Models Distillation with Lifelong Deep Learning for Vietnamese Biomedical Named Entity Recognition

Thi-Cham Nguyen¹ ², Hoang-Quynh Le², Duy-Cat Can²†, and Quang-Thuy Ha²
¹Haiphong University Of Medicine and Pharmacy, Haiphong, Vietnam
nthicham@hpmu.edu.vn
²Faculty of Information Technology
VNU University of Engineering and Technology, Hanoi, Vietnam
{lhuqnh, catcd, thuyhq}@vnu.edu.vn

Abstract—In realistic data, named entities may appear in a variety of rich contexts with unique characteristics and the performance of the named entity recognition (NER) task directly affects other NLP problems. Although both lifelong learning and deep learning have proven effective in many problems, including NER, the suitable combination of these two research directions is still limited. This paper describes a lifelong deep learning model for Vietnamese Biomedical NER based on model distillation mechanism. Our approach achieves potential results, helps to boost 2.16% compared to original deep learning model.

Index Terms—lifelong deep learning, cross-domain, knowledge distillation, model distillation, bi-lstm

I. INTRODUCTION

Named entity recognition (NER) aims to automatically identify the entity mentions in text, which is a vital step for further Natural Language Processing (NLP) tasks and has received much attention from the research community in the past two decades. Although it is a fundamental task, NER is still challenging, in realistic data, named entities may appear in a variety of rich contexts with unique characteristics. Manual data annotation is time-consuming and labour-intensive, and it cannot be guaranteed to cover all instances of the data. Therefore, both taking advantage of knowledge from similar data sources to improve model efficiency as well as automatic feature engineering are important research needs.

Lifelong Machine Learning (LML) is an advanced machine learning model with the ability to imitate human learning [1]. In lifelong machine learning, the learning process is continuous, the knowledge learned from previous tasks is accumulated in the knowledge base, and the knowledge is used to support the learning tasks in the future. Several works have applied LML to solve NER problem, example includes Monaikul et al. (2021) [2], Nguyen et al. (2019) [3], Wang et al. (2021) [4].

Although deep learning models have proven effective in many problems, including NLP, the combination of LML and deep learning is still limited, as well-discussed in Parisi et al. (2019) [5]. Deep learning models typically learn features from stationary batches of training data while LML takes advantage of the incremental information. Since LML continually acquires available information from non-stationary data distributions, it remains a long-standing challenge for combining with deep neural network models. Lange et al. (2019) [6] annotated a new dataset only for the new entity types and use this data to update the model in a LML setting. However, this approach is generally leads to the catastrophic forgetting of previous entity types or interference [1]. Monaikul et al. (2021) [2] instead tried to take advantage of the existing NER model to impart knowledge about the already-recognized entities to a new model.

This paper presents a Vietnamese biomedical Named Entity Recognition in the form of a sequence labeling task using lifelong deep learning technique that takes advantage of the past knowledge pattern close to the current NER task. The main contributions of our work can be concluded as:

i. We proposed a novel model distillation architecture with lifelong deep learning for NER. Our model took advantage of the near-domain distilled knowledge to build a lifelong deep neural network that resolving the limitation of catastrophic forgetting.

ii. We constructed a deep learning model that investigated several types of features sources, suitable to the characteristics of Vietnamese biomedical text.

The remaining of this paper is organized as follows: Section 2 describes our proposed lifelong deep learning model. Section 3 then show the experimental results and gives some evaluation. Section 4 discusses some related works. Finally, Section 5 is the conclusion.

II. PROPOSED LIFELONG DEEP LEARNING MODEL

A. Overall architecture

In order to take advantage of the knowledge from the near data domains, we have proposed an architecture that distills the knowledge (models) in a lifelong deep learning system. Figure 1 depicts the overall architecture of our system that consists of three phase: (i) Model distillation from cross domain datasets, (ii) Model training from in-domain dataset, and (iii) Model evaluating on in-domain dataset.

This work has been supported by VNU University of Engineering and Technology under project number CN20.08.

†Corresponding author.
1) **Model distillation**: In this phase, we extract the knowledge from various cross-domain datasets in the form of distilled models. We focus on two components of NER models: the Bi-LSTM modeling parameters and Classification models. Given a set of data, we build and train an end-to-end NER model to obtain distilled models.

   Each training input is pre-processed through four steps: sentence segmentation, tokenization, Vietnamese word segmentation, and part-of-speech (POS) tagging. In this work, we use pre-trained VNCoreNLP model\(^1\) for pre-processing data. These pre-processed documents are then fed to a input representation model to transform each sequence of tokens into an input matrix. For distilling the knowledge, we use a bi-directional Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) unit to extract distilled Bi-LSTM modeling parameters. The output sequence of Bi-LSTM model are classification into NER tags respectively. We store these Bi-LSTM and classification models for the next training phase.

2) **Model training**: The named entity prediction model is constructed on the in-domain training dataset. Two first input preparation procedures (pre-processing and input representation) are shared with the previous model distillation phase. In the input modeling step, we also use a multi-layer bi-directional LSTM model to represent sequential information. However, to take advantage of the distilled knowledge from the cross-domain dataset, we use pre-trained Bi-LSTM parameters from Phase 1 to extract input representation and combine them with the corresponding sequential information. Besides using distilled Bi-LSTM parameter, we also make use of the predictions from the pre-trained classification model (from Phase 1) as the additional basis for the final classification model. The detail of these combinations are illustrated in the next sections.

3) **Model evaluating**: The testing dataset is also pre-processed by VNCoreNLP tool to extract the same features as the above phases. Then, the distilled models from Phase 1 are used to extract the sequential information and distilled classification model’s predictions. We leverage these extracted features and prediction in the combination with Phase 2 models to predict the final output, and evaluate the performance with standard measures.

   We use the same training datasets and test datasets for all experiments.

### B. Proposed Deep Learning Model

Figure 2 shows the overall architecture of proposed deep learning model for Vietnamese Biomedical Named Entity Recognition. We build the deep learning model on pre-processed documents with three main phases: input representation, Bi-LSTM modeling, and classification phase. The proposed models is divided into two channels with similar architecture: main model channel and distilled models channel. Given a pre-processed document as input, we use FastText with pre-trained word embedding and POS tag representation model to produce an input matrix. Two parallel Bi-LSTM networks are followed to model the sequential information. Finally, two multi-layer perceptron models are combined to assign each token a corresponding NER tag.

1) **Input representation**: In the input representation layer, each token in the input document is transformed into a vector \( \mathbf{x}_e \in \mathbb{R}^d \), where \( d \) is the desired embedding dimension. In order to capture more features along the input sequence, we take advantage of three types of information, including:

---

\(^1\)https://github.com/vncorenlp/VnCoreNLP
- **Pre-trained fastText embeddings** [7]: which learned the word representation based on its external context, therefore allows words that often appear in similar context to have similar representations. Each token $i$ is transformed into a vector $\mathbf{x}_i^c$ by looking up the embedding matrix $\mathbf{T}^c \in \mathbb{R}^{d_c \times |V^c|}$, where $V^c$ is the vocabulary of all words.

- **Character-based embeddings**: Convolutional Neural Network (CNN) is an effective approach to learn the character-level representations that offer the information about word morphology and shape (like the prefix or suffix of word) [8]. Given a token composed of $n$ characters $c_1, c_2, ..., c_n$, we first represent each character $c_i$ by an embedding $\mathbf{r}_i$ using a look-up table $\mathbf{T}^c \in \mathbb{R}^{d_c \times |V^c|}$, where $V^c$ is the alphabet. A deep CNN with various window sizes is applied on the sequence $\{r_1, r_2, ..., r_n\}$ to capture the character features. A pooling layer is followed to produce the final character embedding $\mathbf{x}_i^c$.

- **POS tag embeddings**: A token may have more than one meaning representing by its grammatical tag such as noun, verb, adjective, adverb, etc. To address this problem, we use the part-of-speech (POS) tag information in the token representation. We randomly initialize the embeddings matrix $\mathbf{T}_i \in \mathbb{R}^{d_t \times 56}$ for 56 OntoNotes v5.0 POS tags. Each POS tag is then represented as a corresponding vector $\mathbf{x}_i^t$.

Finally, we concatenate the word embedding, character-based embedding and POS tag embedding, and transform them into the final token embedding as follow:

$$\mathbf{x}_i = \tanh \left( [\mathbf{x}_i^c \oplus \mathbf{x}_i^c \oplus \mathbf{x}_i^t] \mathbf{W}^x + \mathbf{b}^x \right) \quad (1)$$

2) **Bi-LSTM modeling**: We build a RNN with LSTM unit model on the input matrix to calculates the hidden state $\mathbf{h}_t$ for each token $\mathbf{x}_t$ at time step $t$ as below:

$$i_t = \sigma (\mathbf{x}_t \mathbf{W}_i + \mathbf{h}_{t-1} \mathbf{U}_i + \mathbf{b}_i) \quad (2)$$

$$f_t = \sigma (\mathbf{x}_t \mathbf{W}_f + \mathbf{h}_{t-1} \mathbf{U}_f + \mathbf{b}_f) \quad (3)$$

$$\tilde{c}_t = \tanh (\mathbf{x}_t \mathbf{W}_c + \mathbf{h}_{t-1} \mathbf{U}_c + \mathbf{b}_c) \quad (4)$$

$$c_t = i_t \odot \tilde{c}_t + f_t \odot c_{t-1} \quad (5)$$

$$o_t = \sigma (\mathbf{x}_t \mathbf{W}_o + \mathbf{h}_{t-1} \mathbf{U}_o + \mathbf{b}_o) \quad (6)$$

$$\mathbf{h}_t = o_t \odot \tanh (c_t) \quad (7)$$

where, $\mathbf{W}_i, \mathbf{U}_i, \mathbf{W}_f, \mathbf{U}_f, \mathbf{W}_c, \mathbf{U}_c, \mathbf{W}_o$, and $\mathbf{U}_o$ are model’s trainable parameters; $\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c$, and $\mathbf{b}_o$ are bias terms; $\sigma$ denotes the sigmoid function, and $\odot$ denotes the element-wise product.

In addition to the main Bi-LSTM model, we use the distilled Bi-LSTM modeling parameters extracted from cross-domain datasets to extract distilled $\mathbf{h}_t$ for each token $\mathbf{x}_t$.

$$\tilde{\mathbf{H}} = \{\tilde{\mathbf{h}}_i\}_{i=1}^n = \tilde{\text{biLSTM}}(\{\mathbf{x}_i\}_{i=1}^n) \quad (8)$$

The distilled $\mathbf{h}_t$ are then combine with the hidden state $\mathbf{h}_t$ in the weighted-fashion, as follow:

$$\tilde{\mathbf{h}}_t = \mathbf{h}_t \oplus \lambda \tilde{\mathbf{h}}_t \quad (9)$$

where $\oplus$ denotes vector concatenation, and $\lambda \in [0, 1]$ is a trainable weight.
3) Classification: The hidden vector of each token from previous layer are then fed into a fully connected multi-layer perceptron network (MLP). We choose hyperbolic tangent as the non-linear activation function in the hidden layer. I.e.
\[ h = \tanh (\bar{h}W^h + b^h) \] (10)

where \(W^h\) and \(b^h\) are weight and bias terms. We apply multi hidden layer to produce higher abstraction-level features. The output \(h\) of the last hidden layer is the highly abstract representation of input features, which is then fed to a softmax classifier to predict a (K+1)-class distribution over labels \(\hat{y}\):
\[ \hat{y} = \text{softmax} (hW^y + b^y) \] (11)

We also use the prediction of distilled classification model to improve the final results. We only use the distilled predictions that have confidence score exceed heuristic threshold \(\alpha\). Because two predictions could be conflicted, we developed a set of heuristics that merges the outputs for each token:
- If two models predict B- tag, then output=B- with the highest probability.
- If one prediction is a B- tag and the other is O tag, then output=B-.
- If a model predicts an I- tag, output=I- only if it matches the preceding tag in the sequence (same entity type). Otherwise, the prediction is treated as O.
- If all predictions are O tag, then output=O.

### III. EXPERIMENT AND DISCUSSION

#### A. Dataset

In this paper, we evaluate our proposed lifelong deep learning model using the new constructed Vietnamese biomedical NER dataset. Table I summarizes the distribution of the different types of entity in this dataset. The dataset was crawled from two separated domains: online news and scientific report. Both datasets are labeled by two annotators independently with three types of biomedical entities: disease (DIS), chemical (CHE), and treatment (TRM) entities. The inter annotator agreement index reached 0.86%. In this paper, we use the scientific report dataset consisting of 82 documents as the training and testing dataset (in-domain dataset). The online news dataset of 84 documents is used as a cross-domain dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>No. 1</th>
<th>No. 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Cross-domain</td>
<td>In-domain</td>
</tr>
<tr>
<td>Documents</td>
<td>84</td>
<td>82</td>
</tr>
<tr>
<td>Sentences</td>
<td>2360</td>
<td>1369</td>
</tr>
<tr>
<td>Number of entities</td>
<td>CHE</td>
<td>DIS</td>
</tr>
<tr>
<td></td>
<td>437</td>
<td>2134</td>
</tr>
<tr>
<td></td>
<td>228</td>
<td>1171</td>
</tr>
<tr>
<td>Total</td>
<td>3230</td>
<td>1831</td>
</tr>
<tr>
<td>Average entity</td>
<td>DIS</td>
<td>TRM</td>
</tr>
<tr>
<td></td>
<td>1.826</td>
<td>1.842</td>
</tr>
<tr>
<td>Average length</td>
<td>CHE</td>
<td>TRM</td>
</tr>
<tr>
<td></td>
<td>2.150</td>
<td>2.009</td>
</tr>
</tbody>
</table>

The in-domain and cross-domain datasets vary in specification of data. The documents in the in-domain dataset are usually shorter in length, with an average of 17 sentences each. The number of entities is unevenly distributed, disease entities account for the highest rate with 63.96%. The remaining two types of entities - chemical and treatment - account for 12.45% and 23.59%, respectively. The cross-domain dataset has a similar distribution. However, the number of entities of the cross-domain dataset is 1.76 times more than the in-domain dataset. The average length of the entities is also slightly longer.

#### B. Objective Function and Training Method

The proposed deep learning model and the distilled models are trained using cross-entropy penalized objective function. We define the training objective for a data sample as:
\[ L(\theta) = - \sum_{i=0}^{K} y_i \log \hat{y}_i + \lambda \| \theta \|^2 \] (12)

where \(y \in \{0, 1\}^{(K+1)}\) indicating the one-hot vector represented the target label, and \(\lambda\) is a regularization coefficient.

To compute the model parameters \(\theta\), we minimize \(L(\theta)\) by applying mini-batch gradient descent (GD) with Adam optimizer [9] in our experiments. \(\theta\) is randomly initialized with Xavier normal [10] and is updated via back-propagation through neural network structures.

We perform 5-fold cross-validation on the in-domain dataset. The training and testing process has been conducted 10 times with different random seeds and the average results are reported. For evaluation, the predicted entities were strict matching with the golden annotated data with common machine learning evaluation metrics: precision (P), recall (R), and F1 score.

#### C. Experimental results and discussion

1) System’s performance: We compared our model with various common machine learning algorithms, including Conditional Random Fields (CRFs) - a most commonly used model for NER, Recurrent Neural Network (RNN), and Long Short-Term Memory network (LSTM). For a fair comparison, we re-implemented two typical deep learning models for the NER problem BiLSTM and BiLSTM-CRF. We also compare the proposed model with two versions of these models trained on augmented data instead of a lifelong deep learning architecture. Table II shows the classification metrics of our four model variants with different comparative models.

With the comparative models, the BiLSTM-CRF model gives the highest results on in-domain data with macro-average result of 85.94%. The experimental results also show that the CRFs model has the ability to correct the predictions of the BiLSTM model with an improvement of 1.27%. The augmented data from online news also bring improvement to both BiLSTM and BiLSTM-CRF models with the change of F1 score at 0.48%.
## Table II: Statistic of System’s Performances

<table>
<thead>
<tr>
<th>Model</th>
<th>Disease P</th>
<th>Disease R</th>
<th>Disease F1</th>
<th>Chemical P</th>
<th>Chemical R</th>
<th>Chemical F1</th>
<th>Treatment P</th>
<th>Treatment R</th>
<th>Treatment F1</th>
<th>Macro average P</th>
<th>Macro average R</th>
<th>Macro average F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF</td>
<td>85.04</td>
<td>73.23</td>
<td>78.69</td>
<td>73.94</td>
<td>79.14</td>
<td>80.49</td>
<td>82.67</td>
<td>79.33</td>
<td>80.97</td>
<td>78.35</td>
<td>75.50</td>
<td>76.77</td>
</tr>
<tr>
<td>RNN</td>
<td>84.93</td>
<td>75.30</td>
<td>81.59</td>
<td>80.94</td>
<td>80.94</td>
<td>81.04</td>
<td>82.93</td>
<td>80.94</td>
<td>81.87</td>
<td>85.53</td>
<td>77.93</td>
<td>81.02</td>
</tr>
<tr>
<td>LSTM</td>
<td>87.07</td>
<td>83.29</td>
<td>85.14</td>
<td>86.50</td>
<td>80.50</td>
<td>83.40</td>
<td>84.55</td>
<td>80.11</td>
<td>82.87</td>
<td>86.04</td>
<td>81.30</td>
<td>83.60</td>
</tr>
<tr>
<td>BiLSTM</td>
<td>87.46</td>
<td>89.29</td>
<td>87.72</td>
<td>90.78</td>
<td>79.60</td>
<td>84.82</td>
<td>84.52</td>
<td>78.63</td>
<td>81.47</td>
<td>87.59</td>
<td>82.07</td>
<td>84.87</td>
</tr>
<tr>
<td>+ w data aug*</td>
<td>88.16</td>
<td>88.28</td>
<td>88.22</td>
<td>91.37</td>
<td>80.20</td>
<td>85.42</td>
<td>84.48</td>
<td>79.28</td>
<td>81.80</td>
<td>88.00</td>
<td>82.59</td>
<td>85.15</td>
</tr>
<tr>
<td>BiLSTM+CRF</td>
<td>87.88</td>
<td>89.17</td>
<td>88.97</td>
<td>86.19</td>
<td>85.51</td>
<td>85.82</td>
<td>84.07</td>
<td>82.29</td>
<td>83.04</td>
<td>86.34</td>
<td>85.65</td>
<td>85.94</td>
</tr>
<tr>
<td>+ w data aug*</td>
<td>89.60</td>
<td>90.55</td>
<td>90.11</td>
<td>86.66</td>
<td>85.17</td>
<td>85.91</td>
<td>84.30</td>
<td>82.23</td>
<td>83.25</td>
<td>86.87</td>
<td>85.98</td>
<td>86.42</td>
</tr>
<tr>
<td>Proposed model</td>
<td>90.68</td>
<td>91.21</td>
<td>90.95</td>
<td>95.27</td>
<td>82.41</td>
<td>88.38</td>
<td>87.95</td>
<td>82.19</td>
<td>84.97</td>
<td>91.30</td>
<td>85.27</td>
<td>88.10</td>
</tr>
<tr>
<td>+ w CRF</td>
<td>91.27</td>
<td>90.17</td>
<td>90.71</td>
<td>95.32</td>
<td>82.83</td>
<td>88.64</td>
<td>88.10</td>
<td>81.36</td>
<td>84.59</td>
<td>91.56</td>
<td>84.78</td>
<td>87.98</td>
</tr>
<tr>
<td>– w/o distilled BiLSTM</td>
<td>89.82</td>
<td>89.18</td>
<td>89.05</td>
<td>93.59</td>
<td>79.55</td>
<td>86.00</td>
<td>86.39</td>
<td>79.69</td>
<td>82.90</td>
<td>89.65</td>
<td>82.81</td>
<td>85.98</td>
</tr>
<tr>
<td>– w/o distilled prediction</td>
<td>88.36</td>
<td>89.29</td>
<td>88.82</td>
<td>92.98</td>
<td>80.53</td>
<td>86.31</td>
<td>85.58</td>
<td>80.37</td>
<td>82.89</td>
<td>88.97</td>
<td>83.39</td>
<td>86.01</td>
</tr>
</tbody>
</table>

*: with data augmentation (train on both cross-domain and in-domain training datasets)

When comparing our proposed model with the previous comparative models, the lifelong deep learning model outperforms other models by large margin. The F1 score for **DISEASE**, **CHEMICAL** and **TREATMENT** entities are 90.95%, 88.38% and 84.97%. The macro-average of the precision, recall and F1 scores are 3.71%, 3.20% and 3.43% higher than the baseline BiLSTM model, respectively. Previous studies as well as experimental results of comparative models have demonstrated the effectiveness of the CRFs model when combined with the BiLSTM model. We also tried adding CRFs after the proposed model to correct the model’s **B-**, **I-** non-consecutive errors. However, the results did not meet our expectations. When adding a CRFs layer after the proposed model, the precision increases slightly with small improvement at 0.26%. However the macro-average recall dropped from 85.27% to 84.78%. As a result, the overall F1 score is 0.12% lower than the original model.

We also investigate the contribution of each model component by remove each component from the proposed model and observe the change of the F1 results. The experimental results show that both distilled BiLSTM and distilled prediction model have an impact on the final result (2.12% and 2.09% respectively).

2) **Experiment on size of cross-domain datasets:** In this experiment, we change the size of the cross-domain dataset and track the resulting change. Figure 3 gives the illustration of F1 results with the ratio of cross-domain dataset used from 10 to 100 percent. Overall, the F1 scores of all three entity types are directly proportional to the percentage of using online news datasets to distill the knowledge. This experimental result demonstrates the potential of cross-domain dataset scaling in improving the performance of deep learning models. We leave this problem for the future work.

3) **Experiment on threshold of distilled classification model:** In this experiment, we change the decision-making threshold of the distilled classification model and monitor the corresponding F1 score change. Figure 4 shows the change of the F1 score when the threshold is changed from 0.5 to 1.0. With the low decision thresholds (from 0.5 to 0.6), the results for all three entity types are lower than the baseline BiLSTM model.
The reason is that the predictions of the distilled classification model are uncertain, causing the precision of the model to decrease significantly. When gradually increasing the decision threshold, the results gradually increase until the threshold reaches 0.85. However, as the decision threshold increases from 0.85 to 1.0, the overall results of the model decrease to approximately equal to the baseline model.

IV. RELATED WORK

In recent years, LML and deep learning have attracted much attention from the research community in a wide range and abundance of problems, including NER. Although significant advances have been made in domain-specific learning with LML and deep learning, extensive research efforts are required to develop robust combinations of these advantage methods. As mentioned before, this combination has the major drawback of catastrophic forgetting or interference.

Based on the idea of Knowledge Distillation model, Mon-aikul et al. (2021) [2] proposed a LML-based NER model, that retains knowledge of old entities and only requires new training material to learn the new entity types. The Knowledge Distillation architecture allows this model to continuously learn new entity types without losing the ability to recognize previously acquired entities types.

Also applied Knowledge Distillation model, Wang et al. (2021) [4] proposed a lifelong learning model base on deep learning for disease diagnosis on clinical notes. To overcome the disadvantages of catastrophic forgetting, the proposed system used an attention mechanism to combine medical entities and context, embedding episodic memory and consolidation to retain knowledge, such that the learned model was capable of adapting to sequential disease-diagnosis tasks in new data.

Our idea to combine LML and deep learning is quite different to the idea of the Knowledge Distillation model [11] and its improved approaches [2], [4]. In this study, instead of using LML to ensure the model is capable of recognizing old entities, we use LML to increase model performance with entities of the same type in near-domain data. Because Vietnamese is a low-resource language, building large and suitable data for each small sub-domain is time-consuming and labour-expensive. With this idea, we only need a moderate amount of data for a new near-domain data to perform the NER task on basic entities with better performance.

In terms of LML for Vietnamese NER specifically, the work which is most related is by Nguyen et al. (2019) [3], who propose a character-level deep LML for NER in Vietnamese news text, called DeepLML-NER. The main model is a deep neural network that takes advantage of character-level embedding. At each step, the model extracts a set of “prefix-features” into a knowledge base, which is then used to solveNER task in new data. Regardless of the different domain and entity types, our research and Nguyen et al. are very different on many points. Firstly, we proposed different deep learning structures. Secondly, the core idea to build the LML architecture of Nguyen et al. is to construct a knowledge base and used it to solve NER in new data. We distilled the knowledge from the previous model in a models distillation architecture and used this knowledge for NER in near data domains.

V. CONCLUSION

In this paper, we proposed a lifelong deep learning model, based on model distillation architecture for Vietnamese biomedical NER. Our model took advantage of the near-domain distilled knowledge to build a lifelong deep neural network that resolving the limitation of catastrophic forgetting. We also investigated the contributions of the cross-domain dataset in the lifelong deep learning model. The experimental results demonstrated the effectiveness of our proposed model, all of its components as well as in comparison with some other related works. For future work, we plan to apply advanced techniques such as the attention mechanism to improve the system performance.

ACKNOWLEDGMENT

This work has been supported by VNU University of Engineering and Technology under project number CN20.08.

REFERENCES