A BERT based Chinese Named Entity Recognition method on ASEAN News

Haoyu Zhuang¹, Fu Wang¹, Songlin Bo², and Yongzhong Huang¹*

¹Guilin University of Electronic Technology, Guilin 541004, China
²Tianyun junan (Beijing) technology co., LTD., Beijing 100080, China
*Corresponding author’s e-mail: ruhuac@guet.com

Abstract. As the first step of building a knowledge graph to record the ASEAN counties' information, we aim to conduct Named-entity Recognition (NER) on the Chinese news about ASEAN counties. We employ a Bi-directional gated recurrent unit to replace the LSTM architecture to improve both models' effectiveness and capability in understanding polysemous words. The state-of-the-art word embedding model, BERT, has also been included to generate qualified word vectors for the NER task. Besides, we also propose a similarity-based dataset partition method to help model learning the polysemy within the Chinese news. Experiments have been done to demonstrate that the combination of such improvements can benefit the models' performance in identifying different types of named entities.

1. Introduction

Due to the Belt and Road diplomatic strategy and the establishment of ASEAN-China community with a shared future, Chinese researchers have done a lot of studies on related works under a wide range of topics. As a sub-problem of building a knowledge graph to record the ASEAN counties' trending information and diplomatic polices, how to automatically extract key knowledge from internet becomes a challenge. Taking a step on addressing this challenge, we aim to utilize the existing Named-entity Recognition (NER) technologies to extract key information from plan Chinese texts.

The concept of NER first appeared in the Information Understanding Conference[3], and it focuses on extracting structured information from semi-structured and unstructured texts. Early researches on this direction were only based on dictionaries, manual rules, and statistical-based methods. With the improvement of computational technology, the appearance of machine learning and deep learning models tremendous push the development of a large amount of Natural Language Processing (NLP) tasks. However, there are still many difficulties when adopting such technologies on Chinese texts. On the one hand, most NLP methods are based on English and infeasible to apply to Chinese texts directly because Chinese texts do not have explicit delimiter, i.e., the whitespace, to separate words. On the other hand, regarding the various types of entities in the ASEAN countries, a large number of polysemous words would be introduced when using Chinese to store information written in foreign languages. For example, the capital of Vietnam is “Ho Chi Minh City”, where “Ho Chi Minh” may be misidentified as an adult's name. Besides, abbreviations and aliases are also difficult for algorithms to handle, e.g., “Yindunixiya” (Indonesia) is denoted by “Yinni” in most of the news, and the capital of Thailand "Bangkok" can be replaced by the "City of Angels".

In this paper, instead of the LSTM model, we utilize a Bi-directional gated recurrent unit with conditional random field layers to identify named entities because this model suffers less on the gradient
vanishing problem when processing long sequence data and enables a better effect and efficiency than LSTM when processing long sequence data. On the other hand, we introduce the pre-trained BERT model to generate high-quality word vectors, which outperform word2vec methods on word embedding tasks. Moreover, the order of the training examples has a noticeable impact on the trained models' performance, especially when handling polysemous words. Motivated by this, we proposed a similarity based dataset partition methods, which enable model begin to learn the polysemy at early training stage.

2. Related work
Let us start with a brief overview of the development of Named-entity Recognition (NER) on Chinese texts and word embedding methods.

2.1. Chinese Named-entity Recognition
As an essential sub-task of information extraction, NER aims to locate and classify named entities, e.g., organizations, locations, people's names, from unstructured and semi-structured texts[3]. Due to the difference between Chinese and English, which has relatively more straightforward grammar and does not require text segmentation, conducting NER on Chinese text is much more difficult. The development of NER on Chinese texts started in the early 1990s. Su et al. [9] proposed a segmentation and tagging system for Chinese texts. Since then, a great effort has been made to improve the performance of NER methods in this direction, e.g., [4,11,12]

2.2. Word Embedding
Currently, most NLP tasks, including NER, relies on word embedding technologies, which project words and phrases in human languages into vector spaces and enable the downstream models to further process such information in a vectorized form for specific tasks. Proposed by Mikolov et al.[6], word2vec toolkit composed Continuous Bag-of-Words Model (CBOW) and continuous skip-gram model methods and significantly boosted word vectors' representation capability. However, we can only get one corresponding vector through word2vec embedding for each word in human language, which means this technology cannot handle the polysemy of human language properly. Peters et al.[7] introduced a Bi-directional LSTM, Bi-LSTM for short, as the feature extractor and improved the performance of word embedding on handling the polysemy. At present, the most widely used pre-trained word vectors are mostly generated by BERT[2], which replaces the Bi-LSTM extractor with Transformer[10] and is currently the state-of-the-art word embedding technology across most NLP tasks.

3. Methodologies
In this section, we introduce all the necessary components in our Chinese NER method, including data preparation and the details of our model.

3.1. Data Preparation
The data set presented herein is actual news from Chinese embassies' websites in ten ASEAN countries. This data set contains 1000 news texts and covers political, economic, trade, education, culture, science and technology, military, and other fields. We utilized a crawler script to download the plain texts of news under the constraints of network regulations. After that, a standard data cleaning has been conducted on the downloaded data, such as deleting redundant symbols through regular expressions and manual rules, labelling the title and main body of those news texts, and unifying their format. To label the plan texts effectively, we adopted the THU Lexical Analyzer for Chinese (THULAC)[5] toolkit on the cleaned data set for Chinese word segmentation and generating part-of-speech tagging. Then, we double checked the labels manually to correct errors. In this paper, we force on three entities, i.e., locations (LOC), organizations (ORG), and people's names (PER). For a better performance, we employed the IOBES tag[8], which is a variant of IOB, to do sequence labelling. We present an example of the IOBES tag on our dataset in Fig.1.
Figure 1. An example of sequence labelling done by IOBES on a Chinese sentence, where the corresponding English translation can be written as ‘Bhumibol, the monarch of Thailand.’ B- prefix indicates the first word of the named entity, E- prefix indicates the last word of the named entity, and I- prefix indicates the words that in the middle of the named entity.

3.2. Gated Recurrent Unit

Although Transformer[10] outperforms traditional Recurrent Neural Networks (RNNs) on many tasks, RNNs still plays an important role when dealing with sequence data. The LSTM network is a classical recurrent neural network architecture. It utilizes three gate structure to control the data flow and decide which part of information should be memorized and which part to abandon. Gated Recurrent Unit (GRU)[1] is a simplified version of LSTM, which only has two gates, i.e., the update gate and reset gate. The update gate $z_t$ determines the information that needs to be deleted by the hidden layer at time $t - 1$, while the reset gate $r_t$ determines the amount of information that the hidden layer needs to retain at time $t - 1$. An illustration of GRU is presented in Fig.2.

Give an input $X_t$, the above procedure can be written as follow:

$$z_t = \sigma(W_z[h_{t-1}, X_t]), \quad (1)$$

$$r_t = \sigma(W_r[h_{t-1}, X_t]), \quad (2)$$

$$h = \tanh(W_h[r_t \circ h_{t-1}, X_t]), \quad (3)$$

$$h_t = (1 - z_t) \circ h_{t-1} + z_t \circ h_t \quad (4)$$

where $\circ$ is the element-wise multiplication, and $W_z$, $W_r$, and $W_h$ represent parameters of the update gate, reset gate, and hidden unit, respectively.

Figure 2. An illustration of a GRU neuron at time $t$ with a given input $X_t$. We let $r_t$, $Z_t$ represent the reset gate and update gate, the candidate hidden unit is denoted by $h_t$, and $\sigma$ and $\tanh$ denote the sigmoid and hyperbolic tangent activation functions, respectively.

Such design mitigates the gradient vanishing problem when processing long sequence data and enables a better effect and efficiency than LSTM when processing long sequence data. Besides, because the bi-directional property can benefit models to better understand the input sequence, we employ the bi-directional GRU model to extract semantic information in the pre-trained word vector sequences.

3.3. The overall working pipeline

In this paper, we employ a BERT-base model that is pre-trained on Chinese texts to generate word vectors. For NER, we train a Bi-directional Gated Recurrent Unit (BiGRU) with Conditional Random
Field (CRF) layers to extract features from word vector sequences and identify the named entities. The whole pipeline is visualized in Fig.3.

![Figure 3](image)

**Figure 3.** An illustration of the overall working pipeline of our model.

### 3.4. Text similarity based dataset partition

To make better use of the training data and improve the models' performance, we propose a novel data partition method that is based on a Hash Cosine similarity algorithm, namely HashCos. An intuitive way to mitigate the polysemy is to reduce the similarity of adjacent training examples, which allows the trained model to see the polysemous words as early as possible. Therefore, we combine Hash process and cosine distance to quantify the similarity between text examples. This algorithm includes four steps, i.e., word segmentation, hash process, weighted merging, and calculate cosine similarity. The specific process can be described as follow:

1. As we mentioned in Sec. 3.1, the Chinese word segmentation is done by THULAC toolkit. Besides, we also store the top 20 most frequent words and take their frequency as weights that represent the importance of these words in the text.

2. Based on the segmentation result, we then conduct a hash process on each word, which would transfer words into binary numbers with a pre-defined length.

3. After the hash processing, we merge all hashed results from a text example according to their weight that calculated in (1). For example, assume the weight of the word “ASEAN” is 3, and the hash value of “ASEAN” is “101010”, that is, the string of weighted numbers obtained is [3,-3,3,-3,-3]. As for the whole text example, we accumulate the weighted numbers of top 20 most frequent words. For instance, the weighted number of “asean”, “Thailand” are [3,-3,3,-3,-3]. The weighted merging of these two words is [4,-2,2,2,-2].

4. Finally, the similarity between examples is computed via the cosine distance between the corresponding weighted numbers. In practice, we first randomly shuffle the data set and partition it into three-part by the ratio of 7:1:2, where the largest part is for training, the smallest part is the validation set, and the rest is the test set. For the training set, we randomly select an example as the first example and compute its similarity toward other training examples as described above. Then we sort the rest of the training examples in descending order of their similarity to the selected example.

### 4. Experiments

In the section, we conduct experiments to show the improvement brought by pre-trained BERT word embedding and our HashCos partition method.

In the section, we conduct experiments to show the improvement brought by pre-trained BERT word embedding and our HashCos partition method. All experiments have been done on a NVIDIA Tesla V100 GPU with Tensorflow framework. We employ the same Adam optimizer with initial learning rate 0.001 and set the Dropout rate at 0.6 to prevent overfitting for different experiment setups. For comparing to our HashCos method, we also replaced it with the TF-IDF method to do partition.

The experimental results are summarized in Tab.1, where we evaluate models based on three standard criteria, i.e., precision, recall, and F1, and consider the models' performance on three named entities separately. It can be seen that no matter what kind of word vectors are employed, the models trained on the HashCos partition outperform TF-IDF based one on all named entities and criteria. In addition, we noticed that those organization entities are harder for deep learning to identify than others. It seems that the names of organizations are often abbreviated even in the news, which may increase the
difficulty for deep models to identify them. Overall, it is clear that the combination of HashCos and BERT word vector substantially improves models' performance on all named entities.

Table 1. Comparison between pre-trained word vectors and data partition methods.

| Partition  | Word Vector | Precision rate (%) | Recall rate (%) | F1 score (%) |
|------------|-------------|--------------------|----------------|--------------|
|            | LOC  | ORG | PER | LOC  | ORG | PER | LOC  | ORG | PER |
| None       | word2vec | 93.02 | 86.19 | 95.99 | 94.41 | 86.46 | 94.93 | 94.08 | 87.07 | 94.53 |
| None       | BERT   | 93.75 | 87.68 | 94.13 | 94.28 | 87.15 | 95.35 | 93.14 | 86.67 | 95.17 |
| TF-IDF     | word2vec | 92.56 | 87.26 | 95.41 | 94.21 | 88.06 | 94.71 | 93.21 | 86.06 | 94.71 |
| TF-IDF     | BERT   | 93.13 | 87.99 | 95.49 | 94.24 | 88.16 | 95.73 | 93.17 | 87.99 | 95.49 |
| HashCos    | word2vec | 93.28 | 88.56 | 95.71 | 94.33 | 87.95 | 95.65 | 94.21 | 87.06 | 95.71 |
| HashCos    | BERT   | 94.49 | 89.49 | 96.49 | 94.81 | 88.59 | 95.99 | 94.83 | 89.12 | 96.24 |

5. Conclusions
In this paper, we replaced the word2vec method with BERT and introduced a BERT-BiGRU-CRF model to identify named entities from Chinese ASEAN news. Besides, a similarity-based dataset partition method has been proposed to mitigate the influence of polysemous words. This partitioning method is based on our HashCos algorithm, which can compute the similarity between examples effectively. Experiments show that the proposed HashCos based partition outperforms the TF-IDF based method, and combined it with the BERT-BiGRU-CRF model, our method can improve the model's performance substantially on three different named entities.

In the next step of this study, we wish to expand our method on different sources of languages, such as Thai languages and Malay language and eventually design a NER model that can extract key information from multilingual news.

Acknowledgments
This work was supported by National Natural Science Foundation of China under grant no. 61866008 and Basic and Applied Basic Research Fund of Guangdong Province, China under grant no. 2019B1515120085.

References
[1] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Empirical Methods in Natural Language Processing. Doha, Qatar. pp. 1724-1734.
[2] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.01108.
[3] Chinchor N. (1998) MUC-7 named entity task definition. In: Seventh Message Understanding Conference (MUC-7). Fairfax, Virginia. pp. 1-21.
[4] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. http://arxiv.org/abs/1508.01991.
[5] Li, Z., & Sun, M. (2009). Punctuation as implicit annotations for Chinese word segmentation. Computational Linguistics 35(4), 505-512.
[6] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. http://arxiv.org/abs/1301.3781.
[7] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018) Deep contextualized word representations. In: Stent, A., Ji, H. (Eds), Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics:
Human Language Technologies. Association for Computational Linguistics., New Orleans, Louisiana. pp. 2227-2237.

[8] Ratinov, L., & Roth, D. (2009). Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning. Boulder, Colorado. pp. 147-155.

[9] Sun, M., Shen, D., & Huang, C. (1997). CSeg&Tagl 0: A Practical Word Segmenter and POS Tagger for Chinese Texts. In: Conference on Applied Natural Language. Washington, DC. pp. 119-126.

[10] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems. Long Beach, CA. pp. 5998-6008.

[11] Zhang, Y., & Zhou, J. (2000) A trainable method for extracting Chinese entity names and their relations. In: Second Chinese Language Processing Workshop. Sapporo, Japan. pp. 66-72.

[12] Zheng, Y.B., Xia, Z.C., Guo, Z., Huang, Y.Z. (2018). Named entity recognition of news texts in ten asean countries. Science Technology and Engineering,18(35) :162-168.