	text	
16	資格申請送付先	Address for Submitting Application	text	
17	資格申請送付先 部署/担当者名	Department/PIC for Submitting Application	text	
18	入札書送付先	Address for Submitting Bid	text	
19	入札書送付先 部署/担当者名	Department/PIC for Submitting Bid	text	
20	開札場所	Place of Opening Bid	text	

Figure 2. The list of pre-defined tags for information extraction

Abstract

In many computer science regions, transfer learning is usually a hot topic because of insufficient data. If we simply use these less training data, we might get the result which is not as expected. Transfer learning focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. If we can make good use of transfer learning, it can help us to solve the problem of insufficient data easily.

Keywords: National language processing, Information extraction, BERT, fine-tune

ACM Reference Format:
Kuan-Lin Wu, Shung-Cyuan Hong, and Yi-Kai Lu. 2020. Document Information Extraction.

*Both authors contributed equally to this research.
†Github: https://github.com/zaq851017/adl_final.git

1 Introduction

In this task, we only have 82 training data and 22 development data. The task what we need do is information extraction. We need to extract each document 20 correspond tags, and a document is composed of a set of pdf and excel file. The Figure 1 is an easy sample, and the Figure 2 is 20 correspond tags. BERT has achieved state-of-the-art performance on a number of natural language understanding tasks. As a result, our team use BERT as our pre-trained model; then, we use little training data to fine-tune BERT model. We will introduce our steps to preprocess data; then, we will compare three models in this task. We consider models which including not only characters but also words. We believe that the meaning of words could help us to predict the tags. Finally, we will display our experiments and model parameters.

2 Approach

2.1 Input representation

The dataset is composed of Japanese documents which are provided by a Japanese company. Due to the input sequence length limitation of BERT, we have to split each document in multiple texts. From our perspective, each document can

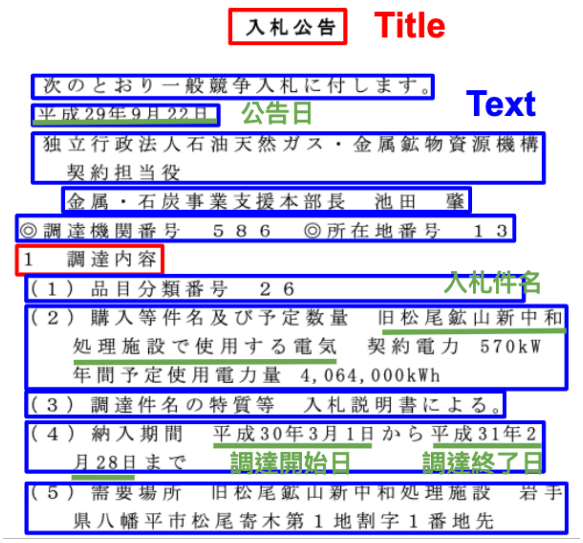


Figure 3. Segmentation of original document

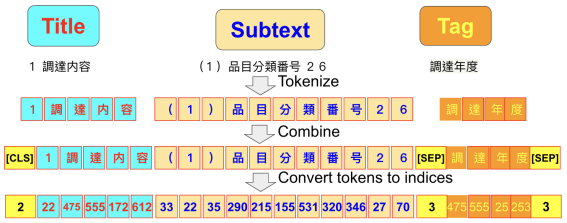


Figure 4. The steps of character tokenization

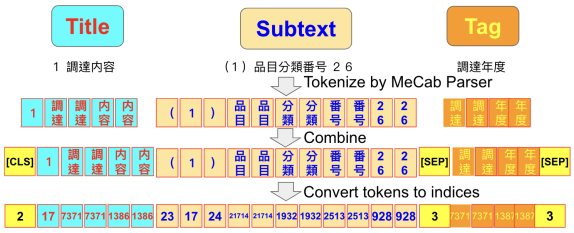


Figure 5. The steps of word tokenization

be seen as consisting of titles and texts, so we find out the index of each title in the document, and then use titles to split the whole document in order to get each title and its subtext. Besides, since the index of immediate title of each line is provided, we utilize them to split the subtext of each title, as Figure 3.

By doing so, the length of most subtexts are short enough to being the input of BERT. However, we found that the title is so significant that may help our model to learn the meaning of text, since most tags are related to the title. Therefore, we concatenate the title and its subtext as our one of inpnut sequences.

Since we treat the task as a question-answering system, we not only need to give the preprocessed text to be one of our input sequence but also its tag as the other input sequence. As a result, preprocessed text and tag are concatenated, and the SEP token is inserted between text and tag in order to separate a pair of subtext and tag. Besides, CLS is inserted at the first of the input sequence to be an representation for classification task, and SEP is inserted at the end of the input sequence to be an representation of sequence ending.

After dealing with the input sequence, the next thing is tokenization. We tokenize the input by two methods. As Figure 4 shows, the first method is character tokenization, simply split each character of input to form tokens. As Figure 5 shows, the second method is word tokenization. To split the words in input sequence, we use a Japanese parser, MeCab, which be able to parse the word in sentence. By MeCab, we can parse every word in input, and we replace each character with its word as a token.

In order to convert input tokens to embedding index of BERT, we use BERT-based Japanese tokenizers, including character tokenizer and word tokenizer. By the tokenizers, we can convert input tokens to their character embedding and word embedding respectively, to be the inputs of our BERT model.

2.2 BERT (character only)

Figure 6 is the basic BERT-QA approach. We import the pre-trained BERT-based Japanese char model with whole word masking, along with three simple linear layers to transfer the hidden state and outputs to answerable, start and end.

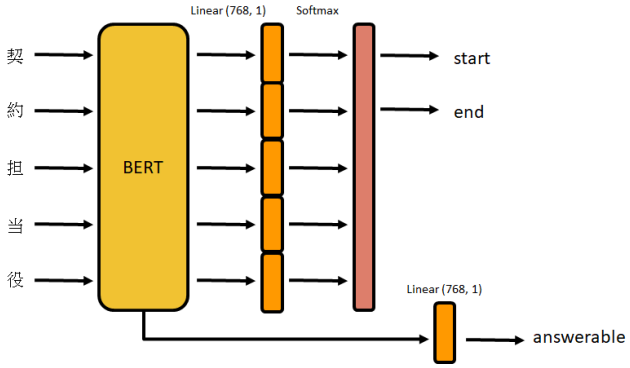


Figure 6. Architecture of BERT-based character model

2.3 BERT + CNN

The baseline model has a disadvantage. For the BERT model based on characters, each input is independent, which means that the model has no concept of words.

In order to improve this shortcoming, we changed the last linear layer to Conv1d layer, so that the model would select multiple characters at once according to the CNN kernel size.

Therefore, the kernel size can be regarded as the expected word length. According to the general understanding of language, we used a kernel of size [5,4,3,2,1] at a time, which means that we expect these lengths of words to appear in the article. It should be noted that when the kernel size=1, it is the same as linear.

As shown in Figure 7, Since Conv1d with different kernel sizes will calculate results with different lengths, we will stack the results with different lengths according to the index, and then use softmax to predict the start and end.

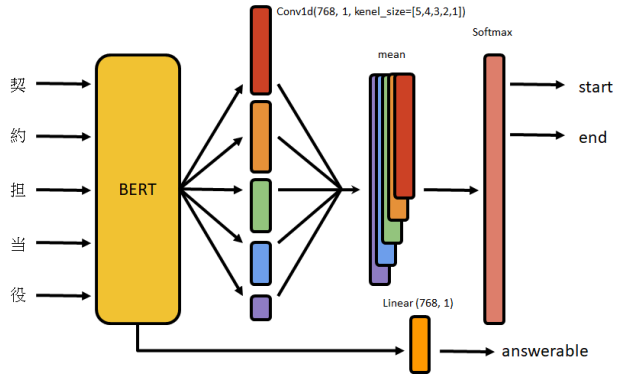


Figure 7. Architecture of BERT + CNN model

2.4 BERT (character + word)

Obviously, using CNN to guess the length of each word is not robust. Since we really want the model to see the words, it would be more straightforward to directly use word-based-BERT.

However, word-based-BERT cannot exist alone. According to [1], the characteristics of Japanese are different from that of English, and it is inconsistent to cut sentences directly with words.

An example is given in [1]. Assuming the sentence is "Tokyotonai", word tokenizer will cut it into "Tokyo" and "tonai". But if the answer is "Tokyoto", this model can never predict the correct answer. Therefore, word-based-BERT and char-based-BERT must exist together.

With this concept, we made a simple adjustment to the baseline model. As Figure 8, in addition to the original char-based-BERT, a parallel word-based-BERT is used, and the output of both is concatenated to make predictions.

3 Experiments

Some details of the experiment are provided in following. All of the Experiments are train in training set, test in development set. Each text has 20 questions, so a text would become 20 samples, which are connected to different questions. Using threshold=0.5 to predict whether the sample is answerable, and choose the top1 probability as start/end

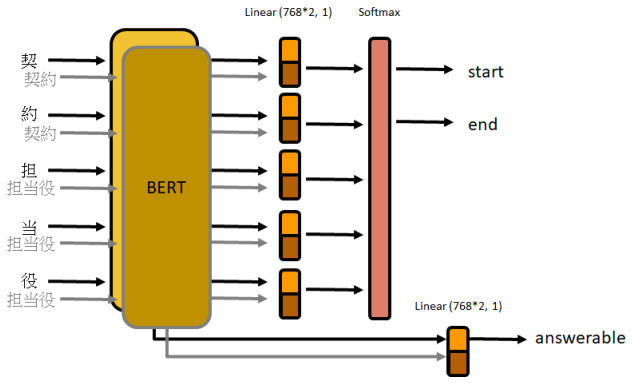


Figure 8. Architecture of BERT-based character + word model

Table 1. Performance of each model

Method	Precision	Recall	F1-score
BERT (char only)	96.08	96.01	95.82
BERT + CNN	96.34	96.38	96.16
BERT (char + word)	96.11	96.51	96.10

index. In addition, Using AdamW (lr=0.0001) as optimizer, cross entropy and binary cross entropy as cost function.

As shown in Table 1, the two models after the change have improved relative to the baseline, but the three methods have little difference in the competition.

The concept of improving by word segmentation is correct, but the latter two methods have some places that are not very make sense. If we can improve on these points, progress can be expected:

BERT+CNN

- It may not be correct to stack the results of different kernels according to the index. Because the meaning of the same index is not the same, it seems to be biased to directly average.
- It is not a good to use the kernel size to guess the length of the word, and kernels of different lengths should also have a weighted relationship with each other.
- Perhaps the output of the last few layers of bert can be extracted at once, so that Conv2D can be implement, and maybe there will be better performance.

BERT (char + word)

- There are only half-shaped words in the pre-training model, but all and half-shaped words appear in the data, resulting in a lot of UNK for word-bert
- The words in word-bert will appear repeatedly according to their length, but this is not the logic of

normal human writing and may cause obstacles to model discrimination.

In addition, it is observed that most of the start/end in the sample will appear at the head and tail, plus the F1 score takes Recall score into account, we believe that the prediction of answerable is crucial. Because the characteristics of the whole sentence are often selected, the start/end of this task is relatively well predicted, so we guess that if the model answers as much as possible instead of giving up, it may get a higher score.

Therefore, we want to observe the relationship between F1 score and Threshold to verify our conjecture. As Figure 9, the result shows such a trend that F1 score and Threshold shows negative correlation.

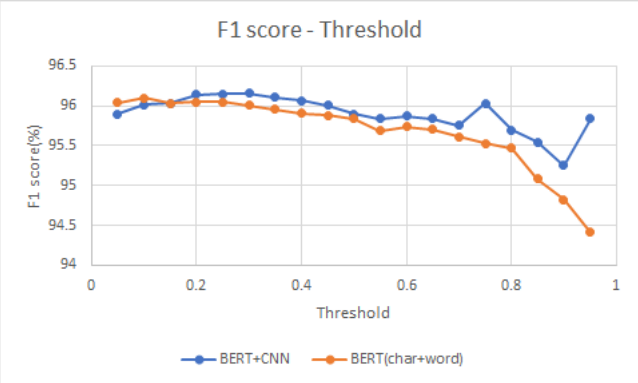


Figure 9. Correlation of F1 score and Threshold

4 Work Distribution

- Yi-Kai Lu: Preprocess
- Shung-Cyuan Hong: Train and test
- Kuan-Lin Wu: Train and test

5 Conclusion

We present a good example of transfer learning,and achieving a productive result despite having insufficient training data. The BERT model has a powerful word embedding, so we can make good use of BERT as pretrained model without sufficient train data. We believe the word understanding is the most important key to complete this task, and BERT is the state of the art on word embedding. As the result, the future work could be focused on the method of preprocessing data. In the future, transfer learning will be future trend because many AI application only have a little training data.

Acknowledgments

We thank Prof. Yun-Nung Chen for teaching Applied Deep Learning course and all the TAs of the course for homework assistance.

References

[1] Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma. 2017. Character-based Bidirectional LSTM-CRF with words and characters for Japanese Named Entity Recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*. Association for Computational Linguistics, Copenhagen, Denmark, 97–102. https://doi.org/10.18653/v1/W17-4114

[2] Minh-Tien Nguyen, Viet-Anh Phan, Le Thai Linh, Nguyen Hong Son, Le Tien Dung, Miku Hirano, and Hajime Hotta. 2020. Transfer Learning for Information Extraction with Limited Data. arXiv:2003.03064 [cs.IR]

[1] [2]