Cross-lingual Extended Named Entity Classification of Wikipedia Articles

The Viet Bui
FPT Technology Research Institute
FPT University, Vietnam
vietbt6@fpt.com.vn

Phuong Le-Hong
Vietnam National University, Hanoi, Vietnam
FPT Technology Research Institute
phuonglh@vnu.edu.vn

ABSTRACT

The FPT.AI team participated in the SHINRA2020-ML subtask of the NTCIR-15 SHINRA task. This paper describes our method to solving the problem and discusses the official results. Our method focuses on learning cross-lingual representations, both on the word level and document level for page classification. We propose a three-stage approach including multilingual model pre-training, monolingual model fine-tuning and cross-lingual voting. Our system is able to achieve the best scores for 25 out of 30 languages; and its accuracy gaps to the best performing systems of the other five languages are relatively small.

TEAM NAME

FPT.AI

SUBTASK

SHINRA2020-ML Shared Task

1 INTRODUCTION

SHINRA is a project to structure Wikipedia based on a pre-defined set of attributes for given categories. The categories and the attributes follow the definition of the Extended Named Entity (ENE).[1] Within this project, a shared task called SHINRA2020-ML was proposed [4]. The FPT.AI[2] team participated in this shared task. This short paper describes our approach to solving the problem and discusses our official results.

In this shared task, we are concerned with the problem of classifying 30 language Wikipedia entities in fine-grained categories, namely 219 categories defined in ENE. The FPT.AI team selects all the 30 target languages to participate. For each language, we run our system to classify Wikipedia pages of that language and submit results for evaluation.

Our method is inspired by an emerging trend in learning general-purpose multilingual representations, which can be applicable to many tasks, including text classification. Many languages have similarities in syntax or vocabulary, and multiple learning approaches that train on multiple languages while leveraging the shared structure of the input space have begun to show good results [2]. We first develop a neural network architecture which trains a multilingual model and then fine-tune this model on language dependent datasets to obtain monolingual models. Our neural network employs multilingual BERT [1] and a special architecture for hierarchical multi-label classification, which is specifically designed for maximizing the learning capacity regarding the hierarchical structure of the labeled data [5]. Finally, we propose a cross-lingual voting technique to perform classification.

Our method obtains good results on all languages. The FPT.AI system achieves best scores for 25 out of 30 languages. For the other five languages, its performance gaps to the best performing systems are relatively small.

The remainder of this paper is structured as follows. Section 2 describes our three-stage method and the proposed neural network architecture. Section 3 presents experimental results. Section 4 concludes the paper and suggests some directions for future work.

2 METHOD

Each sample in the dataset of a target language is a Wikipedia page which contains some fields. The most important fields are *opening text*, *language*, *title* and *text*. The *text* field is the main content. In case the page does not have the *text* field, we use the *title* as the main content to perform classification.

Each category defined in ENEs has four levels, from a coarse-grained type to a fine-grained type. More precisely, the first level E_1 has 5 labels; the second level E_2 has 25 labels, the third level E_3 has 94 labels, and the fourth level E_4 has 195 labels. The last level contains the fine-grained types. This is essentially a multi-label classification problem where each page need to be assigned a subset of E_4 as its labels. Figure 2 shows the partial histogram of the most frequent labels computed on all the target languages. We see that the labels are highly imbalanced. Thus, we need to deal with an imbalanced, hierarchical multi-label classification problem.

We propose a three-stage method to tackle this problem. In the first stage, we train a multilingual model for all the 30 languages using BERT [1]. In the second stage, we fine-tune that model for each language, and train monolingual models using the same BERT architecture. Finally, in the third stage, we propose a simple voting method to perform classification. Figure 1 gives a high-level overview of the first two stages. The subsequent subsections describe these stages in detail.

2.1 Multilingual Model

Let P be an input Wikipedia page and X be its main content, which can be represented by a list of tokens $X = (x_1, x_2, \ldots, x_n)$. We need to find its output label set Y, which is a subset of $E_1 \cup E_2 \cup E_3 \cup E_4$. This subset contains all the terminal category in the ENE categories. We use SentencePiece, a language-independent subword tokenizer and detokenizer [3] to tokenize X into subwords using a vocabulary V of the BERT multilingual base cased model. The multilingual BERT model was pretrained on the top 104 languages with

[1] https://ene-project.info
[2] https://fpt.ai/

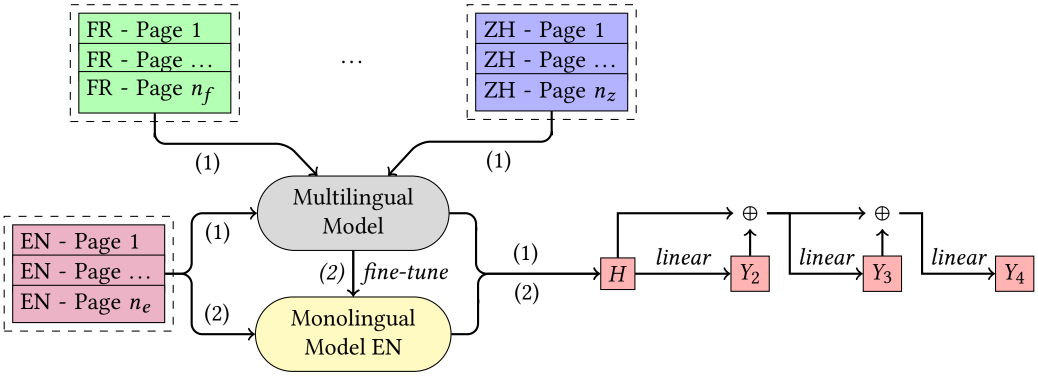

Figure 1: Cross-lingual named entity classification architecture

the largest Wikipedia using a masked language modeling (MLM) objective [1].

Let $Z = \left(d_{[\text{CLS}]}, d_{s_1}, d_{s_2}, \ldots, d_{s_t}, d_{[\text{SEP}]}\right)$ be the corresponding index sequence of S, where d_k is the index of token k in V and $t = \min\{m, l-2\}$. Here, l is a hyper-parameter specifying the max sequence length. These inputs are then fed into the multilingual BERT (mBERT) model, and for each page P, we obtain its output vector of d_h dimensions:

$$H = \text{mBERT}(Z)_{[\text{CLS}]} \in \mathbb{R}^{d_h}.$$

In order to make our model capable of labeling pages as belonging to one or multiple paths in the hierarchy, we integrate HMCN, a neural network architecture for hierarchical multi-label classification [5]. HMCN is capable of simultaneously optimizing local and global loss functions for discovering local hierarchical relationships and global information from the entire class hierarchy while penalizing hierarchical violations. This method has achieved the state-of-the-art for hierarchical multi-class classification.

In this work, we use the feed-forward (HMCN-F) architecture which is specifically designed for maximizing the learning capacity regarding the hierarchical structure of the labeled data. Let $Y_i \in \mathbb{R}^{d_{E_i}}$ is the label information of E_i, where d_{E_i} is the number of labels in E_i, with $i \geq 2$. More precisely:

$$\begin{aligned} Y_2 &= \phi(H, d_{E_2}) \\ Y_3 &= \phi(H \oplus Y_2, d_{E_3}) \\ Y_4 &= \phi(H \oplus Y_2 \oplus Y_3, d_{E_4}), \end{aligned}$$

where $\oplus$ is the concatenation operator, $\phi(x, d)$ is a linear function taking an input x and an output dimension d. We do not use E_1 because its number of labels is too small and there is little information we can learn from E_1. Let $Y_i, i = 2, 3, 4$ is the ground-truth one-hot vector corresponding to the predicted target vector $\hat{Y}_i$. For each sample, we define a loss function to be optimized as follows:

$$J(\theta) = \sum_{i=2}^{4} L_i(\hat{Y}_i, Y_i, W_i),$$

where

$$L_i = -\frac{1}{M} \sum_{j=0}^{M} w_j \left[y_j \log(\sigma(\hat{y}_j)) + (1 - y_j) \log(1 - \sigma(\hat{y}_j)) \right],$$

where $\sigma(\cdot)$ is the sigmoid function

$$\sigma(z) = 1/(1 + \exp(-z))$$

and w_j are weights which specify the importance of the j-th label in the label vector Y. The weight vector $w_i \in \mathbb{R}^{d_{E_i}}$ at level i is computed as

$$w_i = \min\left(\frac{\vec{k}_i}{\vec{c}_i}, \mathbf{1}\right).$$

Here $\vec{c}_i$ is the frequency vector of all labels in E_i, $\vec{k}_i$ is the element-wise average of $\vec{c}_i$ and $\mathbf{1}$ is the unit vector of dimension d_{E_i}.

2.2 Monolingual Models

After the first stage of training the multilingual model, in the second stage, we fine-tune that model on each language, resulting in monolingual models. The monolingual models make use of the same neural network architecture of the multilingual model as described in the previous subsection.

In the prediction step, for each page P in a given language, we compute its prediction vector at the most fine-grained level $\hat{Y}_4$ by passing the main content to the corresponding monolingual model. This vector is then passed to the sigmoid function $\hat{Y} = \sigma(\hat{Y}_4)$, which squashes $\hat{Y}_4$ into a real-valued vector whose elements are all in the $[0, 1]$ range. Finally, we pick the most probably correct labels for P by using a threshold ϵ, that is the l-th label will be assigned to P if $\hat{Y}_l \geq \epsilon$.

2.3 Voting

In this shared task, one is asked to classify each Wikipedia page into ENE types where a page may be linked across different languages. Different language-dependent pages share the same page identifier if they are linked together by the interlanguage links. Thus, they should be classified into the same hierarchical types. We leverage this important property to boost the classification accuracy by using a simple voting method as follows.

Let P be a page and all its linked pages in K other languages are $P_1, P_2, \ldots, P_K$. The monolingual models classify these pages independently and the obtained result is K lists of predicted labels. These label lists are then flattened and their frequency ratios

Table 1: Statistics of Wikipedia in 31 languages

Language	Pages	Links from JP	Ratio
English (en)	5,790,377	439.354	7.6
Spanish (es)	1,500,013	257,835	17.2
French (fr)	2,074,648	318,828	15.4
German (de)	2,262,582	274,732	12.1
Chinese (zh)	1,041,039	267,107	25.7
Russian (ru)	1,523,013	253,012	16.6
Portuguese (pt)	1,014,832	217,896	21.5
Italian (it)	1,496,975	270,295	18.1
Arabic (ar)	661,205	73,054	11.0
Japanese	1,136,222	–	–
Indonesian (id)	451,336	115,643	25.6
Turkish (tr)	321,937	111,592	34.7
Dutch (nl)	1,955,483	199,983	10.2
Polish (pl)	1,316,130	225,552	17.1
Persian (fa)	660,487	169,053	25.6
Swedish (sv)	3,759,167	180,948	4.8
Vietnamese (vi)	1,200,157	116,280	9.7
Korean (ko)	439,577	190,807	43.7
Hebrew (he)	236,984	103,137	43.5
Romanian (ro)	236,984	103,137	23.5
Norwegian (no)	501,475	135,935	27.1
Czech (cs)	420,195	135,935	25.1
Ukrainian (uk)	420,195	135,935	20.5
Hindi (hi)	129,141	30,547	23.6
Finnish (fi)	450,537	144,750	32.1
Hungarian (hu)	443,060	120,295	27.2
Danish (da)	242,523	91,811	35.6
Thai (th)	129,294	59,791	46.2
Catalan (ca)	601,473	139,032	23.1
Greek (el)	157,566	60,513	38.4
Bulgarian (bg)	248,913	89,017	35.7

$c_1, c_2, \ldots, c_k$ are counted, where k is the number of different predicted labels for this page. Finally, all labels whose frequency is above the average value will be chosen as the predicted label set for P. That is, the l-th label will be chosen if

$$c_l \geq \frac{1}{k}\sum_{j=1}^{k} c_j.$$

3 RESULTS

3.1 Datasets

The organizer of this shared task provides the training data for 30 languages, created by the categorized Japanese Wikipedia of 920K pages and Wikipedia language links for 30 languages. The training data for each target language may be a little bit noisy. For example, out of 2,263K German Wikipedia pages, 275K pages have a language link from Japanese Wikipedia, which will be used as the training data for German. Table 1 shows some statistics of the Wikipedia data for 31 languages that are processed by our system.

Table 2: Performance of the FPT.AI system

Lang.	Language	Regular	Late
ar	Arabic	73.25	73.25
bg	Bulgarian	**83.77**	83.28
ca	Catalan, Valencian	52.55	81.10
cs	Czech	**84.47**	83.74
da	Danish	**82.30**	81.74
de	German	22.62	81.26
el	Greek	**84.40**	84.10
en	English	82.23	81.96
es	Spanish, Castillian	80.60	80.60
fa	Persian	**81.70**	81.52
fi	Finnish	**83.62**	83.36
fr	French	21.59	80.68
he	Hebrew	**83.79**	84.21
hi	Hindi	**76.43**	75.65
hu	Hungarian	**85.46**	84.78
id	Indonesian	**81.93**	81.65
it	Italian	26.55	82.81
ko	Korean	**83.67**	83.77
nl	Dutch, Flemish	**83.29**	83.17
no	Norwegian	**80.53**	80.17
pl	Polish	**84.53**	84.07
pt	Portuguese	**83.23**	82.70
ro	Romanian, Moldavian	**84.60**	84.60
ru	Russian	**84.08**	83.44
sv	Swedish	**83.18**	83.44
th	Thai	**81.26**	81.16
tr	Turkish	**86.50**	86.03
uk	Ukrainian	**83.12**	82.61
vi	Vietnamese	**80.34**	80.42
zh	Chinese	**81.25**	80.60

3.2 Experimental Settings

We use the following training details for our proposed neural network architecture described in the previous section. The maximal sequence length of each article is set to 512 tokens. The learning rate is 2×10^{-5}. The batch size is 45. The models are run in 100 epochs, using apex and BERT weight decay is set to 0. The rate of the Adam optimizer is 10^{-8}. The sigmoid threshold for label assignment in monolingual models is set to 0.5. We use both CPU and GPU devices for training and prediction. The CPU is an Intel Xeon (R) E5-2699 v4 @2.20GHz. The GPU is a NVIDIA GeForce GTX 1080 Ti 11GB.

3.3 Evaluation

For each Wikipedia page, the system gives one or more predicted fine-grained categories. A predicted category is considered correct if and only if it is an exact match with the true category. The accuracy of a system is evaluated using the micro average F measure, which is the harmonic mean of micro-averaged precision and micro-averaged recall ratios. Note that the distribution of category

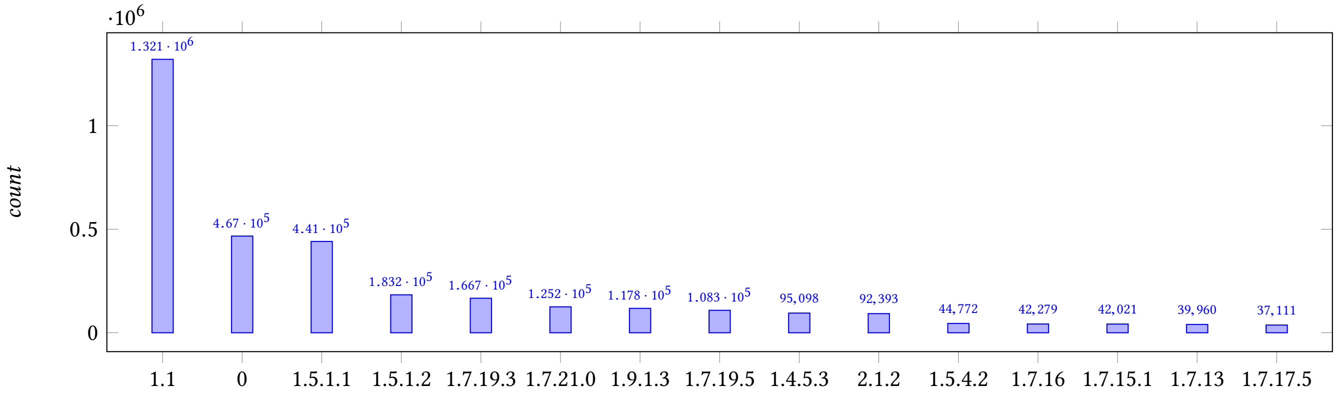

Figure 2: Top fifteen fine-grained ENE labels and their frequency

Table 3: Performance gaps with other top systems

Language	FPT.AI Score	Best Score	Δ	Best Team
Arabic	73.25	**76.27**	3.02	PribL
German	81.26	**81.86**	0.60	Ousia
French	80.68	**81.01**	0.33	Ousia
English	82.23	**82.73**	0.50	Uomfj
Spanish	80.60	**81.39**	0.79	Uomfj

in the test data may differ from that of the training data or target data. The organizer of the shared task keeps test datasets secret.

The FPT.AI system submitted the classification results for all the 30 target languages as regular submissions. Table 2 shows the *F* measures of our system. The scores are extracted from the official results published after the shared task by the organizer [4]. Due to a technical problem, some of our models did not perform well at the first submission (for example the French model) and we tried to re-submit the results a little bit after the deadline. The late submission results are shown in the last column of Table 2. The scores in bold font are the best scores among participating systems. In the regular submission, the FPT.AI system achieves the best scores for 23 languages. In the late submission, our system is able to add two more languages into its top-rank list, namely Catalan and Italian, achieving best scores for 25 out of 30 languages. In addition, this submission also pushes the best scores further for four languages, including Hebrew, Korean, Swedish, and Vietnamese.

Our system is outperformed by other participating systems in five languages, including Arabic, German, English, French and Spanish. Table 3 shows the score gaps between FPT.AI and the best performing systems for each language. We see that the performance gaps between our system with the top ones are relatively small, except for the Arabic language.

3.4 Ablation Analysis

In this subsection, we perform an ablation analysis of our method, where important components are systematically added. This experiment helps investigate the contribution of the hierarchical classification technique, the weighted loss function adjustment, and the voting strategy to the final score. All the scores are evaluated on the open leader-board datasets and directly copied from the leaderboard after each submission. Figure 3 shows the result.

Adding the hierarchy-aware classification technique improves the plain mBERT models by about 4.77% of absolute score in average across 30 langauges. Using the weighted loss function improves further the performance by about 2.10% in average. Finally, the voting strategy increases the performance further by about 0.9%.

4 CONCLUSION

In this paper, we have described the method underlying the FPT.AI system which participated in the SHINRA2020-ML shared task. The method exploits a cross-lingual representations of Wikipedia pages in 30 languages before fine-tuning to specific monolingual models. The method also relies on the strong transformers-based neural network models mBERT and on a special treatment for hierarchical multi-label classification which maximize learning capacity regarding the hierarchical structure of the labeled data. Another important technique in our method is a cross-lingual voting strategy which helps select the most reliable categories for each page.

Our system is able to achieve best scores for 25 out of 30 languages; and its performance gaps to the best performing systems of the other five languages are relatively small.

Our method has been designed toward a general-purpose cross-lingual representation and transfer learning, covering typologically diverse languages. Although it is able to attain good performances on many languages, especially on low resource ones, on high resource languages it has not achieved the best accuracy compared to several other strong participating systems. This suggests that there is room for different fine-tuning methods for high-resource languages that may improve further performance. This is an interesting research direction for our future work.

ACKNOWLEDGEMENTS

We would like to thank the organizing committee of the SHINRA2020 project, notably Satoshi Sekine for introducing this interesting shared tasks to us; Masako Nomoto and Koji Matsuda for supporting us with technical issues that we encountered during the task.

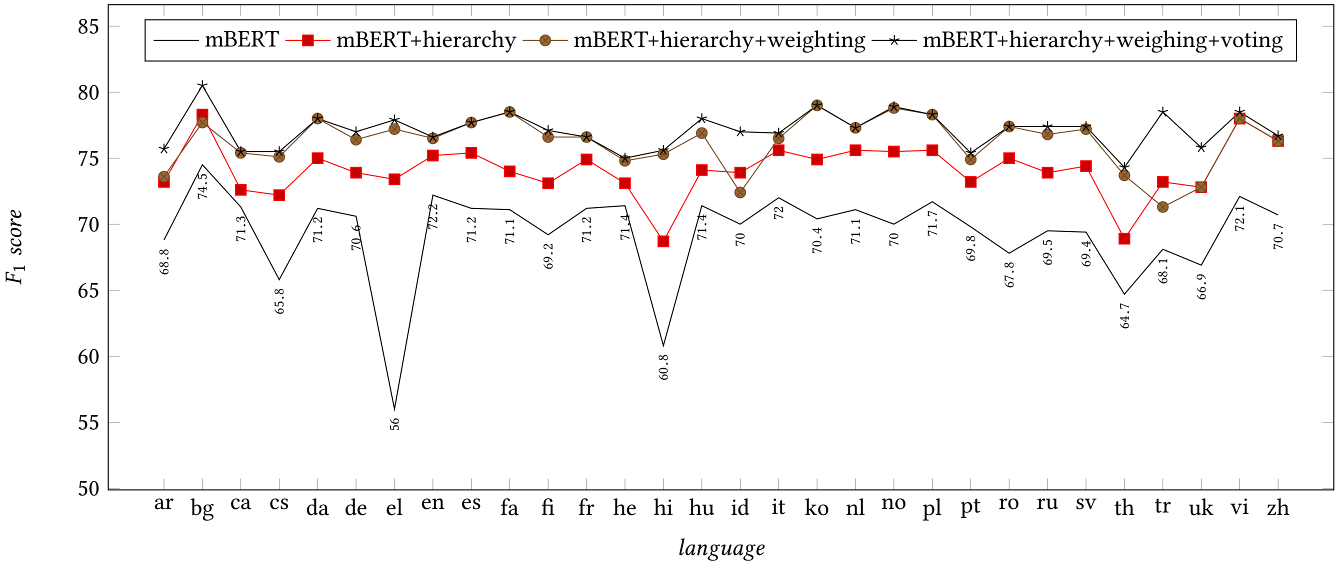

Figure 3: Performance of our model where parts of the method are systematically added.

REFERENCES

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*. Minnesota, USA, 1–16.

[2] Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A Massively Multilingual Multi-task Benchmark for Evaluating Cross-lingual Generalization. In *Preprint arXiv:2003:11080v3*.

[3] Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizerand detokenizer for Neural Text Processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations*. ACL, Brussels, Belgium, 66–71.

[4] Satoshi Sekine, Masako Nomoto, Kouta Nakayama, Sumida Asuka, Koji Matsuda, and Maya Ando. 2020. Overview of SHINRA2020-ML Task. In *Proceedings of the NTCIR-15 Conference*. Tokyo, Japan.

[5] Jônatas Wehrmann, Ricardo Cerri, and Rodrigo C. Barros. 2018. Hierarchical Multi-Label Classification Networks. In *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden.