IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 2, April 2025, pp. 1605∼1615
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i2.pp1605-1615

Multi-task deep learning for Vietnamese capitalization and punctuation recognition

Phuong-Nhung Nguyen[1], Thu-Hien Nguyen[2], Nguyen Truong Thang[1], Nguyen Thi Thu Nga[1], Nguyen Thi Anh Phuong[1], Tuan-Linh Nguyen[3]

[1]Institute of Information Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam
[2]Faculty of Mathematics, Thai Nguyen University of Education, Thai Nguyen, Vietnam
[3]Faculty of Electronics Engineering, Thai Nguyen University of Technology, Thai Nguyen, Vietnam

Article Info

Article history:

Received Feb 29, 2024
Revised Oct 21, 2024
Accepted Nov 14, 2024

Keywords:

Capitalization
Deep learning
Multitask learning
Named entity recognition
Punctuation

ABSTRACT

Speech recognition is the process of converting the speech signal of a particular language into a sequence of corresponding content words in text format. The output text of automatic speech recognition (ASR) systems often lacks structure, such as punctuation, capitalization of the first letter of a sentence, proper nouns, and names of locations. This absence of structure complicates comprehension and restricts the utility of ASR-generated text in various applications, such as creating movie subtitles, generating transcripts for online meetings, and extracting customer information. Therefore, developing standardization solutions for the output text from ASR is necessary to improve the overall quality of ASR systems. In this article, we use the idea of multitask deep learning for the task of capitalization and punctuation recognition (CPR) for the output text of Vietnamese ASR, with the aim of the named entity recognition (NER) task as a supplement to help the CPR model perform better, and proposed to use text-to-speech (TTS) to create a dataset for CPR-NER multitask model training. The experiment results show that the multi-task deep learning model has improved CPR results by 6.2% of F1 score with ASR output and 7.1% on raw text.

Corresponding Author:

Tuan-Linh Nguyen
Faculty of Electronics Engineering, Thai Nguyen University of Technology
Thai Nguyen, Vietnam
Email: ntlinh@tnut.edu.vn

1. INTRODUCTION

Automatic speech recognition (ASR) refers to the various processes, technologies, and methodologies that facilitate improved interaction between humans and computers by converting spoken language into written text [1], [2]. The output text of the ASR system, besides containing some insertion, deletion, and word substitution errors, often has no punctuation and no capitalization. The issue of standardizing text by restoring punctuation and capitalization is necessary. It will make the text more structured and understandable. Furthermore, it will facilitate NLP tasks including named entity recognition (NER), syntactic parsing, part-of-speech tagging, and discourse segmentation [3]. Additionally, it benefits practical applications such as generating movie subtitles, broadcasting news transcripts, creating documents for online meetings, and extracting customer information more effectively [4].

Research in this field has traditionally focused on individual tasks, such as restoring punctuation [5]–[7] or capitalization [8]–[10]. However, these separate processing techniques are challenging to improve the overall effectiveness of the ASR systems. Therefore, recently, authors have proposed models to simultaneously handle two tasks [4], [11]–[13].

Initially, research in this field utilized methods such as maximum entropy [14], hidden markov model (HMM) [15], long short-term memory (LSTM) [16], and conditional random field (CRF) [17], [18]. Currently, the most advanced models for both punctuation restoration and capitalization recognition use neural network models. Studies have explored the character-level recurrent neural network (RNN) model [19], bidirectional RNN [20], [21]. Particularly, with the emergence of the transformer model and its variations like BERT, RoBERTa, ALBERT, DistilBERT, mBERT, and XLM-RoBERTa, studies on restoring punctuation and capitalization on the transformer model [12], [22], BERT [10], [23], RoBERTa [24], have shown positive results. Moreover, research on the Vietnamese language has also followed this innovative and promising trend, using CRFs combined with CNNs and LSTM [25], transformer [26]–[28]. Notably, the ViBERT model, based on RoBERTa for Vietnamese, is the latest model that achieves better performance in both tasks [29].

There are also have some limitations and challenges that need to be solved for the problem of restoring punctuation and capitalization in general and Vietnamese in particular. Regarding data-related issues, to deal with low-resource language data, some studies have experimented with using standardized text without punctuation and capitalization [3], [6], [26]. Because commas and periods appear more frequently than other marks, most research focuses only on these marks [12], [30], [31]. ASR output text does not have punctuation, so it is often long and endless. To process such text in models, input sequences are usually randomly cut into ranges of 20-30 words [32] or 20-50 words [24], maximum length 100 words [33] and 150 words [25]. Determining the appropriate length for cutting is a matter that needs to be considered. The proposal to use overlap-chunking in cutting and concatenating sequences has helped improve the model [9]. The issue of mismatched sequence lengths due to errors like insertions, deletions, and substitutions in the ASR output text is one of the major challenges that need to be addressed. Determining the order of punctuation restoration and capitalization is still an important issue as it can significantly impact the final results [29].

Multi-task deep learning (MTDL) has gained significant attention in the field of machine learning [34] and natural language processing (NLP) [35]. It has been used to address challenges such as overfitting and data scarcity in NLP tasks. Different architectures, including parallel, hierarchical, modular, and generative adversarial architectures, have been employed for multi-task learning (MTL) in NLP. Optimization techniques such as loss construction, data sampling, and task scheduling have been utilized to train multi-task models effectively. MTL has shown promising results in various NLP tasks, including machine translation, dialogue-based systems, sentiment analysis, and machine reading comprehension.

In this study, our goal is to propose a MTDL for simultaneous capitalization and punctuation recognition (CPR). We combine MTL with the hypothesis that integrating a NER model will improve the performance of punctuation restoration and capitalization. Additionally, we also employ the overlap-chunking technique and the ViBERT language model, which have shown effectiveness in NER [13]. Particularly, to cope with low-resource language data, we consider using text-to-speech (TTS) to augment the training data.

This paper is structured as follows: the initial session addresses the problem and reviews related research. The second session discusses our proposed approach and the architecture of the MTDL CPR system. In the third session, the focus is on the dataset's implementation and preparation for both training and testing. Session four details the configuration steps for the testing process and presents the evaluation outcomes. Lastly, the fifth session offers a conclusion and recommendations for future research avenues.

2. METHOD

2.1. Multi-task deep learning for capitalization and punctuation recognition

Using the idea from MTL for the task of punctuation and capitalization restoration, a MTDL approach is proposed with the aim of improving the performance of the CPR task. Figure 1 illustrates the proposed MTDL model, which consists of the main CPR task combined with an auxiliary NER stream that provides additional information about the named entity (NE) for the punctuation and capitalization restoration task. The input data to the model is the output text of Vietnamese ASR without punctuation and capitalization, with a length of N. During recognition, some sentences may contain errors such as substitutions, insertions, and deletions, making the punctuation and capitalization restoration task more challenging. The sentence from input is transferred through the ViBERT, a Vietnamese language representation model. In this study, a transfer learning approach is applied with ViBERT, a pre-trained model that is kept fixed in the proposed MTDL model.

The output of ViBERT is a matrix of size $(Nx768)$. This representation matrix is simultaneously fed into three blocks: (i) the NER information extraction block, (ii) the main CPR restoration block, and (iii) the NER auxiliary learning block using a MTL mechanism.

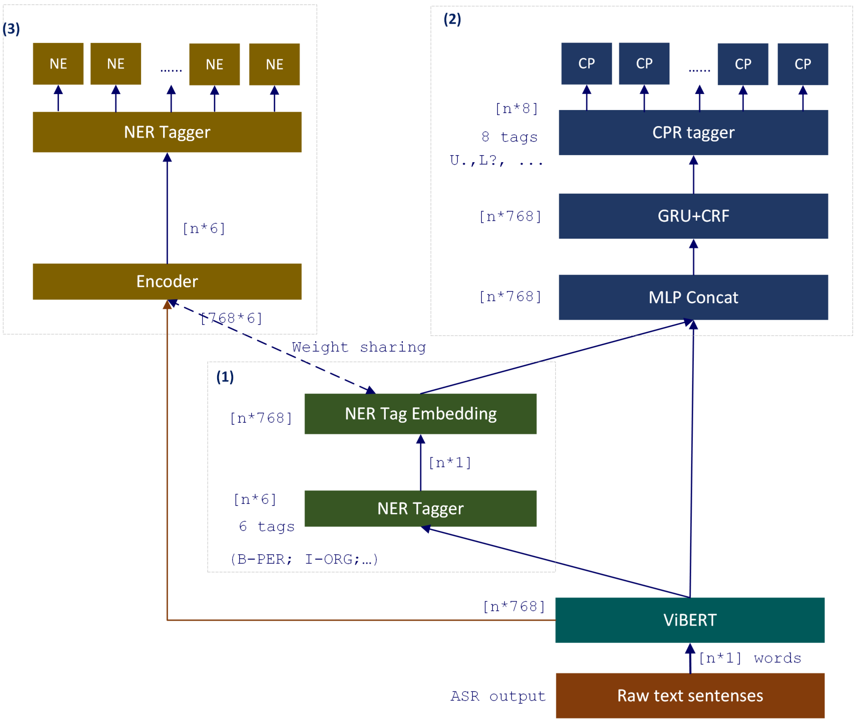


Figure 1. Proposed MTDL CPR model

Figure 1 shows the proposed multitask deep learning model, which includes the main CPR stream and NER stream to support the NE information for CPR. Both CPR and NER tagger are based on the transformer decoder structure in [29]. The input data consisted of unpunctuated and uncapitalized Vietnamese sentences, which were extracted from the output of an ASR system, with a set size of N.

The input sentence underwent processing using the representation of Vietnamese language model known as ViBERT. This model produced an output in the form of a matrix, characterized by dimensions of (Nx768), which encode the representation of the input sentence. This matrix representation was simultaneously directed toward three distinct components: (i) a block that extracts NE information, referred to as the NE encoder; (ii) the main CPR block; and (iii) an auxiliary block for NER.

The supplemental block for extracting NER information incorporated a NER tagging block along with an NER embedding block, which functioned to encode recognizable NER tags. These encoded tags were meant to supply pertinent information to the CPR block. The CPR block represented the core task of the MTDL model and utilized output from the sentence's representation matrix produced by ViBERT. This input was merged (concatenated) with the NER tag encoding matrix, emanating as the output from block (ii). This merger with the output from block (ii) enriched the NER information, thereby enhancing the precision of the CPR labeling process.

The output from the block was a collection of probabilities corresponding to NER tags for the specific input sentence, and the loss function for this process was calculated based on these NER labels. The encoder block was linked to the NER embedding block through a mechanism that shared parameters, as illustrated in (1). Within this configuration, the weight matrix of the NER embedding was replicated from the transposed matrix of the encoder in block (iii).

$$W_{\text{emb}} = W_{\text{enc}}^{T} \tag{1}$$

The network was trained using a MTL approach that involved two tasks: CPR, which served as the primary task, and NER, which functioned as a supportive, auxiliary task. The loss value for the MTDL model was determined by taking a weighted sum of the loss values from both the CPR and NER tasks. The MTDL model's loss value is computed as follows: the losses from the CPR and NER tasks are each assigned a weight, and then these weighted losses are summed together to yield the final loss value for the model, as shown in (2).

$$L_{\text{mtl}} = \alpha L_{\text{CPR}} + \beta L_{\text{NER}} \tag{2}$$

Where α is the weight for the loss value of the CPR task, and β is the weight for the loss value of the NER task. The choice of α and β depends on the importance of each task. In this study, the CPR task is considered the primary task and the NER task as the auxiliary task. Therefore, α and β are chosen to be 0.6 and 0.4, correspondently.

2.2. Input text overlap segmentation

The input for the CPR model receives output text from ASR. This text typically lacks punctuation, resulting in a long, indefinite sequence that poses considerable challenges for processing by models. Therefore, before being fed into the model, the input sequence is often segmented into fixed-length portions, which enhances the ability to process independently or in parallel segments. Related research particularly focuses on the segmentation of the input sentence sequence and often employs a strategy of random cutting within a range of 20-30 words [32] or 20-50 words [24]. However, this approach can lead to a lack of sufficient contextual information around the boundaries of the segments, resulting in predictions that are often inaccurate. To overcome this limitation, research has proposed a new technique for handling the cutting and concatenation of sequences by utilizing overlapping cuts. The main idea is to ensure that the resulting segments have enough contextual information for the words, enabling the CPR model to make the best predictions. After processing the overlapping segments, they are merged back into the output sequence of the original string.

Figure 2 details this architecture, which includes three components: the division of overlapping segments, the CPR model, and the merging of the overlapping segments. It can be seen that the input sentence is divided into three segments, which are stacked in an overlapping fashion. After being processed by the CPR model, the segments are recognized in the middle of the second segment having more surrounding context, thus being tagged more accurately than the words at the cutting boundary segments. Finally, these segments will be merged in an overlapping manner to form the restored sentence.

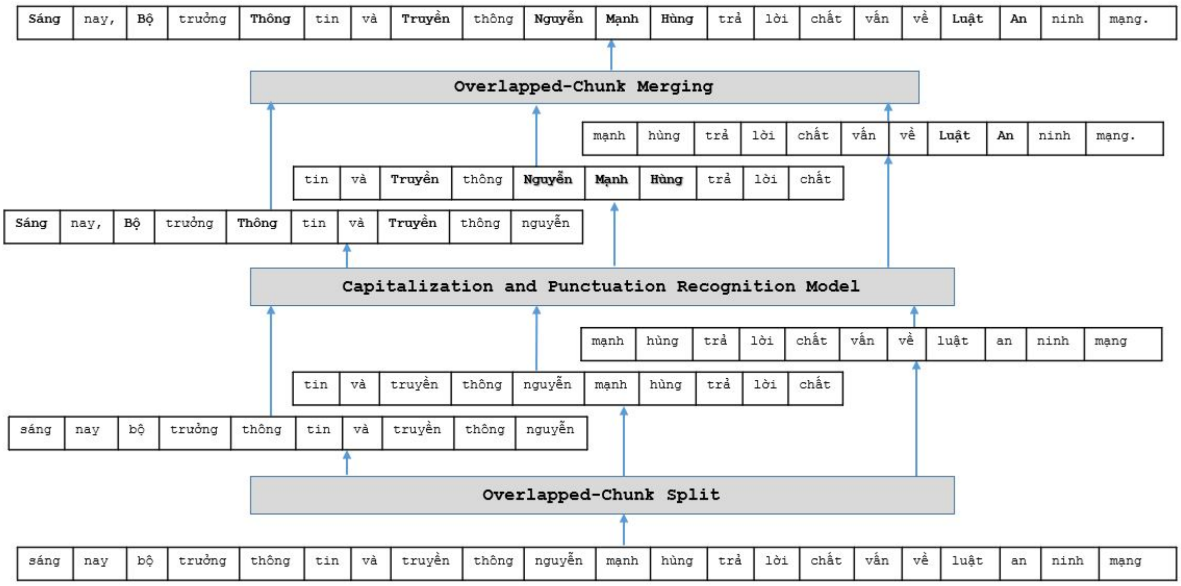

Sáng nay, Bộ trưởng Thông tin và Truyền thông Nguyễn Mạnh Hùng trả lời chất vấn về Luật An ninh mạng.
Overlapped-Chunk Merging
mạnh hùng trả lời chất vấn về Luật An ninh mạng.
tin và Truyền thông Nguyễn Mạnh Hùng trả lời chất
Sáng nay, Bộ trưởng Thông tin và Truyền thông nguyễn
Capitalization and Punctuation Recognition Model
mạnh hùng trả lời chất vấn về luật an ninh mạng
tin và truyền thông nguyễn mạnh hùng trả lời chất
sáng nay bộ trưởng thông tin và truyền thông nguyễn
Overlapped-Chunk Split
sáng nay bộ trưởng thông tin và truyền thông nguyễn mạnh hùng trả lời chất vấn về luật an ninh mạng

Figure 2. Overlap chunk-merging of input text

The chosen length for the segment cuts is an even number of words. Let l represent the length of the segment cut, and k is the length of the overlapping part, then we have $l = 2k$. Each input string S, containing

n words denoted as $w_1, w_2, ..., w_n$, will be divided into $n/l + (n-k)/l$ overlapping segments. Here, the i^{th} segment cut is a substring of words $w_{i-1}k+1, \ldots, w_{i+1}k$. The study has examined various values for l and k, and these values have been selected empirically to be appropriate. As the input sentence is divided into overlapping segments, the challenge in merging these segments is to determine which words should be discarded and which should be retained in the merged portion of the final sentence.

Let c be the length of the segment that will be either retained or discarded in the overlapping parts. For simplicity in calculation, we take $c = k/2$. Observations indicate that the ending words of the first overlapping segment and the initial words of the second overlapping segment (the words around the cutting point) lack substantial context. Therefore, the algorithm will discard the c segment at the end of the first overlapping part and retain the c segment in the second overlapping part. Consequently, the remaining words at the beginning of the first overlapping segment are kept, and the remaining words at the start of the second overlapping segment are discarded. This ensures that the words retained in the overlapping sections, always situated in the middle of segments, will have ample context, aiding in more accurate restoration. The process of discarding and retaining portions of the overlapping segments is repeated for subsequent overlapping segments. The post-catenation merging process is described as (3).

$$[[w_1, \dots w_{(2k-c)}] + \sum_{i=2}^{n-1} [w_{((i-1)k+c)}, \dots w_{(ik+c)}] + [w_{(n-2k+c)}, \dots w_n] \tag{3}$$

2.3. Dataset preparation

Numerous datasets for Vietnamese speech are now accessible to the public, examples of which include MICA VNSpeechCorpus [36], DEMEN567 [37], and VOV [38]. Despite these resources being reported, it should be noted that they may not encompass audio data that is both large-scale and of high quality. Besides, the existing speech datasets are not annotated with punctuation, capitalization, nor NE labels, Therefore, it is necessary to construct a sufficiently large dataset complete with labels for punctuation, capitalization, and NE in order to train the CPR model using a MTL approach. The dataset for NER tailored to the Vietnamese language was developed as part of the activities of the Vietnamese language and speech processing conference (VLSP). It comprises 32,000 instances in the training dataset with an additional 4,272 samples designated for testing. The dataset focuses on extracting specific entities which include individual names (PER), organizational names (ORG), and geographic locations (LOC). Detailed information about this dataset can be found in [39]. The initial dataset was in XML format, featuring entities that were hierarchically nested. The dataset was transformed into the CoNLL format for NER, which simplifies the process by only identifying the primary-level entities.

In the initial stages of data preparation, we transformed all text to lowercase and excised any special characters and punctuation, with the exception of commas, periods, and question marks. Subsequently, we structured the dataset with annotations as per the schema presented in Table 1, which is arranged into three columns. The first column contains the lowercase version of the input sentences. The second column holds the labels for CPR, with eight labels. The third column contains the NER labels, with six labels. Tables 2 and 3 show the statistics of our dataset.

Table 1. Format of dataset for MTDL training

Text	CPR label	NER label
minh	U$	B-PER
phúc	U,	I-PER
con	L$	O
trai	L$	O
ông	L$	O
hai	U$	B-PER
long	U$	I-PER
làm	L$	O
o	L$	O
hà	U$	B-LOC
giang	U.	I-LOC

Table 2. Dataset statistics of CPR tag

Tag	Train	Dev	Test
L$	760,182	38,885	189,058
L,	35,647	1,884	9,327
L.	29,917	1,531	6,596
L?	860	35	151
U$	127,973	6,630	30,313
U,	10,644	572	2,607
U.	8,097	400	1,843
U?	75	3	21

Table 3. Dataset statistics of NER tag

NER tag	Train	Dev	Test
B-LOC	15,871	841	2,002
B-ORG	8,627	383	2,152
B-PER	16,299	883	3,466
I-LOC	15,810	830	1,850
I-ORG	19,132	1,026	3,899
I-PER	11,621	644	3,077
O	886,035	45,333	223,470

An alternative method for data collection utilizing TTS systems has been outlined in Figure 3, which is suggested as a substitute for traditional recording techniques:

– Phase 1: Transform written text into audible speech with the use of a TTS tool

TTS technology has been under exploration in Vietnam for some time. Initially, VnSpeech was recognized as the pioneering Vietnamese TTS system. The VietSound system, which is a product of Ho Chi Minh City University of Technology, followed suit. Yet, these early systems had setbacks, particularly in producing speech that sounded disconnected and unnatural. In recent times, more robust and widely implemented TTS systems have emerged. Notably, FPT.AI offers a TTS service renowned for its effectiveness. Additionally, the Viettel's VTCC.AI system and Vbee's AI TTS system are among the powerful players in the field. Telecom companies like MobiPhone have also introduced their TTS solutions. Google has been providing a TTS tool that is celebrated for its natural-sounding voice, positioning it as one of the top contenders in TTS software. Leveraging DeepMind's speech synthesis expertise, Google has crafted a voice quality that listeners may find challenging to differentiate from human speech. In our study, we utilized the dataset of VLSP 2018 along with Google's TTS system to convert text into speech for both training and validation purposes. Due to resource constraints, our focus was restricted to recording audio for the test set within the dataset, which comprises 4,272 samples totaling approximately 242,000 words. The speech dataset was generated by four individuals who read the text in various settings, resulting in over 26 hours of recorded audio.

– Phase 2: Utilizing the ASR system for speech recognition

Following the acquisition of the audio dataset from both TTS and recordings, the data underwent processing via an ASR system. For this phase, the Vietnam artificial intelligence solutions (VAIS) ASR system was selected due to its superior performance in recognizing Vietnamese speech when compared to other systems such as those by FPT, Viettel, Zalo, and Google, as noted in [40]. Additionally, the VAIS ASR system was previously utilized in research focused on identifying markers in Vietnamese speech processing using a Pipeline approach, as mentioned in [13]. To maintain consistency for result comparison, we opted to continue with the VAIS system for our experimental framework.

– Phase 3: Evaluation of ASR text from synthesized speech

Table 4 presents the error rates for the TTS-ASR process and the recorded text through ASR (REC-ASR) method, assessed across 241,899 words from the dataset. The findings showed only a minor discrepancy in ASR accuracy when comparing two data collection methods, which was deemed provisionally acceptable. In order to amass a substantial volume of training data, we sourced information from authoritative Vietnamese news websites, including vnexpress.net, dantri.com.vn, and vietnamnet.vn. This content was then processed using a speech synthesis system to generate the voice data. We employed the recorded speech from the VLSP 2018 dataset, featuring four distinct speakers, as our test set.

The dataset of VLSP2018 is segmented into training, development, and testing subsets. The training subset contains 30,280 Vietnamese sentences; the development subset includes 1,593 sentences; and the testing subset comprises 4,271 sentences, all formatted with uppercase letters and punctuation. The VLSP sentences were annotated for capitalization, punctuation, and NE recognition. For creating the test subset's audio, four individuals recorded the test text of the subset. The VAIS ASR system (https://vais.vn/en/speech-to-text-core/) was utilized to convert the spoken words into text data for evaluation purposes. For the generation of synthesized audio data from the textual VLSP data, Google's TTS system was utilized. Additionally, VAIS's ASR system was employed to transcribe the synthetic audio, producing a text dataset for training the MTDL model.

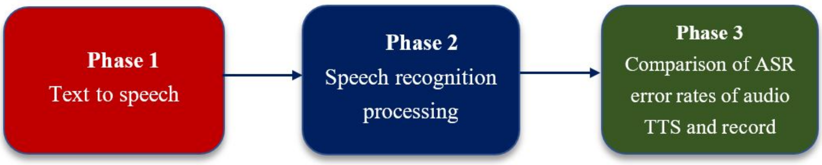

Phase 1
Text to speech
Phase 2
Speech recognition processing
Phase 3
Comparison of ASR error rates of audio TTS and record

Figure 3. Data preparation process

Table 4. Analysis of ASR error rates (%) of synthesized and recorded speech

	Number (%)	Outlier (%)	Other (%)	Total (%)
TTS-ASR	1.061	2.418	1.582	5.068
REC-ASR	1.427	2.371	2.228	6.032

2.4. Model setting

The ViBERT model is a scaled-down version of the $RoBERTa_{base}$, which is enforced using the fairseq framework. This streamlined model retains 4 encoder layers as its $RoBERTa_{base}$ counterpart. The number of attention heads has been reduced from 12 in the original to 4 in ViBERT. Each training instance can accommodate up to 512 tokens. The ViBERT model is trained using the Adam optimization algorithm with a batch size of 512. It has a peak learning rate set at 0.0003 and a warm-up phase consisting of 3,000 updates. The overall number of updates during training reaches 800,000. This training process is performed on two Nvidia 2080Ti GPUs, each with 12 GB of memory, and spans a duration of 5 weeks.

The suggested MTL model features both a CPR tagger and an NE tagger, both configured identically. Each tagger utilizes a transformer decoder to extract text features. At the output, a CRF model is employed to produce seven distinct labels. These labels include the beginning (B-X) and inside (I-X) tags for entities in the categories of organizations (ORG), persons (PER), and locations (LOC), as well as the O tag, which denotes tokens that do not correspond to any NE. For the output of CPR, the model will assign labels that include 'U' to denote uppercase, 'L' for lowercase, combine with labels '$' (indicating the absence of punctuation); '.' (period); ',' (comma); '?' (question mark) to add punctuation to the input word. Consequently, each word will be associated with one of the following eight labels: 'U$'; 'L$'; 'U.'; 'L.'; 'U,'; 'L,'; 'U?'; 'L?'.

3. RESULT AND DISCUSSION

3.1. Evaluation metrics

Our model's performance is assessed through precision, recall, and the F1-score metrics for both tasks as shown in (4) to (5).

$$precision = \frac{TP}{TP + FP} \tag{4}$$

$$recall = \frac{TP}{TP + FN} \tag{5}$$

$$recall = \frac{2 * precision * recall}{precision + recall} \tag{6}$$

where true positive (TP) denotes the number of punctuation and capitalization tags that were correctly identified by the model. False positive (FP) signifies the number of punctuation and capitalization tags that were mistakenly identified. On the other hand, false negative (FN) refers to the number of punctuation and capitalization tags that the model fail to recognize.

3.2. Experiment result

Table 5 presents the performance of our MTDL model in comparison to the baseline models (without NER auxilary task) transformer decoder CRF. For the single-task training, the state-of-the-art method involves refining ViBERT pre-trained language model for token classification tasks by adding a CRF layer on top [29]. The result presents the F1 scores associated with various types of inputs and highlights the shortcomings of the ASR-generated text when contrasted with text that is not case-sensitive (the reference text without capitalization and punctuation information) while the CPR results decreased from 0.7818 to 0.6319.

Table 5. Capitalization and punctuation recognition results

Input type	Single task	Multi-task
Uncased reference text	0.7818	0.8375
ASR output	0.7319	0.7780

The MTL approach outperforms the single-task learning by a significant margin, as evidenced by the scores of 0.8375 and 0.7818, respectively. This suggests that MTL is more effective at generalizing from uncased reference text, potentially due to its ability to capture and utilize shared representations from NER that are relevant to the input text. The high performance of 0.8375 for MTL in processing uncased reference text indicates robust feature extraction and learning, which can handle the absence of case information effectively. This could be attributed to the multi-task framework's potential to leverage auxiliary information from NER task, improving the model's understanding of context.

The findings further confirmed that integrating the NER component enhances the performance of the CPR model processing ASR-generated text. The F1 score for the CPR model saw a notable increase of approximately 6.2%, rising from 0.7319 to 0.778, after adding the NE recognition model to the text of ASR output. Additionally, the NER model proved to be highly valuable, as it substantially raised the F1 score of the CPR model by 7.1% with unprocessed text, which in this case, lacked capitalization and punctuation in the reference text.

3.3. Ablation study

Our model's performance assessment reveals the significance of its individual components in comparison to our optimal model, which incorporates ViBERT as the encoding layer alongside all components. Table 6 illustrates the essential role each component plays in our multitask learning model's effectiveness. System performance deteriorates upon alteration or omission of components. The complete model achieved an F1 score of 0.8375, but this score dropped to 0.8034 when we excluded the NER feature and relied solely on contextual embedding for the capitalization and punctuation prediction tasks. When we substituted the overlap-chunk-merging the input text with the non-overlap-chunk-merging, the F1 scores for the capitalization prediction task and punctuation prediction task fell to 0.8182. The findings indicate that the presence of overlapping words in the model supplies additional information for prediction, and our method of merging chunks is capable of appropriately selecting segments from the areas of overlap. The most substantial decline occurred when we replaced contextual embeddings with word embeddings (GloVe), resulting in a 0.16 decrease in F1 score.

Table 6. Performance metrics with component modifications

Component	Modification	Precision	Recall	F1
Full	None	0.8398	0.8352	0.8375
NER	Removal	0.8114	0.7953	0.8034 (4.08% ↓)
Chunking	Non overlap	0.8215	0.8149	0.8182 (2.29% ↓)
Embedding	GloVe	0.6826	0.6720	0.6773 (19.12% ↓)

4. CONCLUSIONS

In our research, we introduced the application of a NER model to enhance capitalization and punctuation recovery in Vietnamese speech processing with the MTL scheme. The experimental results provided evidence of the synergy between these methodologies. We also created and presented the inaugural speech dataset that establishes a foundation for future explorations into entity extraction from Vietnamese speech. Additionally, we highlighted the significant benefits of utilizing a pre-trained language model tailored for the Vietnamese language processing in CPR and NER multi-tasks, which set a new benchmark on the VLSP 2018 dataset. We have made this pre-trained model available to the research community to encourage further exploration. This contribution is particularly valuable for underrepresented languages such as Vietnamese.

ACKNOWLEDGEMENTS

This work was supported by the Institute of Information Technology, Vietnam Academy of Science and Technology under project number CS23.05.

REFERENCES

[1] D. Yu and L. Deng, *Automatic speech recognition: a deep learning approach*. in Signals and Communication Technology. London: Springer, 2015, doi: 10.1007/978-1-4471-5779-3.

[2] A. Jarin, A. Santosa, M. T. Uliniansyah, L. R. Aini, E. Nurfadhilah, and Gunarso, "Automatic speech recognition for indonesian medical dictation in cloud environment," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 2, pp. 1762–1772, 2024, doi: 10.11591/ijai.v13.i2.pp1762-1772.

[3] R. Pappagari, P. Zelasko, A. Mikolajczyk, P. Pezik, and N. Dehak, "Joint prediction of truecasing and punctuation for conversational speech in low-resource scenarios," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2021, pp. 1185–1191, doi: 10.1109/ASRU51503.2021.9687976.

[4] F. M. M. Batista and D. N. J. N. Mamede, "Recovering capitalization and punctuation marks on speech transcriptions," *Ph.D Thesis*, Fields of Science and Technology, Instituto Superior Técnico, Lisboa, Portugal, 2011.

[5] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text," in *Interspeech 2013*, ISCA, 2013, pp. 3097–3101, doi: 10.21437/Interspeech.2013-675.

[6] Q. H. Pham, B. T. Nguyen, and N. V. Cuong, "Punctuation prediction for vietnamese texts using condi tional random fields," in *Proceedings of the Tenth International Symposium on Information and Communication Technology - SoICT 2019*, New York, USA: ACM Press, 2019, pp. 322–327, doi: 10.1145/3368926.3369716.

[7] P. Żelasko, P. Szymański, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," in *arXiv-Computer Science*, 2018, pp. 1–5.

[8] R. H. Susanto, H. L. Chieu, and W. Lu, "Learning to capitalize with character-level recurrent neural networks: an empirical study," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Stroudsburg, USA: Association for Computational Linguistics, 2016, pp. 2090–2095, doi: 10.18653/v1/D16-1225.

[9] H. N. T. Thu, B. N. Thai, H. N. V. Bao, T. D. Quoc, M. L. Chi, and H. N. T. Minh, "Recovering capitalization for automatic speech recognition of vietnamese using transformer and chunk merging," in *2019 11th International Conference on Knowledge and Systems Engineering (KSE)*, IEEE, 2019, pp. 1–5, doi: 10.1109/KSE.2019.8919342.

[10] R. Rei, N. M. Guerreiro, and F. Batista, "Automatic truecasing of video subtitles using bert: a multilingual adaptable approach," in *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 2020, pp. 708–721, doi: 10.1007/978-3-030-50146-4_52.

[11] A. Gravano, M. Jansche, and M. Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2009, pp. 4741–4744, doi: 10.1109/ICASSP.2009.4960690.

[12] A. Vāravs and A. Salimbajevs, "Restoring punctuation and capitalization using transformer models," in *Statistical Language and Speech Processing*, Cham, Switzerland: Springer, 2018, pp. 91–102, doi: 10.1007/978-3-030-00810-9_9.

[13] T. B. Nguyen, Q. M. Nguyen, T. T. H. Nguyen, Q. T. Do, and C. M. Luong, "Improving vietnamese named entity recognition from speech using word capitalization and punctuation recovery models," in *Interspeech 2020*, 2020, pp. 4263–4267, doi: 10.21437/Interspeech.2020-1896.

[14] F. Batista, I. Trancoso, and N. Mamede, "Automatic recovery of punctuation marks and capitalization information for iberian languages," in *Proceedings of the I Iberian SLTech 2009*, 2009, pp. 99–102.

[15] J. P. Yamron, I. Carp, L. Gillick, S. Lowe, and P. V. Mulbregt, "A hidden markov model approach to text segmentation and event tracking," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 1998, pp. 333–336, doi: 10.1109/ICASSP.1998.674435.

[16] O. Tilk and T. Alumäe, "LSTM for punctuation restoration in speech transcripts," in *Interspeech 2015*, ISCA, 2015, pp. 683–687, doi: 10.21437/Interspeech.2015-240.

[17] X. Wang, H. T. Ng, and K. C. Sim, "Dynamic conditional random fields for joint sentence boundary and punctuation prediction," in *Interspeech 2012*, ISCA, 2012, pp. 1384–1387, doi: 10.21437/Interspeech.2012-398.

[18] M. Lui and L. Wang, "Recovering casing and punctuation using conditional random fields," in *Proceedings of Australasian Language Technology Association Workshop*, 2013, pp. 137–141.

[19] G. Ramena, D. Nagaraju, S. Moharana, D. P. Mohanty, and N. Purre, "An efficient architecture for predicting the case of characters using sequence models," in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, IEEE, 2020, pp. 174–177, doi: 10.1109/ICSC.2020.00035.

[20] O. Tilk and T. Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration," in *Interspeech 2016*, 2016, pp. 3047–3051, doi: 10.21437/Interspeech.2016-1517.

[21] A. Zahra, A. F. Hidayatullah, and S. Rani, "Bidirectional long-short term memory and conditional random field for tourism named entity recognition," *IAES International Journal of Artificial Intelligence*, vol. 11, no. 4, pp. 1270–1277, 2022, doi: 10.11591/ijai.v11.i4.pp1270-1277.

[22] R. Pan, J. A. García-Díaz, and R. Valencia-García, "Evaluation of transformer-based models for punctuation and capitalization restoration in spanish and portuguese," in *Natural Language Processing and Information Systems*, 2023, pp. 243–256, doi: 10.1007/978-3-031-35320-8_17.

[23] A. Nagy, B. Bial, and J. Ács, "Automatic punctuation restoration with bert models," *arXiv-Computer Science*, pp. 1–11, 2021.

[24] M. Courtland, A. Faulkner, and G. McElvain, "Efficient automatic punctuation restoration using bidirectional transformers with robust inference," in *Proceedings of the 17th International Conference on Spoken Language Translation*, 2020, pp. 272–279, doi: 10.18653/v1/2020.iwslt-1.33.

[25] H. T. T. Uyen, N. A. Tu, and T. D. Huy, "Vietnamese capitalization and punctuation recovery models," in *Interspeech 2022*, ISCA, 2022, pp. 3884–3888, doi: 10.21437/Interspeech.2022-931.

[26] B. Nguyen *et al.*, "Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging," in *2019 22nd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, IEEE, 2019, pp. 1–5, doi: 10.1109/O-COCOSDA46868.2019.9041202.

[27] V. T. Bui and O. T. Tran, "Punctuation prediction in vietnamese asrs using transformer-based models," in *PRICAI 2021: Trends in Artificial Intelligence*, 2021, pp. 191–204, doi: 10.1007/978-3-030-89363-7_15.

[28] H. Tran, C. V. Dinh, Q. Pham, and B. T. Nguyen, "An efficient transformer-based model for vietnamese punctuation prediction," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, 2021, pp. 47–58, doi: 10.1007/978-3-030-79463-7_5.

[29] T. T. H. Nguyen, T. B. Nguyen, N. P. Pham, Q. Truong, T. L. Le, and C. M. Luong, "Toward human-friendly ASR systems: recovering capitalization and punctuation for vietnamese text," *IEICE TRANSACTIONS on Information and Systems*, vol. 104, no. 8, pp. 1195–1203, 2021.

[30] F. Batista, D. Caseiro, N. Mamede, and I. Trancoso, "Recovering punctuation marks for automatic speech recognition," in *Interspeech 2007*, ISCA: ISCA, 2007, pp. 2153–2156, doi: 10.21437/Interspeech.2007-581.

[31] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *EMNLP 2010 - Conference on Empirical Methods in Natural Language Processing*, 2010, pp. 177–186.

[32] E. Cho, J. Niehues, and A. Waibel, "NMT-based segmentation and punctuation insertion for real-time spoken language translation," in *Interspeech 2017*, ISCA: ISCA, Aug. 2017, pp. 2645–2649, doi: 10.21437/Interspeech.2017-1320.

[33] T. Pham, N. Nguyen, Q. Pham, H. Cao, and B. Nguyen, "Vietnamese punctuation prediction using deep neural networks," in *SOFSEM 2020: Theory and Practice of Computer Science*, 2020, pp. 388–400, doi: 10.1007/978-3-030-38919-2_32.

[34] N. Nikentari and H. L. Wei, "Multi-task learning using non-linear autoregressive models and recurrent neural networks for tide level forecasting," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 1, pp. 960–970, 2024, doi: 10.11591/ijece.v14i1.pp960-970.

[35] R. M. Samant, M. R. Bachute, S. Gite, and K. Kotecha, "Framework for deep learning-based language models using multi-task learning in natural language understanding: a systematic literature review and future directions," *IEEE Access*, vol. 10, pp. 17078–17097, 2022, doi: 10.1109/ACCESS.2022.3149798.

[36] T.-N. Phung, M. C. Luong, and M. Akagi, "An investigation on perceptual line spectral frequency (PLP-LSF) target stability against the vowel neutralization phenomenon," in *2011 3rd International Conference on Signal Acquisition and Processing (ICSAP 2011)*, 2011, pp. 512–514.

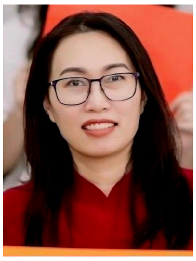
[37] P. T. Nghia, L. C. Mai, and M. Akagi, "Improving the naturalness of concatenative vietnamese speech synthesis under limited data conditions," *Journal of Computer Science and Cybernetics*, vol. 31, no. 1, 2015, doi: 10.15625/1813-9663/31/1/5064.

[38] V. L. Phung, H. K. Phan, A. T. Dinh, and Q. B. Nguyen, "Data processing for optimizing naturalness of vietnamese text-to-speech system," in *2020 23rd Conference of the Oriental COCOSDA International Committee for the Co-ordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA)*, IEEE, 2020, pp. 1–6, doi: 10.1109/O-COCOSDA50338.2020.9295025.

[39] H. T. M. Nguyen, Q. T. Ngo, L. X. Vu, V. M. Tran, and H. T. T. Nguyen, "VLSP shared task: named entity recognition," *Journal of Computer Science and Cybernetics*, vol. 34, no. 4, pp. 283–294, 2019, doi: 10.15625/1813-9663/34/4/13161.

[40] C. H. Nga, C.-T. Li, Y.-H. Li, and J.-C. Wang, "A survey of vietnamese automatic speech recognition," in *2021 9th International Conference on Orange Technology (ICOT)*, IEEE, 2021, pp. 1–4, doi: 10.1109/ICOT54518.2021.9680652.

BIOGRAPHIES OF AUTHORS



Phuong-Nhung Nguyen received the master degree from the University of Technology - Hanoi National University, majoring in information technology. Her research interests include data mining, machine learning, soft computing, and fuzzy computing. She is currently a researcher at the Institute of Information Technology, Vietnam Academy of Science and Technology. She can be contacted at email: ntpnhung@ioit.ac.vn.

Thu-Hien Nguyen has received a Ph.D. in information systems from the Institute of Information Technology, Vietnam Academy of Science and Technology. Her dissertation was "Research on text normalization methods and named entity recognition in Vietnamese speech recognition". She has been a lecturer at the Faculty of Mathematics, Thai Nguyen University of Education since 2005. Her research interests include natural language processing, artificial intelligence, databases, information system analysis and design, and digital transformation in education. She can be contacted at email: hienntt.math@tnue.edu.vn.

Nguyen Truong Thang received the Ph.D. degree from the Japan Advanced Institute of Science and Technology (JAIST), Japan, in 2005. He is currently with the Institute of Information Technology (IoIT), Vietnam Academy of Science and Technology. His major research interests include software quality assurance, software verification, program analysis, data mining, and machine learning. He can be contacted at email: ntthang@ioit.ac.vn.

Nguyen Thi Thu Nga received her Ph.D. in mathematics at the Department of Training, Military Academy of Science and Technology, Vietnam since 2023. In addition, she also holds the position of researcher in Computer Science and Computational Intelligence Research (2016-present). Her research field is technology and computer science. She currently working at the Institute of Information and Technology, Vietnam Academy of Science and Technology. She can be contacted at email: thungalnt@gmail.com.

Nguyen Thi Anh Phuong is currently a master at Hanoi University of Science – VietNam National University HaNoi, majoring applied mathematics computing. Her research interests include data mining, soft computing, and fuzzy computing. She is currently a researcher at the Institute of Information Technology, Vietnam Academy of Science and Technology. She can be contacted at email: ntaphuong@ioit.ac.vn.

Tuan-Linh Nguyen received the Ph.D. degree from the Kyungpook National University (KNU), Korea, in 2020. He is currently with the Thainguyen University of Technology (TNUT), Vietnam. His major research interests include deep fuzzy-neural network, deep learning, interpretable AI, and NLP. He can be contacted at email: ntlinh@tnut.edu.vn.