MINISTRY OF EDUCATION AND TRAINING

VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

GRADUATE UNIVERSITY OF SCIENCE AND TECHNOLOGY

……..….*……**

NGUYEN THI THU HIEN

RESEARCH ON TEXT NORMALIZATION AND NAMED ENTITY RECOGNITION METHODS IN VIETNAMESE SPEECH RECOGNITION

Major: Information System
Major code: 9 48 01 04

SUMMARY OF COMPUTER DOCTORAL THESIS

Ha Noi – 2023

The thesis has been completed at the Graduate University of Science and Technology- Vietnam Academy of Science and Technology

Supervisor 1: **Assoc. Prof. Dr. Luong Chi Mai**
Supervisor 1: **Dr. Nguyen Thi Minh Huyen**

Reviewer 1: Assoc. Prof. Dr. Ngo Xuan Bach
Reviewer 2: Dr. Do Van Hai
Reviewer 3: Assoc. Prof. Dr. Nguyen Phuong Thai

The thesis shall be defended in front of the Thesis Committee at Vietnam Academy Of Science And Technology - Graduate University Of Science And Technology, at ……. hour……, date…… month…….year 2023

This thesis could be found at:
- The Library of Graduate University of Science and Technology
- The National Library of Vietnam

INTRODUCTION

Natural Language Processing (NLP) is a field of computer science that combines artificial intelligence and computational linguistics to process human-computer interactions so that computers can understand or imitate human language. NLP consists of two major branches which include speech processing and text processing.

One of the important tasks in understanding the meaning of written or spoken text is Named Entity Recognition (NER). Numerous studies have achieved promising results for NER with regular written text in many languages worldwide, including Vietnamese. However, research on Named Entity Recognition for the output of Automatic Speech Recognition (ASR) still poses its own challenges compared to written text, and there are very few works available for Vietnamese in this area so far.

Speech recognition refers to a process that converts speech signals of a particular language into a sequence of corresponding content words in text format. The output text of ASR often lacks structure, such as punctuation, capitalization of the first word in a sentence, or proper names and place names. This leads to difficulties in comprehension and limits the exploitation of ASR output in most applications. Therefore, NER from the ASR output exhibits distinctive characteristics as it frequently contains recognition errors, especially for entities that are often outside the dictionary. ASR errors often occur within words that form named entities or in the context of those words, directly affecting the performance of NER.

Furthermore, NER systems also need to deal with issues related to missing important cues like capitalized words and punctuation marks. To enhance the ASR output, *text normalization* is necessary. After processing, the final text will have a better structure and be more understandable compared to the raw ASR output, resulting in higher effectiveness when deployed in real-world applications. Therefore, the development of text normalization and NER solutions from ASR output is essential to improve the overall quality of ASR systems.

However, the normalization of ASR output, particularly the punctuation and capitalization restoration, still faces several issues that need improvement. Besides the significance in enhancing the quality of ASR output, punctuation and capitalization are also crucial pieces of information for the NER task. It can be observed that not all capitalized words in Vietnamese are considered named entities. Conversely, named entities are not necessarily comprised of entirely capitalized words or phrases. Moreover, named entities can be classified into various types. Therefore, the restoration of punctuation and

capitalization is one of the critical factors that contribute to optimizing the NER system in ASR output texts.

In fact, there are many NER processing methods for ASR output text but mainly focus on resource-rich languages such as English, Chinese. Until now, there are very few studies that apply NER to Vietnamese ASR. Also, these studies have only focused on short conversational texts. From those challenges, the I chose the topic *"Research on Text Normalization Methods and Named Entity Recognition in Vietnamese Speech Recognition"* for my Doctorial Degree.

Research targets and tasks: The thesis focuses on proposing solutions and implementing experiments for two goals. The first is to normalize the output text of the Vietnamese ASR system by restoring punctuation and capitalization. The second goal is the NER on the output text of the Vietnamese ASR system.

Research contents: The dissertation investigates the peculiarities of data and output errors in Vietnamese ASR systems, and explores fundamental issues of the NER problem, as well as the challenges. After that, I construct a dataset for training and evaluating modelsto restore punctuation and capitalization to normalize the ASR output of Vietnamese text. The NER task for the ASR output in Vietnamese is addressed through two approaches: pipeline and end-to-end.

Research scope: The research will focus on addressing issues related to processing ASR output with long and challenging spoken text. In addition to the text normalization problem of ASR output, the study will concentrate solely on designing a model to predict punctuation and capitalization, assuming the ASR system achieves a word error rate (WER) of 0%. As for the NER problem-solving model, the dissertation utilizes a real ASR system with a WER of 4.85%.

Research methods: The thesis has carried out theoretical research, including an overview of the problems to be solved, the methods and techniques, and their effectiveness. On that basis, I propose solutions to overcome some outstanding problems. The dissertation focuses on implementing experimental methods to measure and evaluate proposed models to solve problems, and compare them with other methods. Regarding experimental data, building datasets of text combined with corresponding speech.

Contributions: Construction of text-and-speech datasets for training and evaluation of normalized models and identifier entities for output text of ASR systems. These data are described in the works [CT1, CT2, CT4, CT6]; Proposing and improving the punctuation and capitalization recovery model to help standardize the output text of Vietnamese ASR. This model has been

introduced, evaluated, and improved in the works [CT2, CT3, CT5]; Proposing two solutions for identifying entity identifiers in output documents of Vietnamese ASR according to Pipeline and E2E approaches. These solutions are presented and evaluated in the works [CT4, CT6].

Layout: In addition to the introduction and conclusion, the dissertation is structured into 4 chapters. Chapter 1 provides an overview of the research issues. Chapter 2 - Fundamentals, presents the foundational knowledge used for guiding and proposing the normalization and NER models for the ASR output text. Chapter 3 introduces the problem of restoring punctuation and capitalization for Vietnamese ASR systems. Finally, chapter 4 proposes a method for NER in the ASR output of Vietnamese through two approaches: pipeline and E2E.

Chapter 1. RESEARCH ISSUES OVERVIEW

This chapter will present an overview of NLP, and difficulties in Vietnamese language processing. A general understanding of the ASR system, the features in the output text of the ASR system, and the research related to the normalization of the output text of ASR help support the NER model. At the end of this chapterI am going to describe the NER problem, the difficulties in processing NER for Vietnamese speech, and related studies.

1.1. Natural language processing

NLP is a subfield of computer science that combines artificial intelligence and computational linguistics. Tools such as sentiment analysis, NER, syntax analysis, semantics, etc., have made NLP a popular research topic in various fields, such as machine translation, information extraction, text summarization, automatic question answering, and more. Numerous NLP applications on smart devices have emerged, gaining significant attention from the community.

NLP can be broadly divided into two main branches: speech processing and text processing. The problem of text processing after speech recognition poses a challenge that needs to be addressed. The dissertation also raises the issue of normalizing the output text of Vietnamese speech recognition and named entity recognition.

1.2. Automatic speech recognition

1.2.1. Introduce

Automatic speech recognition definition was firstly stated by Yu and Deng as follows: *"it is a term used to describe processes, technologies and methods that enable better human-computer interaction through translate human speech into text format"* [3].

The most commonly way to evaluate the performance of an ASR system is using the WER metric, which is based on the Levenshtein

distancemeasuring the number of insertions, deletions, and replacements in a string:

$$WER = \frac{I + D + S}{N} * 100 \quad (1.1)$$

Where *I, D, S, and N* are the number of insertions, number of deletions, number of replacements, and number of words in the text, respectively

For the Vietnamese ASR system, the syllable error rate (SyER) was used instead of the word error rate to evaluate the performance of the ASR system (the VLSP's model)

$$SyER = \frac{S + D + I}{N} \quad (1.2)$$

where *S, D, I, C* is the number of substitutions, number of deletions, number of insertions, number of syllables, and number of syllables in the text, respectively

1.2.2. Processing the output text of the speech recognition system

The output text of ASR often exhibits distinct characteristics, differing from regular written text, especially in Vietnamese including: The lacks punctuation and capitalization; Foreign proper nouns and abbreviations may not be accurately recognized; Numeric and currency formats are recognized as continuous phrases, violating standard rules; Vietnamese incorporates many loanwords from other languages to create new words; Insertions, deletions, and substitutions of words are common. Therefore, the ASR output text needs to be normalized to create a well-structured and more understandable final text, in order to enhancethe usability of ASR text in various NLP applications.

1.3. Normalize the output text of speech recognition

1.3.1. The problem of restoring punctuation and capitalization

Capitalization involves accurately identifying the form of a word, distinguishing among four types: all lowercase, all uppercase, only the first letter of each syllable capitalized, and mixed case, which includes a combination of uppercase and lowercase letters. *Punctuation* restoration is the task of inserting punctuation marks into appropriate positions within a text that lacks any punctuation for a meaningful sentence. On the other hand, the rule of capitalizing the first letter of the first syllable in a complete sentence demonstrates the relationship between capitalization and punctuation, implying that both tasks need to be handled simultaneously. However, research often focuses on solving specific tasks separately. Clearly, processing individual tasks alone cannot effectively improve the output quality of ASR. As a result, there has been a growing trend of research that integrates both tasks. Even with integrated processing, determining whether to restore punctuation or capitalization first is

also an issue, as the processing order may affect each other and the final results [13].

1.3.2. Process methods

One of the early implementations of the auto-capitalization method was rule-based, that is, using the principle of determining the beginning of a new sentence to indicate which character should be capitalized [17]. Studies have shown that rule-based systems are difficult to maintain because they may constantly require the addition of new rules. A language model is a probabilistic model that helps predict the next word in a sequence of words. The linguistic model calculates the probability of a given word w_k in the context of *n-1* previous words w_{k-1}, w_{k-2}, ..., $w_{k-(n-1)}$. This probability can be expressed by $P(w_k| w_{k-1}, w_{k-2}, ..., w_{k-(n-1)})$. Studies on punctuation recovery and association modeling [19] based on the *n-gram* language model have been proposed. The disadvantage of the *n-gram* model is lack of evaluation the entire sentence context, so in many cases, it is not possible to give an exact probability. Furthermore, today's computing resources in terms of storage and processing capabilities, models with high *n* numbers are still difficult to handle due to their storage requirements. According to the researchers, capitalization or punctuation can be considered a sequential labeling problem. Given a string $W=w_0w_1w_2...w_n$, the model predicts the capitalization string $C=c_0c_1c_2...c_n$ where c_i with all lowercase, first capitalization, all capitalization, or mixed capitalization. Similarly, predict punctuation $E=e_0e_1e_2...e_n$ where e_i indicates a punctuation mark or no punctuation at all. Some studies use the Maximum Entropy model [21], the Hidden Markov model [22], and the Maximum Entropy Markov Model. [23] for both missions. The Conditional Random Field is also a probabilistic model used to segment and label sequence data [24]. CRF has advantages over MEMM and other Markov models because CRF is an undirected graph model, allowing CRF to define the probability distribution of the whole state. However, most of the studies that restore punctuation, capitalization, and lowercase often use a combination of CRF in the last layer of the neural network architecture.

Recently, studies have used neural network architecture for the problem of recovering punctuation and capitalization. Susanto et al. [27] proposed a model using a Recurrent Neural Network at the character level to handle bias in mixed capitalization cases. RNNs have proven useful in modeling sequential data. Tilk et al. [28] used a two-dimensional recurrent neural network model with an additional hidden layer that allows data processing in the opposite direction more flexibly than traditional RNNs, combined with an attention mechanism to restore punctuation. This model is able to use long contexts in both directions and directs attention when necessary, allowing for better performance on former English and Estonian data sets. Since 2017, with

the introduction of Transformer architecture [29], different versions of BERT [30], RoBERTa [31] have opened up many new research directions. Rei et al. [32] applied the capitalization recovery of video subtitles generated by the ASR system using the BERT model. This approach is based on pre-trained contextual word encoding and applied fine-tuning. This method proves superior to other approaches not only in performance but also in calculation time. Alam's research group [33] has tested several Transformer models such as BERT, RoBERTa, ALBERT, DistilBERT, mBERT, XLM-RoBERTa for English and Bangla language. For English, the best results are observed on the RoBERTaLARGE model when recovering well, but the efficiency of handling commas and question marks is relatively low. The lower Bangla performance compared to English is easily explained due to the lack of resources for training.

The research problem of restoring punctuation and capitalization for Vietnamese speech output is still relatively new. Recently, Thuy Nguyen and colleagues [34] have experimented with a BiLSTM deep neural network model, while Hieu Dinh and colleagues [35] used a Transformer model for punctuation restoration. The initial recent research mainly focused on addressing the issue of capitalization restoration separately. Subsequently, integrated models were developed to simultaneously restore punctuation and capitalization for Vietnamese speech. These recent research findings for Vietnamese are considered a basis for further studies aiming to normalize the output text of Vietnamese ASR systems for specific purposes.

Specifically, Uyen *et al.* [13] have just proposed the JointCapPunc model architecture for punctuation and capitalization restoration using a hierarchical structure, where capitalization restoration was firstly performed and followed by the punctuation restoration layer. However, a pre-trained Transformer language model with a large number of parameters can introduce significant delays in the Pipeline model. Additionally, the research data was conducted on short conversational segments within the medical domain.

1.4. Named Entity Recognition

1.4.1. Define

Sundheim and Grishman introduced for the first time at the MUC-6 conference [36]: *"Named entity recognition is a deterministic process of searching for meaningful words or phrases from classified natural language documents into predefined unique groups such as: name of person, name of organization, date and time, location, number, currency…"*.

1.4.2. Challenge for research problem

Vietnamese does not have ASR output text data labeled with standard NER large enough for training and evaluation. The challenges for the NER problem in the output text of the Vietnamese ASR include: In the output

documents of the ASR, capitalization is omitted, making it difficult for the recognition system. In addition, the problem in sentences that don't exist of any kind of punctuation is really a difficult one and it's not easy to segment or parse sentences correctly. Also, determining the boundary of a Vietnamese word is more difficult than in other languages owing to subjection to an isolated language, that is, a word can be made up of one or more languages. Proper names also pose certain challenges for the NER system. Due to the loose constraints on proper names, it is possible for the system to ignore or mistake it for another entity. In particular, the ASR error causes identifier entities to be missed and formatted entities to be misrecognized. If one or more of the words constituting the identifier entity is misrecognized, it is difficult to recognize the correct identifier entity. Conversely, even if all the words constituting the identifier entity are correctly recognized, the correct identifier entity may not be recognized due to the lack of context in the output text of the ASR. Foreign names, and abbreviations in the ASR output text can also be recognized in different versions. The phenomenon of homophones with different meanings is more common in Vietnamese than in Indo-European languages.

1.4.3. Research situation

1.4.3.1. Pipeline approach

Kim *et al.* [42] firstly proposed identifying entity identification method on the output text of ASR based on rule set. The advantage of the method is that it requires small storage, and the rules can be expanded. However, the disadvantage is that the rules need to be built manually. Especially when the input is the output text of the ASR, the capitalization information for the identifier entity therefore is no longer available, so getting the language information is not possible, while the language needed to formulate laws will be difficult. To overcome this disadvantage, many machine learning-based studies have been proposed such as HMM model [43], ME [44], CRF [45], [46]], HMM-CRF [47], support vector machine (SVM) [48] and focus mainly on English, Chinese, Japanese, French. Studies also show that it is necessary to incorporate more syllable features, combine punctuation and capitalization information, and improve errors in the output text of ASR to increase NER performance.

Recently, researches on NER focus mainly on deep learning platform owing to its outstanding advantages in vector representation, computational ability, non-zero mapping ability, etc. linearity from input to output, the ability to learn latent semantic information of large dimensions, and the ability to train E2E.

It can be seen that, with the Pipeline approach, the NER component has to deal with a denormalized text like normal text and contains noise, thus having

a great impact on the NER performance [52]. This approach will be affected by the output text error of the ASR and the error propagation through each step.

1.4.3.2. E2E approach

Ghannay *et al.* [53] proposed the first test of an E2E-oriented French speech-based entity recognition method. They proposed a deep RNN architecture model, consisting of *nc* convolutional layers (Convolutional Neural Network - CNN), followed by *nr* one-way or two-way repeating layers Gated Recurrent Unit or Long Short Term Memory, a search Lookahead Convolution and a fully connected layer immediately preceding the Softmax layer. The system is E2E trained using the CTC-loss function [10] to predict the character sequence from the input audio. The experimental results show that the E2E model is still less efficient than the Pipeline incorporating the POS (Part of Speech) feature used to label the ASR output before processing NER and that POS is really important. for the NER task. Caubriere *et al.* [54] implemented E2E based on the DeepSpeech2 system with an architecture consisting of a 2D-invariant convolutional two-layer stack, five biLSTM layers and a final softmax layer. The system also uses the CTC-loss function that allows the association between the input audio and the output character string. Compared with the best results of the ETAPE evaluation campaign, the proposed E2E system showed a relative improvement of 4%. This approach also did not achieve better performance than the Pipeline method that proposed by the same author in the same study. According to Chan *et al.* [55], when experimenting with the Pipeline model, the proposed Pipeline model using BERT for pre-training still achieves higher performance than E2E and argues that, although the modules in the Pipeline can be affected by error propagation, they can still take advantage of pre-training to increase performance, especially when the ASR system is well improved.

The E2E model has not proposed the better performance compared to the Pipeline model. However, when the amount of training data is large enough, the systems need to move towards building E2E models. This helps to optimize the training process, all the parameters of the model are trained at the same time, the errors generated between the components are calculated thereby minimizing the error propagated through each model- heat. Training and inference using the E2E model is simpler as well as more convenient to put the model into application. Even so, E2E model design will require a high degree of integration of component models into a common model, bypassing the intermediate stages, making the design process more difficult. At the same time, it requires advanced model training algorithms such as weight tying, multitasking learning, etc.

1.5. Data Overview

To serve the training and evaluation purposes of the ASR output text normalization model in Chapter 3, a large dataset of text samples is constructed, which is unformatted.

For the NER problem-solving model, following the pipeline and E2E approaches in Chapter 4, the labeled text and audio dataset are leveraged from the VLSP 2018 NER text dataset. Corresponding to this standard text set is a set of unformatted text and recorded audio data with various reading voices in different environments. To reduce recording costs, all VLSP text data will be utilized with Google's TTS system to generate synthetic audio data. Subsequently, the synthesized audio dataset will pass through VAIS' ASR system to obtain the text dataset for training the NER E2E model.

Further details regarding the datasets will be described specifically in Chapter 3 and Chapter 4.

1.6. Conclusion of Chapter 1

Chapter 1 provided an overview of NLP and the challenges in processing the Vietnamese language. NER is an important problem in NLP, but it faces several difficulties in ASR output texts. Therefore, research on ASR output text features, the issues to be addressed, and an overview of related studies for normalizing ASR output text were presented. Besides introducing the basics of the NER problem, its significance, and system evaluation methods, the research also highlighted the challenges of NER in the Vietnamese ASR output and related studies, aiming to identify the areas that need to be addressed. Additionally, an overview of the datasets, specific feature datasets will be introduced in Chapters 3, 4.

Chapter 2. FUNDAMENTAL KNOWLEDGE

Chapter 2 provides a detailed presentation of several deep learning models for sequence processing, word representation, and sequence labeling. This foundational knowledge is crucial for guiding the proposal of normalization and named entity recognition models for the Vietnamese ASR output text in Chapters 3 and 4. Additionally, Chapter 2 introduces the multi-task learning method, which will be applied in Chapter 4 to design an end-to-end NER model.

2.1. Sequence Processing Models

2.1.1. GRU

It is challenging to capture long-range dependencies using traditional RNN models due to the vanishing or exploding gradients with long sequences. To address this issue, the GRU model [73] was proposed. The main difference between the regular RNN and GRU lies in GRU's support for hidden state control, meaning it includes mechanisms to decide when to update and when

to reset the hidden state. Meanwhile, the GRU model reduces the number of gating signals to two compared to the LSTM model. These two gates are called the update gate and the reset gate. Despite its advantages, GRU also has some limitations when processing very long sequences. It may lose important information during sequence processing, and it still has limitations in modeling due to complex relationships within sequences. Moreover, it requires a large number of parameters for training, increasing the demand for training data and computational resources.

The introduction of the Transformer model has brought a new breakthrough, enabling effective processing of various tasks while addressing some drawbacks of RNN and its variants like LSTM or GRU. In this thesis, the Transformer model has been applied in designing the normalization model for Vietnamese ASR output text in Chapter 3.

2.1.2. Transformer

The Transformer is a deep learning model that utilizes attention mechanisms to calculate the influence of input variables on the output results. This model has been widely used in the field of NLP, but recently, it has been developed for other applications such as computer vision and speech processing. Similar to other machine translation models, the overall architecture of the Transformer model consists of two main parts: the Encoder and the Decoder. In the Transformer model, the Encoder is responsible for processing the input and representing words or sentences as meaningful vectors while, the Decoder's task aims to transform the input representations into an output sequence. The Transformer model uses multiple encoding and decoding blocks to process data. Each block includes a multi-head self-attention layer and a feedforward neural network. The multi-head self-attention layer allows the model to learn multi-dimensional representations of sentences, while the feedforward neural network learns non-linear representations at each position.

Self-attention: It is an important mechanism in the Transformer model, allowing the model to determine the importance of words in a sentence by calculating a weight for each word based on its correlation with other words. This helps the model understand the semantic and syntactic relationships within the sentence.

Multi-head attention: In the Transformer model, each self-attention layer uses multi-head attention. This mechanism enables the model to learn multi-dimensional representations of the sentence by calculating attention from multiple different representation spaces, helping it capture complex relationships within the sentence. Using multi-head attention allows the model to learn various aspects of the sentence and provides richer representations for the input data.

2.2. Word Representation Models

2.2.1. Word2Vec

Developed by Tomas Mikolov and colleagues at Google in 2013, Word2Vec is a word vector representation technique used to address advanced NLP problems. It can be trained on a large corpus of text to learn the relationships or dependencies between words. Word2Vec establishes semantic relationships between words by predicting the current word based on the surrounding context or vice versa. The output of Word2Vec is word vector representations that can be utilized in various machine learning models [69].

Word2Vec provides two neural network-based variants: Continuous Bag of Words (CBOW) and Skip-gram. CBOW predicts the current word based on its surrounding context. The input to CBOW is a window of surrounding words, and the objective is to predict the current word. On the other hand, Skip-gram attempts to predict the context surrounding the current word based on the current word. Skip-gram takes the current word and predicts the words in its surrounding context. Once word vector representations are extracted from the Word2Vec model, they can be used to perform tasks in NLP.

When dealing with large amounts of data and the need for models to learn complex word representations, capturing correlations between words in a sentence, understanding the context-specific meanings of words, and generating appropriate representations, deep learning models become more suitable. With the advent of the Transformer model, many new variations have emerged. The thesis improved the BERT model for Vietnamese data when proposing the NER model.

2.2.2. BERT

BERT is a deep language model introduced by Jacob Devlin and colleagues at Google Research in 2018. The BERT model has a deep learning architecture that uses multiple layers of Transformer encoders. However, the difference in BERT method is it uses two types of word representations: input word representations and output word representations [71].

BERT is a novel method for pretraining word vector representations. One distinct feature of BERT that can not be found in previous word vector representation models is the ability to be fine-tuned. When BERT is fine-tuned for a specific task, the pretrained Transformer encoder acts as an encoder, and a randomly initialized classifier is added on top. In the case of NER, the classifier is simply a projection from the size of words to the size of the label set, followed by a Softmax operator to convert the scores into label probabilities.

2.3. Label Models

2.3.1. Softmax

Softmax is a commonly used activation function in multi-class classification models to convert the network's output into a probability

distribution. It is typically applied to the final output layer of the model to compute the predicted probabilities for each class. The softmax function is continuous and differentiable, which is very useful for calculating derivatives to update the weights during neural network training. Using softmax is not only beneficial for multi-class classification tasks but can also be applied in other problems, such as determining the confidence level of predictions or generating a probability distribution from input values. However, softmax has some limitations. When the number of classes is large, computing and processing the exponential values simultaneously can become complex and computationally expensive. Additionally, softmax is not robust to noise, meaning that if there is significant variation in the input values, the output probability values can easily become biased, leading to inaccuracies in predictions.

2.3.2. Conditional Random Fields

CRF were proposed by Lafferty in 2001. It is a probabilistic undirected graphical model that combines the characteristics of Hidden Markov Models and Maximum Entropy Models. CRF is a special case of a random Markov field, which addresses the label bias problem caused by Hidden Markov Models. Additionally, contextual features can be taken into account to select better features. CRF is used to compute the conditional probability distribution of a set of output random variables based on a set of input random variables. The goal of training a CRF is to learn the parameters of the feature functions to maximize the log-likelihood function of the training data. This can be achieved using maximum likelihood estimation or other optimization methods.

2.4. Multi-Task Learning

Humans might learn multiple tasks simultaneously. During the learning process, humans can use the knowledge acquired from one task to learn another task. Taking inspiration from the human learning ability, Multi-Task Learning (MTL) aims to jointly learn multiple related tasks so that the knowledge contained in one task can be leveraged by other tasks with the hope of improving the overall generalization performance of all tasks [76].

According to Zang *et al.*, MTL is defined as follows: *"With m tasks, where all tasks or a subset of them are related, Multi-Task Learning aims to learn these m tasks together to improve the learning of the model for each task by utilizing the knowledge present in all or some tasks."* [77]. In deep learning, two common methods for MTL are hard parameter sharing and soft parameter sharing [78].

In many cases, a model is only concerned with the performance of a specific task. However, to take advantage of MTL, auxiliary tasks can be added to further improve the performance on the main task. These auxiliary

tasks are assumed to be related to the main task in some way and useful for predicting the main task.

With the assumption that the punctuation and capitalization restoration model can provide additional information, better support, and enhance the efficiency of NER, the thesis leverages the knowledge of MTL methods and auxiliary tasks to propose a NER model for the output text of Vietnamese ASR in an E2E approach.

2.5. Conclusion of Chapter 2

Chapter 2 presented foundational knowledge about word representation techniques. It provided detailed descriptions of the characteristics and architectures of several sequence processing models. Additionally, labeling models were introduced. Furthermore, the chapter covered the special focus on the methods of hard parameter sharing, soft parameter sharing, and auxiliary tasks in MTL. The models introduced in this chapter serve as the groundwork for building models for standardization and NER for the output text of Vietnamese ASR, as presented in Chapter 3 and Chapter 4.

Chapter 3. NORMALIZATION OUTPUT TEXT OF THE VIETNAMESE SPEECH RECOGNITION SYSTEM

Chapter 3 presents the problem of restoring punctuation and capitalization in Vietnamese speech output documents, the difficulties and limitations when performing this task, and then propose solutions and ways to build data, model setting and experimental results. Research results on two approaches have been published in the works [CT2], [CT3], [CT5].

3.1. Research problem and solution

Input: output text of Vietnamese ASR system

Output: text restored with punctuation, capitalization

Research scope: About data: Research on restoring punctuation and capitalization on long speech texts. About punctuation: Focus on the restoration. There are three types of punctuation marks: period, comma, and question mark. About capitalization: Distinguish the two main labels, which are lowercase and capitalization, do not handle labels such as mixed or all capitalization.

Solution: Propose an approachto segment the input string and merge the output, taking into account the context of the words surrounding the cut. Design a deep learning model to incorporate recovery of punctuation, capitalization. Building a dataset for research purposes from official Vietnamese online newspapers, with an error rate of 0% in the text.

3.2. Model architecture

The processing model is carried out according to the following steps: (1) The output text of the Vietnamese ASR will be passed through the overlapping segmentation module to cut the input string. (2) The punctuation and capitalization recovery model will take the trimmed segments and process them in parallel and generate an output punctuation label list. (3) Finally, use the overlap segment merge module to merge the labeled output corresponding to input text.

3.2.1. Proposed processing of input string split and output string merge

3.2.1.1. Chunk-overlap split

For the overlapping segmentation module, the proposed workaround is to split the input sequence into segments of fixed size, with the overlap taking up half of the segment length. The form can be described as follows: The selected split length is an even number of words. Let l be the length of the split, k is the length of the overlap, then we have $l = 2k$. Each string from input S containing n words denoted $w_1, w_2,..., w_n$ will be cut into overlapping segments, where the i^{th} is a substring of words $[w_{(i-1)k+1}, \ldots, w_{(i+1)k}]$. In the study, the values of l, k were investigated and selected experimentally.

3.2.1.2. Chunk-overlap merging

Let c be the segment length to keep or discard in the overlapping segments. For simplicity of calculation, take $c = \lfloor k/2 \rfloor$. It can be observed that the last words of the first overlap and the first words of the second overlap (the words surrounding the cut) will not have much context. Therefore, the algorithm will remove the segment at the end of the overlapping segment (1) and keep the segment at the overlapping segment (2). Accordingly, the remaining words at the beginning of the overlapping paragraph (1) are kept and the remaining words at the beginning of the overlapping paragraph (2) are discarded. This ensures that the words in the anti-overlapping part are always kept in the middle of the paragraph, there will be more context to help the recovery more accurate. The discard and retain segments of the overlapping sections will be repeated for the next overlapping segments. The merge section describes as follows:

$$[w_1, \ldots w_{2k-c}] + \sum_{i=2}^{n-1}[w_{(i-1)k+c}, \ldots w_{ik+c}] + [w_{n-2k+c}, \ldots w_n] \qquad (2.1)$$

3.2.2. Design a deep learning model for recovering punctuation, capitalization

Figure 3.1 introduces the proposed CaPu model for the problem of recovering punctuation and capitalization for Vietnamese ASR output text, including components: Word Embedding, Transformer Encoder and CRF.

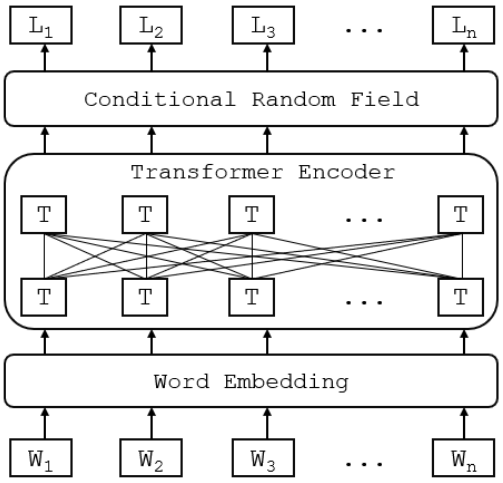

Figure 3.1: Proposed CaPu model for output text of Vietnamese ASR

3.3. Data construction

In order to have the output text data source of Vietnamese ASR large enough, the $Text_{CaPu}$ dataset was collected from Vietnamese electronic news sites including *vietnamnet.vn*, *dantri.com.vn*, *vnexpress.net*. The $Text_{CaPu}$ dataset is divided into the $Text_{CaPu\text{-}train}$ for training set, the $Text_{CaPu\text{-}vl}$ evaluator, and the $Text_{CaPu\text{-}test}$ for test set. The $Text_{CaPu\text{-}train}$ will be converted to lowercase and remove the punctuation to simulate the output of ASR, keeping the number, date, and word errors in the text. Table 3.1 provides information on the number of labels for each type of punctuation and capitalization and lowercase in the training and test datasets.

Table 3.1: Dataset information

Label	**Training dataset (*)**	**Test dataset (*)**
U	15.400	74
L	69.300	507
$	76.600	525
.	2.700	24
,	5.300	30
?	53	2.6

(*)Unit: 1.000

3.4. Model setting

The research has conducted experiments on LSTM, Transformer models and the proposed new Transformer Encoder - CRF model. The models are built based on the Fairseq library [71]. LSTM and Transformer are encode-decode models. Each model has two encoding layers, two decoding layers and has the same hidden layer size of 1024. Another difference of Transformer from LSTM is that Transformer has a number of attention peaks of 4. For comparison under the same conditions, Transformer Encoder - CRF also has a

number of encoding stages of 4, with each layer has 4 attention vertices and has the same hidden size of 1024. The embedding size for all three models is 256. The experiment was trained on NVIDIA 2080Ti GPU with the corpus includes 85 million words , and the random paragraph size is 4 to 22 words.

3.5. Experimental results

3.5.1. Review of using overlap merging

Figure 3.2: Results of using and not using overlapping segment merge models

Figure 3.2 shows a comparison chart with the results of LSTM, Transformer, Transformer Encoder - CRF models with different segment sizes, with and without overlapping fragment merge cases. As can be seen, models using overlapping segment merge always give better results. In particular, in the proposed model Transformer Encoder – CRF, the highest result of using merge is 0.88. The results confirm the hypothesis of the study that adding more context by stacking overlapping segments and segments, merging overlapping segments may improve the model.

The study presents the results for the proposed Transformer Encoder - CRF model with or without overlapping fragment merge and also only statistics in the labels ('U' '.' ',' '?') , omitting the labels ('L' '$'), since the exact count is many, it is not necessary for an efficient comparison. Table 3.2 shows the superiority of the overlapping segment merge method when the F1 scores across all classes improved significantly from 0.01 to 0.05. The results show that words in the middle of the overlap provide the model with more predictive information, and the merging process can select the appropriate part of this overlapping region.

Table 3.2: Comparison of Transformer Encoder - CRF model results with and without overlapping merge

Model	**Label**	**Precision**	**Recall**	**F1**
Transformer Encoder-CRF with chunk-overlap merging	U	0.90	0.86	0.88
	.	0.71	0.57	0.63
	,	0.66	0.53	0.59
	?	0.75	0.52	0.62
Transformer Encoder-CRF with non chunk-overlap merging	U	0.89	0.85	0.87
	.	0.69	0.54	0.61
	,	0.65	0.50	0.57
	?	0.74	0.47	0.58

3.5.2. Evaluate ciphertext and raw text output

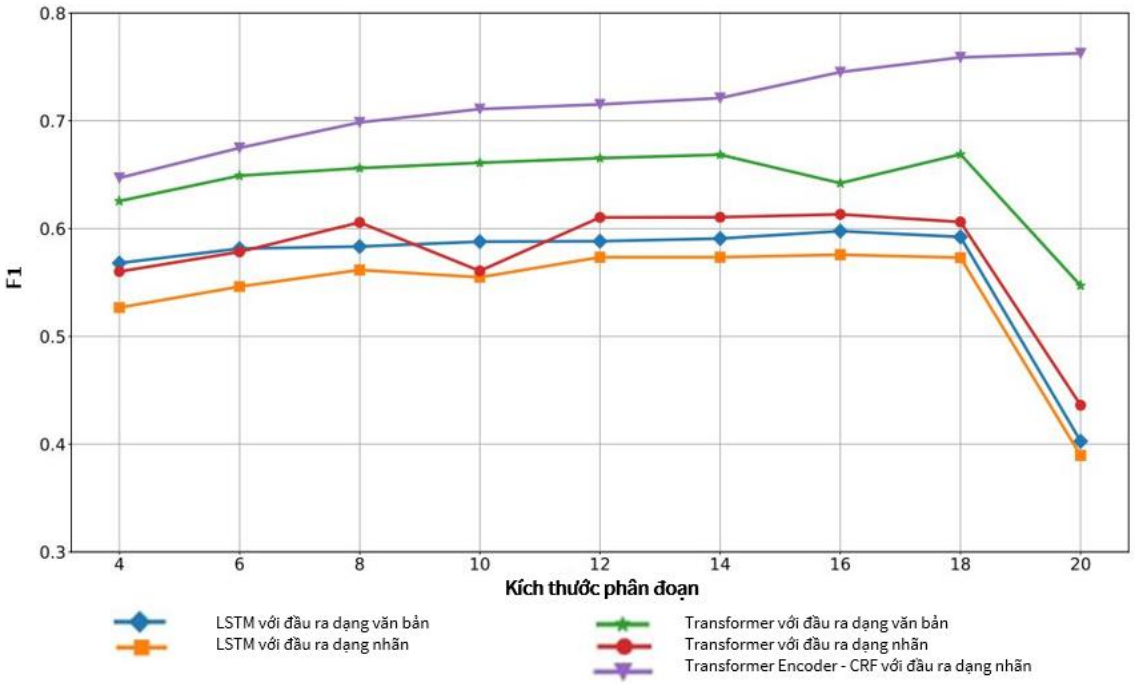

Figure 3.3: Model results with text or label output

Models using the usual text and label output are compared in Figure 3.3. LSTM and Transformer models with regular text have better results than labeled output and suggested appropriate model. At the same time, the error matrix in Figure 3.4 also reveals the percentage of true/false predictions of punctuation and capitalization for the Transformer Encoder - CRF recommendation model. The ability to correctly restore lowercase, capitalization and unsigned is very high (0.86-0.99), then decreases with periods, commas, and question marks.

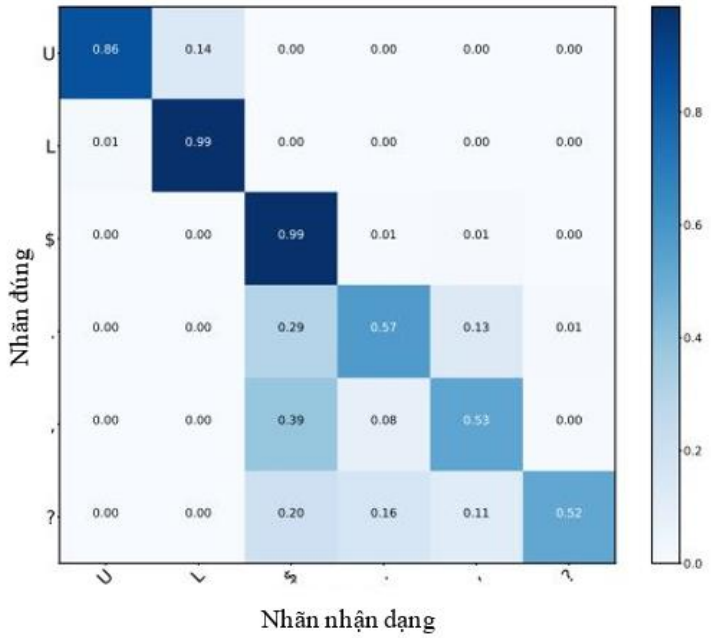

Fig ler – CRF model

3.5.3. Speed rating

The results of the comparison of the execution times of the 3 models with coded output text and plain text are shown in Table 3.3 with 2080 ti (GPU), batch_size: 128. With ciphertext output, the model has faster processing time than plain text. The ciphertext output even shows outstanding performance when it is used with the proposed model.

Table 3.3: Speed rating (tokens/second)

Output	**Transformer**	**LSTM**	**Transformer Encoder - CRF**
Label	$263s \rightarrow 2209t/s$	$217s \rightarrow 2678t/s$	$90s \rightarrow 6457t/s$
Plain Text	$355s \rightarrow 1637t/s$	$230s \rightarrow 2526t/s$	-

3.6. Conclusion Chapter 3

Chapter 3 has built a combined model of Transformer Encoder-CRF for the purpose of restoring capitalization and punctuation for output text of Vietnamese ASR. The main contribution of the study is to propose a solution to chunk and merge overlapping paragraphs. Under the same conditions as the Transformer model, the Transformer Encoder – CRF provides a significantly smaller number of parameters, thereby increasing processing speed.

Chapter 4. NAMED ENTITY RECOGNITION FOR OUTPUT TEXT OF VIETNAMESE SPEECH RECOGNITION

Chapter 4 details the NER problem and proposes models, constructs the data, and implements the experiments to evaluate and compare solutions according to Pipeline and E2E approaches. The Pipeline approach assumes that combining a punctuation and capitalization recovery model such as the CaPu model will provide useful information as input to help the NER model achieve higher performance. The E2E approach is a complex process, which making the system more convenient to operate, avoiding errors propagated through the steps, and simultaneously solving both problems of restoring punctuation, capitalization and named entity recognition. Research results on two approaches have been published and listed in the works [CT4], [CT6].

4.1. Problem statement

Input: Output text of Vietnamese ASR

Output: Named entity recognition with Pipeline and E2E approach

Research scope: About data: Long text, large vocabulary. The ASR system for evaluation has a WER of 4.85%. About identifier entity: Identifies three main types of entities namely person, organization and place.

Research direction: Building a suitable data set. For the Pipeline approach, this study proposes approach to incorporate the CaPu model into the system with the aim of improving the performance of the NER model. The E2E approach, which using CaPu module pre-training for the model also be discussed. Study also has proposed NER architecture using deep learning models.

4.2. Data construction

4.2.1. Training dataset

The first dataset, $Text_{CaPu}$, is a large dataset taken from Vietnamese online newspapers, removing punctuation and capitalization formatting and assigning punctuation and capitalization for model training purposes, standardization of the output text of the ASR system; *The second dataset*, $Text_{ViBERT}$, is a ViBERT model training dataset collected from many domains on the Internet; *The third dataset*, $Text_{VLSP}$, is the NER labeled text dataset of VLSP 2018. This standard text set is used to train the NER model by the Pipeline approach; *The fourth dataset*, $Text_{VLSP\text{-}TTS\text{-}ASR}$, is the dataset to train the NER model according to the E2E approach. First, speech data is synthesized from the training text of the NER VLSP 2018 dataset using Google's TTS system. Then, it is passed through the VAIS ASR system to obtain the ASR output text.

4.2.2. Test dataset

Both the Pipeline and E2E approaches use a dataset recorded by four voices read in different environments from the VLSP 2018 NER test dataset with 26 hours of audio. Then, this audio data set is put through VAIS's ASR system (with WER equal to 4.85%) to receive the output text data set of ASR, $Text_{VLSP\text{-}Audio\text{-}ASR}$ for evaluation purposes. proposed models. Simultaneously, the standard VLSP test dataset $Text_{VLSP\text{-}test}$ or the VLSP dataset de-formatted $Text_{VLSP\text{-}UnCaPu}$, is also used to evaluate and compare the model under different input conditions.

4.3. Named entity recognition by Pipeline approach

4.3.1. Proposed model

4.3.1.1. General model

Proposed general architecture of NER system in the direction of Pipeline:

(1) The ASR system will convert the speech signal into text.

(2) Next, through the CaPu model, the output text of ASR will be restored with punctuation and capitalization.

(3) Finally, from the CaPu model, the information of the entities is labeled using the NER model.

4.3.1.2. Model restore punctuation, capitalization

The study also hypothesizes that the CaPu model will support increasing the performance of the Vietnamese NER model. The proposed model and

experimental results have been detailed in Chapter 3 of the thesis, and published in works (CT2), (CT3), (CT5).

4.3.1.3. Deep learning model for NER

The study proposed to use RoBERTa architecture [31] and train on Vietnamese corpus to create a pre-trained language model. Due to the limitation of computational power, the training model has reduced the number of hidden layers, the number of attention peaks and the word size from the base architecture model RoBERTa and named ViBERT. Figure 4.1 depicts the design of the NER model, in which, ViBERT is used to embed the input sentence, the two-dimensional GRU model and the CRF layer are attached to the ViBERT head to classify the entity label of each input word.

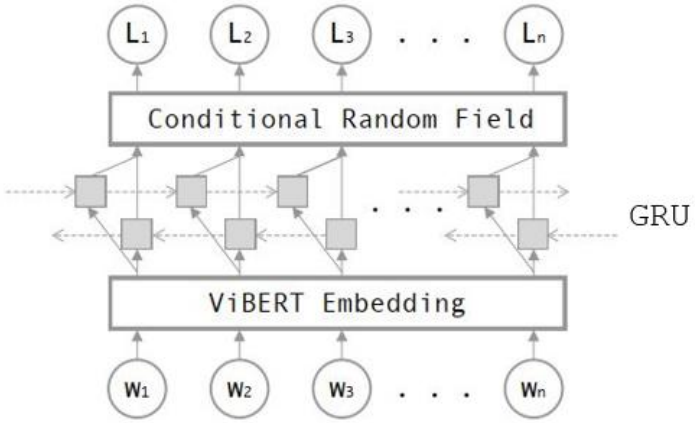

Figure 4.1: Proposed NER model

4.3.2. Result of evaluation

In the NER model, the study combining ViBERT with GRU and CRF layers shows that the efficiency when producing F1 results is 0.9018, significantly higher when compared with previously published results. Table 4.1 indicates the results directly evaluated using the $Text_{VLSP\text{-}test}$ dataset of NER VLSP 2018

Table 4.1: Evaluation of NER models based on the 2018 NER VLSP dataset

Model	**F1**
Vi Tokenizer + Bidirectional Inference [76]	0.8878
VNER [77]	0.7752
Multi layers LSTM [76]	0.8380
CRF/MEM + BS [76]	0.8408
ViBERT+GRU+CRF (Our proposed model)	**0.9018**

With the word error rate of the ASR system of 4.85%, Table 4.2 concluds that if the output text of the ASR is fed directly into the NER model, the entity recognition efficiency will decrease from 0.9018 to 0.6389. The importance of capitalization and punctuation is also observed in the test running the NER

model on text with punctuation and capitalization omitted. In this case, the F1 score drops from 0.9018 to 0.7535.

Table 4.2: Evaluation of the proposed NER model according to the Pipeline approach with different input text types

Input	F1
$Text_{VLSP\text{-}test}$	0.9018
$Text_{VLSP\text{-}Audio\text{-}ASR}$	0.6319
$Text_{VLSP\text{-}Audio\text{-}ASR}$ + CaPu	**0.6713**
$Text_{VLSP\text{-}UnCaPu}$	0.7535
$Text_{VSP\text{-}UnCaPu}$ + CaPu	**0.8141**

Table 4.2 also demonstrates the effectiveness of the CaPu model in improving the accuracy of the NER model. The F1 score of the NER model increased from 0.6319 to 0.6713 when applying this model on the output text of ASR and improved over 0.06 points of the NER model when applied to the text without punctuation and capitalization.

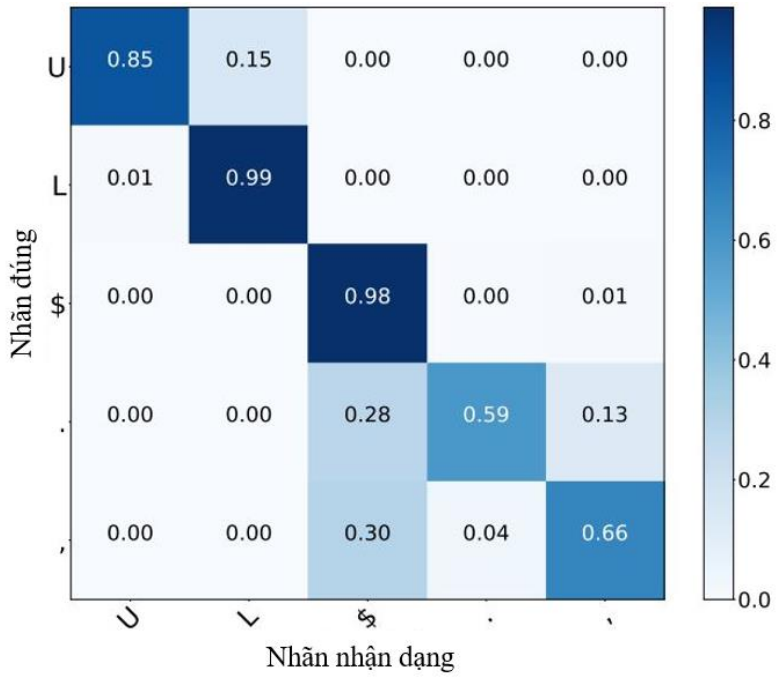

Figure 4.2: Evaluation of the CaPu model on standard text remove punctuation and capitalization letters

4.4. End-to-end named entity recognition

4.4.1. Proposed model

Figure 4.3 shows the proposed E2E model, which consists of a main NER stream based on a Pipeline structure combined with a flow of punctuation recognition, capitalization plays the role of supplementing information about for the recognition stage. The data input into the model is the output text of the Vietnamese ASR without punctuation and no capitalization of length *N*. During the recognition process, some sentences appear errors such as

substitution, insertion, and deletion, causing errors to occur identification process. The input sentence is passed through the Vietnamese language representation set ViBERT. In this study, the transfer learning approach is applied to the ViBERT model which is pre-trained and preserved in the E2E model proposed here. The output of ViBERT is a matrix of size *(Nx768)* which is a matrix representation of the input sentence.

This representation matrix is fed to three blocks simultaneously: (1) block for extracting punctuation auxiliary information, upper case, (2) NER recognition block, and (3) punctuation recognition auxiliary block. CaPu flower according to multi-task learning mechanism.

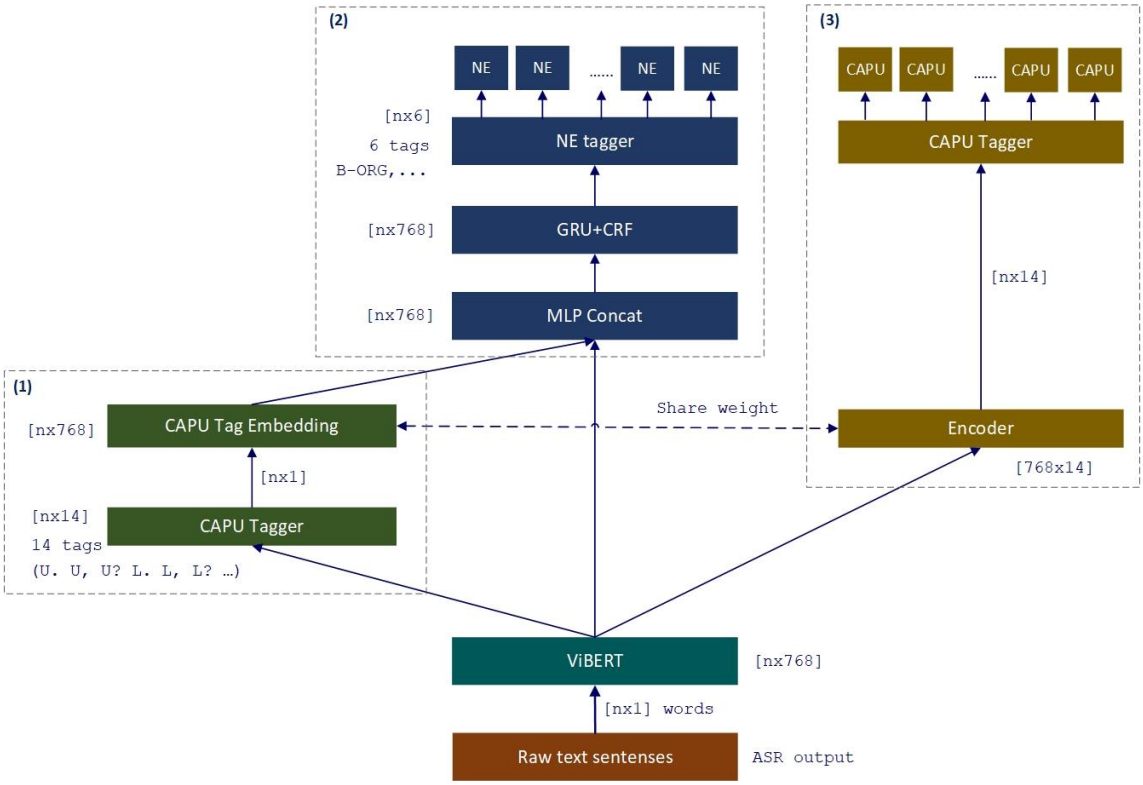

Figure 4.3: Proposed NER architecture according to E2E approach

4.4.2. Experimental results

Table 4.3 shows that if the output text of ASR is fed directly into the NER model, the entity recognition results will decrease from 0.9018 to 0.6319.

Table 4.3: Evaluation of the proposed NER model according to the E2E approach with different input text types

Input	**F1**
$Text_{VLSP\text{-}test}$	0.9018
$Text_{VLSP\text{-}Audio\text{-}ASR}$	0.6319
$\boldsymbol{Text_{VLSP\text{-}Audio\text{-}ASR}}$ **+CaPu E2E**	**0.6780**
$\boldsymbol{Text_{VLSP\text{-}UnCaPu}}$ **+CaPu E2E**	**0.8178**

The importance of capitalization and punctuation was also observed in the NER model run test. The results show that the effectiveness of the CaPu

model to improve the accuracy of the NER model on the output text of the ASR, the F1 score of the NER model increased by approximately 0.05 from 0.6319 to 0.6780. The model also significantly improves 0.1398 points F1 when applied to standard text without punctuation and capitalization compared to the output text of ASR.

Table 4.4, E2E model suggests better but insignificant results than Pipeline (0.0067 for output text of ASR and 0.0037 for standard text without punctuation and capitalization).

Table 4.4: Comparison of E2E model with Pipeline model

NER system	**F1**
$Text_{VLSP\text{-}Audio\text{-}ASR}$ +CaPu Pipeline	0.6713
$\mathbf{Text_{VLSP\text{-}Audio\text{-}ASR}}$ **+CaPu E2E**	**0.6780**
$Text_{VLSP\text{-}UnCaPu}$ +CaPu Pipeline	0.8141
$\mathbf{Text_{VLSP\text{-}UnCaPu}}$ **+CaPu E2E**	**0.8178**

Although the results have not significantly improved affect, the Pipeline model facilitates the process of training to separate components, requires separate training algorithms and separate loss functions for each component and quantity. Large hyperparameters lead to high complexity in design and training. The errors generated in each component are not calculated when combined with other components, so the cumulative error is large. In contrast, with the E2E model, all the parameters of the model are trained simultaneously with only one loss function. The entire computational flow graph is optimized simultaneously by the back-propagation algorithm. The errors generated between the components are all calculated thereby minimizing the overall error. The inference process is also simpler and faster as there are no intermediate transitions between the component models. Therefore, the E2E model still has certain advantages and the continuous improvement of the E2E model for the Vietnamese speech NER problem is necessary to achieve higher performance and take advantage of the superiority in training. model training and in practical application deployment.

4.5. Conclusion of Chapter 4

Chapter 4 proposed the NER model for the Vietnamese ASR system using both Pipeline and E2E approaches. The experiments demonstrated the effectiveness of incorporating the CaPu model, which improved the NER model's performance. The E2E model performed slightly improve performance. Combining the multitask learning model with the punctuation and capitalization restoration model increased the F1 score by approximately 0.05 and significantly improved the NER model's F1 score by 0.14 when applied to standard texts without capitalization and punctuation.

CONCLUSION

The main results of the thesis

(1) Building initial data sets for experimenting with standardized models and identifying entity identifiers for the output text of the Vietnamese ASR system. The datasets described in the work [CT1, CT2, CT4, CT6].

(2) Propose a method to restore capitalization and punctuation for the output text of Vietnamese ASR using Transformer Encoder - CRF model combined with segmentation, merging overlapping paragraphs. Research results have been published in the works [CT2, CT3, CT5].

(3) Propose a pre-training language representation model for Vietnamese text with the name ViBERT based on RoBERTa architecture.

(4) Proposing a Pipeline model for the Vietnamese speech NER problem. Shows the importance of incorporating the CaPu model into normalizing the input text for the NER model. The results are listed in the work [CT4].

(5) Propose the first E2E model to solve the NER problem for Vietnamese voices along with new proposals such as parameter sharing techniques, multi-tasking training techniques. Research results have been published in the work [CT6].

sResearch development direction

(1) Optimize the solutions to minimize the impact of data errors in the output text of ASR to improve the quality of Vietnamese NER model.

(2) Develope the experimental NER for upper case recovery, making E2E ASR system more improved.

(3) Conduct the experiment using proposed models for other published data sets to compare the effectiveness of the model.

(4) Apply the proposed model to identify identity entities for documents belonging to specialized domains to clarify the feasibility of the model.

(5) Continue to improve the E2E model and the corresponding training algorithms to achieve better performance for the Vietnamese speech NER problem.

LIST OF AUTHOR'S DISCLOSURES

[CT1]. Nguyen Thi Minh Huyen, Ngo The Quyen, Vu Xuan Luong, Tran Mai Vu, **Nguyen Thi Thu Hien** (2018), *"VLSP shared task: Named Entity Recognition"*, Journal of Computer Science and Cybernetics, V.34, N.4, p.283–294.

[CT2]. **Thu Hien Nguyen**, Binh Nguyen Thai, Hung Nguyen Vu Bao, Truong Do Quoc, Mai Luong Chi, Huyen Nguyen Thi Minh (2019), *"Recovering Capitalization for Automatic Speech Recognition of Vietnamese using Transformer and Chunk Merging"*, Proceedings of 2019 the 11th International conference on knowledge and systems engineering, Vietnnam, p.430-434.

[CT3]. Binh Nguyen, Vu Bao Hung Nguyen, **Thu Hien Nguyen**, Pham Ngoc Phuong, The Loc Nguyen, Quoc Truong Do, Luong Chi Mai (2019), *"Fast and Accurate Capitalization and Punctuation for Automatic Speech Recognition Using Transformer and Chunk Merging"*, Proceedings of COCOSDA, Cebu, Philippines, p. 1-5, doi: 10.1109/O-COCOSDA46868.2019.9041202.

[CT4]. Nguyen, T. Binh, Nguyen, Q. Minh, Nguyen, **Nguyen, T.Hien**, Do, Q. Truong, & Luong, C. Mai (2020) *"Improving Vietnamese Named Entity Recognition from Speech Using Word Capitalization and Punctuation Recovery Models"*, Proceedings of Interspeech 2020, p.4263-4267, Shanghai, China.

[CT5]. **Thu Hien Nguyen,** Thai Binh Nguyen, Ngoc Phuong Pham, Quoc Truong Do, Tu Luc Le, Chi Mai Luong (2021), *"Toward Human-Friendly ASR Systems: Recovering Capitalization and Punctuation for Vietnamese Text"*, Journal of IEICE TRANSACTIONS on Information and Systems, Vol.E104-D, No.8, p.1195-1203 (Scopus).

[CT6]. **Thu Hien Nguyen**, Thai Binh Nguyen, Quoc Truong Do, Tuan Linh Nguyen (2022), *End-to-end named entity recognition for vietnamese speech*, Proceedings of O-COCOSDA 2022 (The 25th conference of the Oriental COCOSDA), p.193-197, 979-8-3503-9855-7 ©2022 IEEE