Cross-lingual Extended Named Entity Classification of Wikipedia Articles

The Viet Bui
FPT Technology Research Institute
FPT University, Vietnam
vietb6@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.AI2 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 $E_4 \cup E_3 \cup E_2 \cup E_1$. 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

1https://ene-project.info
2https://fpt.ai/
the largest Wikipedia using a masked language modeling (MLM) objective [1].

Let \( Z = \{d_{[CLS]}^i, d_1^i, d_2^i, \ldots, d_s^i, d_{[SEP]}\} \) be the corresponding index sequence of \( S \), where \( d_k^i \) is the index of token \( k \) in \( V \) and \( i = \min(m, l - 2) \). 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 \( d_{\theta} \) dimensions:

\[
H = mBERT(Z)_{[CLS]} \in \mathbb{R}^{d_{\theta}}.
\]

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 \( Y_i \in \mathbb{R}^{d_{E_i}} \) be the label information of \( E_i \), where \( d_{E_i} \) is the number of labels in \( E_i \), with \( i \geq 2 \). More precisely:

\[
\begin{align*}
Y_2 &= \phi(H, d_{E_i}) \\
Y_3 &= \phi(H \circ Y_2, d_{E_i}) \\
Y_4 &= \phi(H \circ Y_2 \circ Y_3, d_{E_i}),
\end{align*}
\]

where \( \circ \) 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 \( Y_i, i = 2, 3, 4 \) be the ground-truth one-hot vector corresponding to the predicted target vector \( \tilde{Y}_i \). For each sample, we define a loss function to be optimized as follows:

\[
J(\theta) = \sum_{i=2}^{4} L_i(\tilde{Y}_i, Y_i, W_i),
\]

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

\[
w_i = \min \left( \frac{\tilde{c}_i}{\tilde{c}_i}, 1 \right).
\]

Here \( \tilde{c}_i \) is the frequency vector of all labels in \( E_i \), \( \tilde{c}_i \) is the element-wise average of \( \tilde{c}_i \) and \( 1 \) 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 \( \tilde{Y}_4 \) by passing the main content to the corresponding monolingual model. This vector is then passed to the sigmoid function \( \hat{Y}_4 = \sigma(\tilde{Y}_4) \), which squares \( \tilde{Y}_4 \) 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 \( \epsilon \), 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

<table>
<thead>
<tr>
<th>Language</th>
<th>Pages</th>
<th>Links from JP</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>English (en)</td>
<td>5,790,377</td>
<td>439,354</td>
<td>7.6</td>
</tr>
<tr>
<td>Spanish (es)</td>
<td>1,500,013</td>
<td>257,835</td>
<td>17.2</td>
</tr>
<tr>
<td>French (fr)</td>
<td>2,074,648</td>
<td>318,828</td>
<td>15.4</td>
</tr>
<tr>
<td>German (de)</td>
<td>2,262,582</td>
<td>274,732</td>
<td>12.1</td>
</tr>
<tr>
<td>Chinese (zh)</td>
<td>1,041,039</td>
<td>267,107</td>
<td>25.7</td>
</tr>
<tr>
<td>Russian (ru)</td>
<td>1,523,013</td>
<td>253,012</td>
<td>16.6</td>
</tr>
<tr>
<td>Portuguese (pt)</td>
<td>1,014,832</td>
<td>217,896</td>
<td>21.5</td>
</tr>
<tr>
<td>Italian (it)</td>
<td>1,496,975</td>
<td>270,295</td>
<td>18.1</td>
</tr>
<tr>
<td>Arabic (ar)</td>
<td>661,205</td>
<td>73,054</td>
<td>11.0</td>
</tr>
<tr>
<td>Japanese</td>
<td>1,136,222</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Indonesian (id)</td>
<td>451,336</td>
<td>115,643</td>
<td>25.6</td>
</tr>
<tr>
<td>Turkish (tr)</td>
<td>321,937</td>
<td>111,592</td>
<td>34.7</td>
</tr>
<tr>
<td>Dutch (nl)</td>
<td>1,955,483</td>
<td>199,983</td>
<td>10.2</td>
</tr>
<tr>
<td>Polish (pl)</td>
<td>1,316,130</td>
<td>225,552</td>
<td>17.1</td>
</tr>
<tr>
<td>Persian (fa)</td>
<td>660,487</td>
<td>169,053</td>
<td>25.6</td>
</tr>
<tr>
<td>Swedish (sv)</td>
<td>3,759,167</td>
<td>180,948</td>
<td>4.8</td>
</tr>
<tr>
<td>Vietnamese (vi)</td>
<td>1,200,157</td>
<td>116,280</td>
<td>9.7</td>
</tr>
<tr>
<td>Korean (ko)</td>
<td>439,577</td>
<td>190,807</td>
<td>43.7</td>
</tr>
<tr>
<td>Hebrew (he)</td>
<td>236,984</td>
<td>103,137</td>
<td>43.5</td>
</tr>
<tr>
<td>Romanian (ro)</td>
<td>236,984</td>
<td>103,137</td>
<td>23.5</td>
</tr>
<tr>
<td>Norwegian (no)</td>
<td>501,475</td>
<td>135,935</td>
<td>27.1</td>
</tr>
<tr>
<td>Czech (cs)</td>
<td>420,195</td>
<td>135,935</td>
<td>25.1</td>
</tr>
<tr>
<td>Ukrainian (uk)</td>
<td>420,195</td>
<td>135,935</td>
<td>20.5</td>
</tr>
<tr>
<td>Hindi (hi)</td>
<td>129,141</td>
<td>30,547</td>
<td>23.6</td>
</tr>
<tr>
<td>Finnish (fi)</td>
<td>450,537</td>
<td>144,750</td>
<td>32.1</td>
</tr>
<tr>
<td>Hungarian (hu)</td>
<td>443,060</td>
<td>120,295</td>
<td>27.2</td>
</tr>
<tr>
<td>Danish (da)</td>
<td>242,523</td>
<td>91,811</td>
<td>35.6</td>
</tr>
<tr>
<td>Thai (th)</td>
<td>129,294</td>
<td>59,791</td>
<td>46.2</td>
</tr>
<tr>
<td>Catalan (ca)</td>
<td>601,473</td>
<td>139,032</td>
<td>23.1</td>
</tr>
<tr>
<td>Greek (el)</td>
<td>157,566</td>
<td>60,513</td>
<td>38.4</td>
</tr>
<tr>
<td>Bulgarian (bg)</td>
<td>248,913</td>
<td>89,017</td>
<td>35.7</td>
</tr>
</tbody>
</table>

$c_1, c_2, \ldots, c_k$ 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.

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 $2 \times 10^{-5}$. 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 $10^{-6}$. 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
important components are systematically added. This exper-
fications technique, the weighted loss function adjustment, and the
ment helps investigate the contribution of the hierarchical classi-

In this subsection, we perform an ablation analysis of our method,
except for the Arabic language.

Table 3: Performance gaps with other top systems

<table>
<thead>
<tr>
<th>Language</th>
<th>FPT.AI Score</th>
<th>Best Score</th>
<th>∆</th>
<th>Best Team</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>73.25</td>
<td>76.27</td>
<td>3.02</td>
<td>PribL</td>
</tr>
<tr>
<td>German</td>
<td>81.26</td>
<td>81.86</td>
<td>0.60</td>
<td>Ousia</td>
</tr>
<tr>
<td>French</td>
<td>80.68</td>
<td>81.01</td>
<td>0.33</td>
<td>Ousia</td>
</tr>
<tr>
<td>English</td>
<td>82.23</td>
<td>82.73</td>
<td>0.50</td>
<td>Uomfj</td>
</tr>
<tr>
<td>Spanish</td>
<td>80.60</td>
<td>81.39</td>
<td>0.79</td>
<td>Uomfj</td>
</tr>
</tbody>
</table>

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 sub-
mission 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 Span-
ish. Table 3 shows the score gaps between FPT.AI and the best per-
forming 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 exper-
iment helps investigate the contribution of the hierarchical classi-
fication technique, the weighted loss function adjustment, and the
voting strategy to the final score. All the scores are evaluated on
the open leaderboard datasets and directly copied from the leader-
board 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 aver-
age across 30 languages. Using the weighted loss function im-
proves 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 hierarchi-
cal multi-label classification which maximize learning capacity re-
garding the hierarchical structure of the labeled data. Another im-
portant 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 lan-
guages; 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 re-
source 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 inter-
esting 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**


