LIAT Team’s Wikipedia Classifier at NTCIR-15 SHINRA2020-ML: Classification Task

Kouta Nakayama
RIKEN AIP, Japan
kouta.nakayama@riken.jp

Satoshi Sekine
RIKEN AIP, Japan
satoshi.sekine@riken.jp

ABSTRACT
This paper reports the document classification system that our team LIAT submitted to the classification task in NTCIR-15 SHINRA2020-ML[2]. We used the outputs of BERT[1] as document embeddings to deal with the longer sentences of Wikipedia. We used the Transformer[3] encoder to classify the document embeddings into each class. Our system was not better than other submission results, but we hope that our results will also be used as a resource.

TEAM NAME
LIAT

SUBTASKS
SHINRA2020-ML: Classification Task

1 INTRODUCTION
SHINRA2020-ML[2] is a shared task to classify Wikipedia in 30 languages into Extended Named Entity (ENE) Hierarchy (ENEH). This task employs version 8.0 of ENEH and classification into 221 classes. We participated in all 30 languages targeted in this task. In this paper, we describe in detail the system we used for classification.

In recent years, pre-trained language models, such as BERT[1], have been utilized for document classification. However, BERT and other transformer-based models can generally only handle around 500 tokens at once due to memory limitations. Therefore, we exploit the outputs of BERT as document embeddings and classify the embeddings into each class using the encoder of the Transformer. This approach allows us to handle input tokens longer than the limit of BERT.

2 MODEL
2.1 BERT for Document Embedding
We fine-tune BERT on the classification task to obtain task-specific document embedding. Specifically, the documents to be classified are divided into a number of tokens that BERT can handle, and they are classified using BERT. In general, when classifying with BERT, the special token [CLS] is combined with the input, and the output for the special token is the classification result, as shown in Figure 1. Here, we handle the intermediate output $T_{CLS}$ of fine-tuned BERT for [CLS] as a task-specific document embedding.

2.2 Transformer Encoder for Document Classification
We classify the document embeddings obtained by BERT using the encoder of Transformer, as shown in Figure 2. During training, document embeddings are fixed. The encoder, like BERT, uses the output corresponding to [CLS] to classify the document. The embedding assigned to [CLS] is generated using a dedicated embedding layer and is updated during training.

3 EXPERIMENTS
3.1 Model Details
We use pre-trained BERT-Base in 104 languages\footnote{We used the cased model. https://github.com/google-research/bert/blob/master/multilingual.md}. Also, we use the Transformers library[4] to build our models. The hyperparameters...
we used to train BERT and Transformer encoder are shown in Table 1 and Table 2, respectively. We used the same values as in Table 1 for hyperparameters not mentioned in Table 2.

### 3.2 Submission Results

We show the official results of the SHINRA2020-ML in Table 3. All scores are macro average F1 measure. Late submission means reference result submitted after the deadline. The results of our system seem to be inferior in all languages to the results of the best system, such as FPTAI and uomfj. The difference between our system and the best system is shown in Table 4. We seem to have a very low score in hi for our system. Since we did not conduct any hyperparameter search, we consider the training of the model to be converging to a local minimum. In future research, we will monitor the development data score during training to prevent learning failure, such as this one. Our system seems to score particularly poorly in minor languages. We may need to conduct a hyperparameter search, as the learning accuracy depends more heavily on the hyperparameters the less data we have. In future research, we will be searching for hyperparameters of learning as far as our computational resources will enable.

### 4 CONCLUSIONS

This paper describes the our system submitted to SHINRA2020-ML. We did not achieve a higher score than other systems. However, the purpose of SHINRA2020-ML is collaborative resource construction, and our results will also be used for ensembles and other purposes. In future work we will adjust our training in more detail, such as the exploring of hyperparameters.

### REFERENCES


