**DEVELOPMENT OF MULTI-TASKING MODEL IN NATURAL LANGUAGE PROCESSING FOR VIETNAMESE**

1 Vietnam-Korea Information Technology and Communication University, Vietnam

2 Olinno Technology Ltd., Vietnam

# **Abstract.** (huy)

In recent years, due to the increasing demand for searching, exploring, and processing Vietnamese information, issues related to Vietnamese language processing have been receiving significant attention from the research community both domestically and internationally. Part-of-Speech (POS) tagging and Named Entity Recognition (NER) are among the crucial steps in processing and mining Vietnamese data. This report presents and implements solutions for the tasks of POS tagging and NER based on the VLSP (Vietnamese Language and Speech Processing) dataset. These results will contribute to guiding the development of an effective POS tagging system for the Vietnamese information mining community in general, and Vietnamese language processing in particular. Using VLSP-2016 dataset, which contains more 14000 Vietnamese sentences,we build a multitask NLP model.To do that, we use PhoBERT as encoder of the model and fine-tune while training. After that, we use Conditional Random Field model to decode and predict the tag.. For the evaluation, we test the model on more than 2700 sentences.

**Keywords: Name entity recogition, Part of Speech**

# 1 Introduction(huy)

Nowadays, technology is rapidly advancing, and artificial intelligence applications are increasingly being integrated into daily life. Keeping up with this trend, tasks related to natural language processing are continuously developing and being applied in practice, particularly for the Vietnamese language. In the process of Vietnamese language processing, the identification of Part-of-Speech (POS) tags and Named Entities plays an immensely crucial and beneficial role in various fields, ranging from natural language processing, machine learning, to artificial intelligence. Part-of-Speech tagging is the process of labeling each word in a sentence with corresponding tags such as noun, verb, adjective, adverb, and so on. With this information, computers can comprehend the grammatical structure of the sentence, thereby gaining simple understanding of its meaning. POS tagging greatly supports many natural language tasks, such as syntactic analysis, machine translation, speech synthesis, information extraction, and various other applications. On the other hand, Named Entity Recognition (NER) is the process of identifying and categorizing components with special meanings in a text, such as names of people, locations, organizations, time expressions, quantities, currencies, and more. This process helps computers understand the context and extract specific information from the text. Consequently, it facilitates easier information extraction, text summarization, sentiment analysis, as well as the construction of information systems, automatic response systems, and other artificial intelligence applications. we use a fairly well-known approach recently, which is the Transformer model, a model that uses a attention mechanism to help capture the relationship of words with each other, while reducing the speed of computation.In addition, to ensure the correct data format with Vietnamese, we also use segmentation instead of tokenizer method. For the decoding part, we use the Conditional Random Field model,a simple model that computes fast but is efficient in sequential prediction tasks such as POS and NER, which helps to make the outcome predictions for POS and NER. In summary, we present the proposed method as follows:

i) We define a data format to feed to PhoBERT model

ii)) During training , we fine-tune PhoBERT to gain the best result.

iii) After that, we use Conditional Random Field to model the probability of the relationship between words, and use the Viberti algorithm to decode and predict the tags for NER and POS.

# 2 Preliminary(lộc)

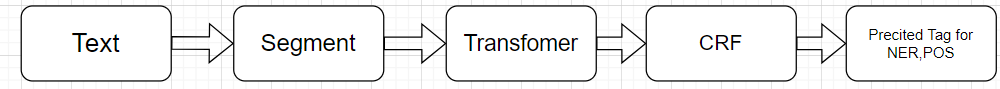
In order to have a good language model, there must be a good enough coding and embedding model, and in Vietnam, there are many individuals and organizations that already have their products. In addition,the dataset is very important in NLP, since this is a multi-tasking model, we need a text data set with parallel labels for all words and phrases. About the decoder part, we need a solution, algorithm or model that is reliable enough and ensures computation speed. Our first approach is to use LSTM for encoding and embedding, however, we soon realized that this approach had many flaws:

+Long Training Times: Training LSTM models can take a long time, especially on large datasets. The recurrent nature of LSTMs makes it challenging to parallelize computations, which can result in slower training times compared to other architectures.

+Difficulty Capturing Long-Term Dependencies: While LSTMs are designed to handle long-term dependencies better than vanilla RNNs, they can still struggle to capture very long-term dependencies. Extremely long sequences can lead to memory issues and make it challenging for LSTMs to retain relevant information over extended periods.

+Lack of Parallelism: LSTMs are inherently sequential models, making them difficult to parallelize across time steps. This limitation can be a drawback when training on GPUs, which excel at parallel processing.

To solve those, we use a fairly well-known approach recently, which is the Transformer model, a model that uses a attention mechanism to help capture the relationship of words with each other, while reducing the speed of computation.In addition, to ensure the correct data format with Vietnamese, we also use segmentation instead of tokenizer method. For the decoding part, we use the Conditional Random Field model,a simple model that computes fast but is efficient in sequential prediction tasks such as POS and NER, which helps to make the outcome predictions for POS and NER. We present the step in the figure blow.



# 3. Vlsp-2016 and data format for model (lộc)

To build a model that can perform both POS and NER tasks at the same time, it is necessary to have a dataset whose labels include both. In addition, the data must be at least large enough to cover the complex grammar of Vietnamese. Also, if possible, sentences in this data should be segmented first, which will reduce the time it takes to reprocess the data. So that we chose the VLSP-2016 dataset, it includes about more 14000 thousand Vietnamese sentences already segmented, and words and phrases labeled POS and NER.

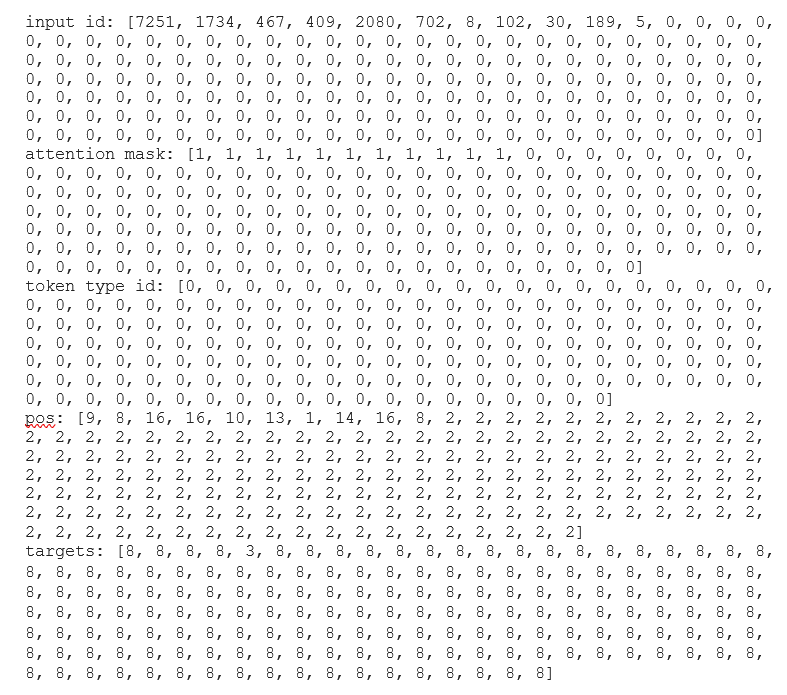
The next step is to prepare the format for model training. To do this, we declare a python class that returns a dictionary, which contains 5 lists, they are of equal length and equal to the number of words and phrases of the longest sentence in the data set. And because the lengths of sentences are unequal, for shorter sentences, we will insert elements that represent a word or symbol that has no meaning,this will apply to all 5 lists.Here are those:

+A list containing the encodings of words and phrases in a sentence, these encodings will be used with an embedded weight matrix for embedding and use.

+A list containing attention mask elements, representing actual elements (words and phrases), and elements that should be ignored (padding elements), for real elements, will represent 1, otherwise it will be 0.

+A list contains types of tokens , which can be understood to distinguish when one sentence ends and another begins, because we only use 1 sentence at a time, so the whole list will be 0.

+And finally 2 lists containing the labels of POS and NER for the sentence.Their positions for the encoded words are the same.In the figure below , here an example of the format for a sentence



# 4. Main component in model(lộc)

# 4.1 POS and NER

## 

Part-of-Speech (POS) tagging is the process of assigning a label to each word in a sentence based on its grammatical role, such as noun, verb, adjective, adverb, preposition, or conjunction. POS tagging is crucial in Vietnamese natural language processing as it helps computers understand grammar and semantics, enabling syntactic analysis, semantic prediction, and dependency graph construction. It also benefits machine translation, information retrieval, text data analysis, and information summarization, enhancing accuracy and efficiency in NLP applications.

In this report, the team proposed to implement labeling for the following types of entity labels:

| STT | **Code word type** | **Description** | **For example** |
| --- | --- | --- | --- |
| 1 | N | Noun | bàn, ghế, lợn, gà, ngày, tháng,... |
| 2 | Np | proper noun | Tên người, tên địa điểm |
| 3 | Nc | classifier noun | cái, con, chiếc, cục, cây (cột, nến, rơm), lá (gan), quả (tim, cật), tờ, tập, tệp, mẩu, bánh, mảnh, miếng, cuốn, quyển, pho, ngôi, toà, túp, căn, thửa, súc, suất, ngọn (gió), bông, đoá, cái (tình), cái (tát), cái (vỗ vai), (ngã một) cái, (đùng một) cái... |
| 4 | Nu | Measure and Monetary Unit | watt, jun, ha, cm, mm, kg, m², m³, cân, yến, tạ, tấn, lít, độ, hào, đồng, xu, quan, đôla, nhân dân tệ, yên, bảng, ..... |
| 5 | Ny | Abbreviation of Noun | Nv (nhân viên), a (anh), e (em),… |
| 6 | A | Adjective | tốt, xấu, to, nhỏ,... |
| 7 | C | Conjunction | và, hoặc, nhưng, nếu, thì, vì, nên |
| 8 | CH | Symbol | - @ # $ % & \* ! < > ( ) { } [ ] ... |
| 9 | E | Preposition | của, để, từ, đến... |
| 10 | FW | Foreign words | Nhãn này thường được gán cho những từ ngữ, những cụm từ hay câu tiếng nước ngoài (như tiếng Anh chẳng hạn) được trộn mã hay chuyển mã vào tiếng Việt: made, in, Thailand (made in Thailand), anyway, and, or, but, I, love, you, too (I love you too), how, are, you (how are you)…. |
| 11 | I | Interjection | ái chà, ôi, chao ôi, trời ơi...; haizzz, hihihi |
| 12 | L | Quantifier | các, những, vài, đôi, dăm, mọi (người), mỗi (lần), từng (ngày), chút (thời gian), một\_vài, một\_số, dăm\_ba, mỗi. tất\_cả, cả, mọi, cái (ba cái con gà này), một (có một người đang)... |
| 13 | M | Number of words (Numeral) | một, hai, trăm, nghìn, 3, ½, 2018, 20/10, 30/4/1975, ... |
| 14 | P | Pronoun | tôi, tao, mày, nó, ấy, bao\_nhiêu, ai, kia, gì, nào, vậy, thế, sao,... |
| 15 | R | Adverb | đã, sẽ, đang, rất, lắm, hết\_sức, vô\_cùng, có\_lẽ... |
| 16 | T | Auxiliary | à, ư, nhỉ, nhé...; ngay, cả, chính, đích\_thị, chỉ, những,… |
| 17 | V | verb | muốn, đi, chơi, ăn, uống,…. |
| 18 | Vy | Abbreviation of Verb | Kt (kiểm tra), nc (nói chuyện),… |
| 19 | X | Undetermined group | x² = a, x+y = 36, √A² = │A│, A1 x √A², 9X {người thế hệ những năm 1990}, 2ker {người thế hệ những năm 2000} |
| 20 | Z |  |  |

## 

Named Entity Recognition (NER) plays a crucial role in natural language processing for the Vietnamese language. The task of NER is to identify and classify named entities in the text, including names of people, locations, organizations, dates, amounts, and other significant entities.

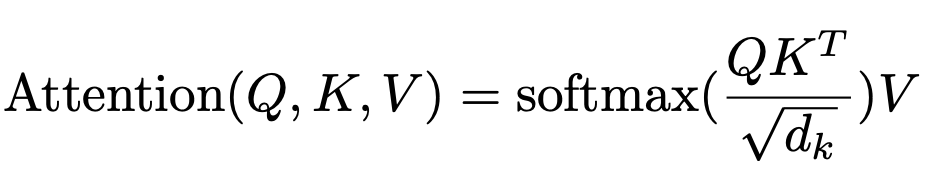
In this report, the team proposed to implement labeling for the following types of entity labels:

|  | Label | **For example** |
| --- | --- | --- |
| 1 | Persons: | - First, middle and last names of people, animals and fictional characters, aliases |
| 2 | Organizations: | - Companies(press agencies, studios, banks, stock markets)  - Government bodies (ministries, councils, courts, political unions of countries (e.g. the {\it U.N.}))  - Public organizations (schools, universities, charities)  - Other collections of people (sports clubs, sports teams, associations, theaters companies, religious orders, youth organizations) |
| 3 | Locations: | - Roads (streets, motorways)  - Regions (villages, towns, cities, provinces, countries, continents, dioceses, parishes)  - Natural locations (mountains, mountain ranges, woods, rivers, wells, fields, valleys, gardens, nature reserves, allotments, beaches, national parks) |
| 4 | MISC | - The MISC label is used to mark up ambiguities between the country name( Location) and meaningful names belonging to that country  - Used to markup the name of the work, the name of the event containing the name of the person, the geographical name (the content of the announcement do not directly talk about people's names, geographical names)  - Used to markup product names, applications with brand names |

# 4.2 Transformers (lộc)

In natural language processing, to represent the relationship between words,people try to embed words on numerical vector space which are called word vectors. Word vectors are used to capture their context relationships. In other words, embedding is the encoder part of a language model, it decide how good the model will be.

The most popular embedding method is transformer, it uses self-attention mechanisms, which allow the model to selectively focus and incorporate the complex relations on different parts of the input sequence. Transformer models typically consist of an encoder and a decoder where the encoder processes the input sequence using self-attention and feed-forward neu-ral networks, producing a series of encoded representations from each input sen-tences. The decoder then uses the encoded representations and self-attention to generate output sequence for the trained language tasks such as a machine trans-lation. The attention mechanism is the heart of Transformer based language models as follows:



We use PhoBERT,a language model specifically designed for Vietnamese text. It is based on the transformer architecture and is pre-trained on a large corpus of Vietnamese text data. It consists of many components:

+Embeddings layer:It takes 3 inputs: word\_embeddings,position\_embeddings,token\_type\_embeddings. It is used to represent words and their contextual information in a dense vector spaces

+Layer Normalization and Dropout: LayerNorm: Layer normalization is applied after each layer in the model. It helps stabilize the training process and speeds up convergence. dropout: Dropout is applied with a probability of 0.1 during training. It helps prevent overfitting by randomly setting some activations to zero, forcing the model to learn more robust representations.

+RobertaSelfAttention: This component computes the self-attention scores between the input words. It consists of three linear layers for the query, key, and value transformations. The self-attention mechanism helps the model understand the dependencies between different words in the sequence.

+RobertaSelfOutput: After the self-attention mechanism, this module processes the attention output by applying a dense layer, layer normalization, and dropout

+RobertaIntermediate: This component applies a dense layer to transform the self-attention output to a higher-dimensional space (3072 dimensions) before applying the activation function (GELU) for non-linearity. +RobertaOutput: The intermediate output is transformed back to the original dimension (768 dimensions) using another dense layer, followed by layer normalization and dropout.

# 4.3 CRF(Huy)

CRF is a method for building probabilistic models that help segment and label sequential data or identifiable entities. Those entities represent words in sentences representing objects such as names of people, organizations, places, etc.

In CRF, the input data is sequential and we have to rely on the previous context to make predictions about a data point. We use feature functions with multiple input values,:

+Label of data point − 1 in

+Label of data point in

+Set of input vectors

+Location of the data point we are predicting

Feature functions in CRFs capture the relationships between input observations and output labels in sequence labeling tasks. They encode relevant information about the input data and are used to compute the conditional probabilities necessary for predicting the labels:

Each feature function is based on the label of the previous word and the current word. To build the conditional field, we next assign each feature function a set of weights (lambda values), which the algorithm is going to learn:

With and n is number of tokens

To estimate lambda parameters, we use the Maximum likelihood estimation method, apply to the linear negative log function of the above function to make the partial derivative easier to calculate:

Calculate the partial derivative with 𝜆 to find the minimum value of the log function alone because by finding the value argmin will achieve the maximum value for the whole negative log function:

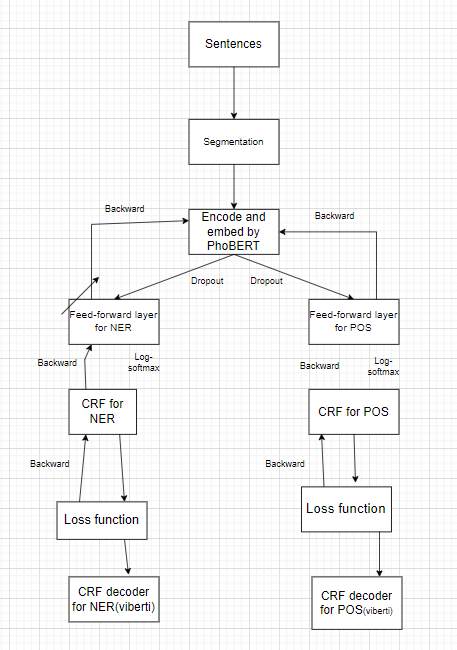
With

We use the Partial Derivative as a step in Gradient Descent. Gradient Descent updates parameter values iteratively, with a small step, until the values converge. Our final Gradient Descent update equation for CRF is:

# 4.4 train model and fine tune phobert

After preparing the data format as in Section 3, we proceed to train the model. The structure of the model is as follows: phobert will recognize the data and perform the embedding, then the embedding matrices will be passed to 2 feed-forward layers for POS and NER with dropout of 0.3, with their output equal to the number of layers of POS and NER. Next, the output of the two feed-forward layers will be transformed into two CRF models, and will use log-likehood as the loss function and the Viberti algorithm to decode the prediction labels.

During training, we also fine-tune the PhoBERT weights for best results. The weights of the feed-forward layers and the 2 CRF models are also updated. We used AdamW optimizer, a learning rate of 0.0003 and a weight decay of 0.001,weight decay is a method to reduce overfitting and exploding gradient, by adding the multiplication of squares of all the parameters with a small number(usually 0.1, 0.01 or 0.001) to our loss function, and batch\_size is 32, we also perform model evaluation during training with evaluation data of more than 2700 sentences. Combining training and evaluation reduces time and can better track model improvement.



# 5. Results

# 6. Conclusions

So we've finished building a model that can do 2 tasks together and get pretty good results, thanks to the pretty good VLSP-2016 dataset, and the PhoBERT and CRF models. With the segmentation API, we can use this model for most Vietnamese text and obtain acceptable results.

In the future, we plan to look at cases where the model gives misleading results, to review and create a new dataset that includes those cases,also, the new dataset will include many types of labels for the NER task and to find ways to improve the decoder by combining with other techniques with CRF.