Vietnamese Relation Extraction with BERT-based Models at VLSP 2020

Thuat Nguyen and Hieu Man Duc Trong
Hanoi University of Science and Technology, Hanoi, Vietnam
{thuat.nh163964,hieu.mdt161530}@sis.hust.edu.vn

Abstract

In recent years, BERT-based models have achieved the state-of-the-art performance over many Natural Language Language tasks. Because of that, BERT-based model becomes a trend and is widely used for so many NLP task. And in this paper, we present our approach on how we apply BERT-based model to Relation Extraction shared-task of VLSP 2020 campaign. In detail, we present: (1) our general idea to solve this task; (2) how we preprocess data to fit with the idea and to yield better result; (3) how we use BERT-based models for Relation Extraction task; and (4) our experiment and result on public development data and private test data of VLSP 2020.

1 Introduction

Nowadays, Natural Language Processing (NLP) is a very interesting and necessary field of research. The results of the works in the field of natural language processing can bring many benefits to human. As an interesting task in the field of NLP research, the result of Information Extraction (IE) works in general and Relation Extraction (RE) works in particular can help people a lot on automating text processing tasks. However, compared to other popular languages (e.g., English, Chinese), evaluations and research results for Relation Extraction in Vietnamese language are still limited. In this year's international workshop on Vietnamese Language and Speech Processing (VLSP 2020)[1], for the first time, there is a shared task about Relation Extraction in Vietnamese. This is really great as it means that Relation Extraction in Vietnamese is gaining more attention from the research and industry communities. In the Relation Extraction shared task in VLSP Campaign 2020, organizers will release training, development and test data. Training and development data contain Vietnamese electronic newspapers, labeled entity types of all entity mentions in the articles (there are only three types of entity entities) and labeled relations between entity mentions that belong to the same sentence. In the meantime, the test data also contains the similar information contained in the training and development data (newspapers and entity mentions), but will not be provided with the labels of relation between entities. And participating groups are asked to build learning systems based on training and development data, capable of predicting the relationship labels between entities belonging to the same sentence in the test data. And in the next section of this paper, we describe in detail about VLSP 2020 RE task's dataset, how we preprocess the data and about our BERT-based model's architecture that we use for this year's VLSP RE task.

2 Data and Methodology

2.1 Data

All three sets of data (training, development and test) contain files in WebAnno TSV 3.2 File format[2]. Each file only contains one raw document (electronic newspapers) that has not been split into sentences. There are three types of Named Entities (NE): Locations (LOC), Organizations (ORG), and Persons (PER). And four types of relation between annotated entities; three of four relation types are directed and the last one is undirected. These relation types are described in Table 1.

The detailed information is given in the VLSP 2020 RE task's page[3] and the annotation guideline of this task.

[1] https://vlsp.org.vn/vlsp2020/eval

[2] https://webanno.github.io/webanno/releases/3.6.6/docs/user-guide.html#sect_webannotsv

[3] https://vlsp.org.vn/vlsp2020/eval/re

No.	Relation	Arguments	Directionality
1	LOCATED	PER–LOC, ORG–LOC	Directed
2	PART-WHOLE	LOC–LOC, ORG–ORG, ORG-LOC	Directed
3	PERSONAL–SOCIAL	PER–PER	Undirected
4	ORGANIZATION–AFFILIATION	PER–ORG, PER-LOC, ORG–ORG, LOC-ORG	Directed

Table 1: Relation types in the VLSP 2020 dataset.

2.2 General Idea

In this section, we describe our general idea about how we process data:

- We need to split original raw documents by sentences since the dataset contains only pre-labeled relationships between entities belonging to the same sentence.

- Assuming that there are total n entities in a sentence, we create $\frac{n(n-1)}{2}$ sentences corresponding to $\frac{n(n-1)}{2}$ pairs of entities. Each of these sentences is a data point that is passed to our BERT-based model later. The label for each data point is the relation label between the pair of entities in this sentence.

- These are four types of relation. Three of them are directed, so we create new two undirected relations for each directed relations, depending on whether the directed relation label is on the preceding or following entity in the sentence. See below EXAMPLE I and EXAMPLE II for more clarity.

  EXAMPLE 1: In the sentence: "Hà Nội là thủ đô của Việt Nam", the relation between two entities ("Hà Nội" and "Việt Nam") is PART-WHOLE. This relation label is on the "Việt Nam" entity, which is the entity that comes after in the sentence. We set this data point's label to PART-WHOLE.

  EXAMPLE 2: In the sentence: "Việt Nam có thủ đô là Hà Nội", the relation between two entities ("Hà Nội" and "Việt Nam") is PART-WHOLE. This relation label is on "Hà Nội" entity, which is the entity that comes first in the sentence. We set this data point's label to WHOLE-PART.

- There are many entities in the same sentence but there are no relations between them, so we create a new type of relation called "OTHERS" for them.

- Finally, we pass these data points into our BERT-based model.

In the end, we have a total of seven types of relations.

2.3 Preprocessing data

This section presents details on how we preprocess data. Because the dataset contains only pre-labeled relationships between entities belonging to the same sentence. So we need to split original raw documents by sentences. To do that, we try to use two of the best libraries out there for Vietnamese language processing: VnCoreNLP[4] (VNC) and Underthesea[5] (UTS). In our own experiment, Underthesea seem better to us when compared to VnCoreNLP:

- VNC has problems with Unicode normalized: "Thanh Thủy" will be "Thanh Thuỷ". While UTS seem to have better Unicode normalized.

- VNC has problems with splitting a correct sentence into two sentences. While UTS seems or very rarely has this problem. It is quite hard for us to fix this problem.

- VNC can split sentences perfectly by some characters like single dot, three dots ... while UTS sometimes does not split sentences by these characters. However, we can find, and fix these sentences easily.

Besides, there are some other small problems when we use these two libraries. But results from Underthesea seem to be better than results from VnCoreNLP. So we decide to use Underthesea for preprocessing data.

We follow the following steps to preprocess data:

- Normalize data with "NFC" form.

[4] https://github.com/vncorenlp/VnCoreNLP

[5] https://github.com/undertheseanlp/underthesea

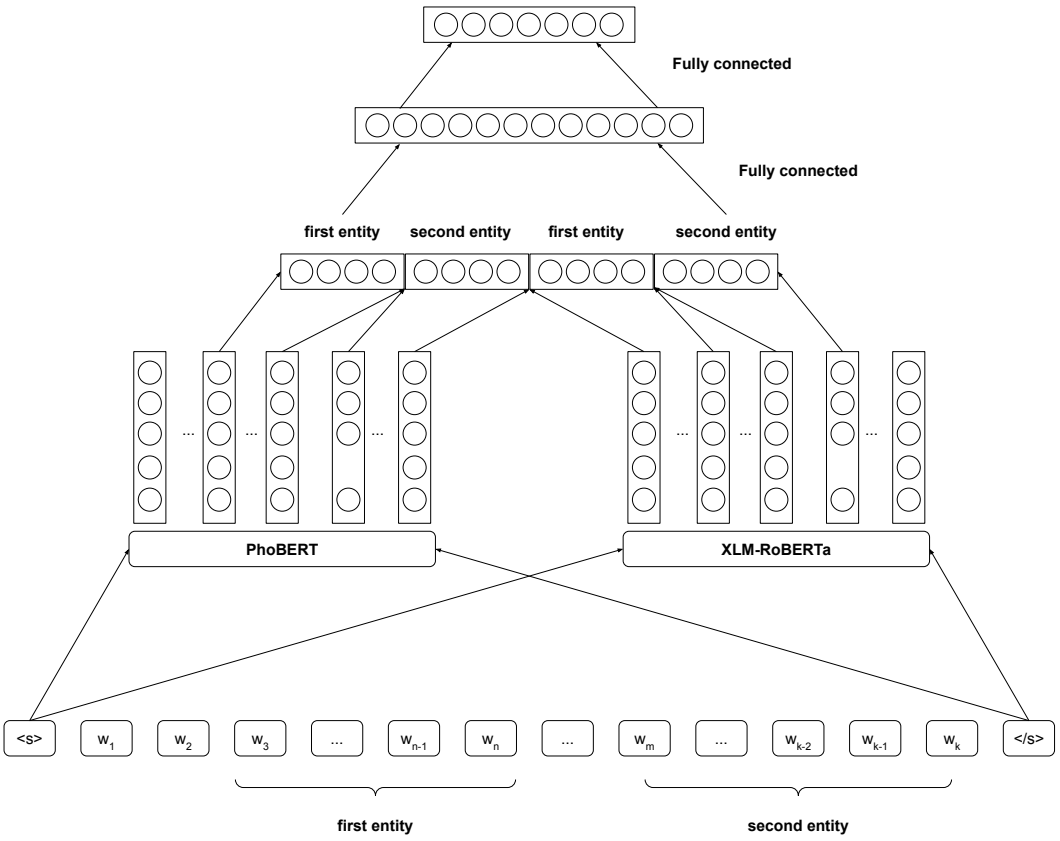

Figure 1: Our BERT-based model for Relation Extraction.

- Using Underthesea to split raw documents to sentences.
- Find and review sentences contain characters like: dot, three dots. However, these characters are not the ending characters of these sentences. Then if there are mistakes in these sentences (two different sentences are combined into a single sentence), we will split these sentences using rules.
- Split sentences by colon punctuation using rules.
- Remove characters that are not alphanumeric (either alphabets or numbers) at the beginning or at the end of an entity.
- Fix the problem with faulty Word Segmentation of Underthesea.

Besides, we also do some other preprocess steps like: Check and fix if there is a relation between entities belonging to different sentences to make sure that data extracted from raw data is correct.

2.4 BERT-based model

In this section, we present our BERT-base model's architecture. We use two BERT-based models that support Vietnamese language: PhoBERT (PB) (Nguyen and Tuan Nguyen, 2020) and XLM-RoBERTa (XLMR) (Conneau et al., 2019). We use these two BERT-based models to generate embedding vectors for each pair of entities of each sentence. Then we combine (using pooling methods) these embeddings into one single embedding vector, and pass it into a multi layer neural network with seven (the number relation types) units and Softmax activation function in the last layer. The architecture of our model is shown in Figure 1.

About details, we follow the following steps to process sentences:

- We pass sentences into the BERT-based models to generate embedding vectors for each pair of entities of each sentence. We try to use both of two BERT-base models PB and XLMR; we also try to use only PB or only XLMR.

- In particular, each entity may have multiple word pieces. So with each entity's word pieces, we try to use and combine embeddings of it from different BERT layers to only one single embedding vector for that word piece. We tried several combinations like: concating embeddings from the last four layers, element wise max pooling embeddings from the last two layers.

- Then, with each entity, we do the same process like each entity's word pieces to generate only one single embedding vector for an entity from its word pieces embedding vectors.

- Each sentence has two entities, so we have two embedding vectors. Let the first entity's embedding vector be h_1; the second entity's embedding vector be h_2. From these two vectors, we generate one single embedding vector for the current sentence: $[h_1, h_2]$.

- Each PB and XLMR model have its own final sentence embedding. In the combination model of PB and XLMR, we concatenate two sentence embedding of these two models to obtain one single sentence embedding vector.

- Finally, we pass the final sentence embedding vector to a multi layer neural network with seven (relation types amount) units and Softmax activation function in the last layer.

3 Experiments and results

In our experiments, we try to use only one of the two BERT-based models (PB or XLMR) and compare with using both models, but using both models always gives much better results. We use Google Colab[6] GPU for training. Since the maximum GPU memory of Colab is 16GB, our biggest model is a combination of fine-tuned PB base model with non fine-tuned XLMR Large (Model 1). We found that if we fine tune PB with high epoch numbers (about 8) and with small learning rate of E-05 can give results that are close to the best we have ever had. And the model results seem more stable when using average pooling instead of using max pooling.

[6] https://colab.research.google.com

Model	Development data	Test data
Model 1	**93.23**	71.19
Model 2	93.10	69.30
Model 3	93.09	**72.06**

Table 2: The performance of the models (Micro-averaged F-score) on the public development data and the private test data.

Each participating team can submit three final results on the test set. The official evaluation measures are micro-averaged F-score. So we choose three models that have the highest micro-averaged F-score on the public development data. Details of the results (on both public development data and private test data) are presented in Table 2.

All of our three best models using PB base and XLMR base model, with PB base is fine-tuned with a learning rate of E-05. Our worst model on the development data (Model 3) give the best result on the private data. We think that two other models may too overfit on the training data tuning on public development data.

With results in Table 2, we achieved the best result with Model 3, ranking the 1st of the scoreboard on the private test set of Relation Extraction shared-task at VLSP 2020 campaign.

4 Conclusion and Future Work

In this paper, we have presented our approach to solve the Relation Extraction task proposed at the VLSP Shared Task 2020. We find out that the BERT-base model is actually really good, since our models are quite simple but achieve acceptable results. In the future, we want to use better GPU to train bigger models like fine tuned PB large with fine tuned XLMR large, since bigger models seem to have better results.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. PhoBERT: Pre-trained language models for Viet-

namese. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.