



Improving Vietnamese Named Entity Recognition from Speech Using Word Capitalization and Punctuation Recovery Models

Thai Binh Nguyen¹, Quang Minh Nguyen¹, Thi Thu Hien Nguyen^{1,2},
Quoc Truong Do¹, Chi Mai Luong^{1,3}

¹Vietnam Artificial Intelligence System, Vietnam

²Thai Nguyen University of Education, Vietnam

³University of Science and Technology of Hanoi, Vietnam

{binhnguyen|minhnguyen|truongdo}@vais.vn, hienntt.math@tnue.edu.vn, lcmmai@ioit.ac.vn

Abstract

Studies on the Named Entity Recognition (NER) task have shown outstanding results that reach human parity on input texts with correct text formatings, such as with proper punctuation and capitalization. However, such conditions are not available in applications where the input is speech, because the text is generated from a speech recognition system (ASR), and that the system does not consider the text formatting. In this paper, we (1) presented the first Vietnamese speech dataset for NER task, and (2) the first pre-trained public large-scale monolingual language model for Vietnamese that achieved the new state-of-the-art for the Vietnamese NER task by 1.3% absolute F1 score comparing to the latest study. And finally, (3) we proposed a new pipeline for NER task from speech that overcomes the text formatting problem by introducing a text capitalization and punctuation recovery model (CaPu) into the pipeline. The model takes input text from an ASR system and performs two tasks at the same time, producing proper text formatting that helps to improve NER performance. Experimental results indicated that the CaPu model helps to improve by nearly 4% of F1-score.

Index Terms: speech recognition, named entity recognition, capitalization and punctuation insertion

1. Introduction

Named Entity Recognition firstly has introduced in MUC-6 [1] in 1995, since then NER is a very active researched topic. Almost studies were done using written text, meanwhile, spoken text (that output from ASR) is much harder. One of the most challenges of doing NER from speech is dealing with the mismatch between ASR output and NER input. The mismatch is showing in many forms like do not have punctuation, not case sensitive. Sometimes, ASR error makes the meaning of the speech transcription output is changed that affecting the NER module harder processing accuracy. In [2, 3] evinced a significant improvement of NER result when the input is case-sensitive. Of course not any language need this kind of information like in [4] doing NER in Japanese speech data, [5] in Chinese language.

Many types of architectural models were study in this problem can be referred as hidden markov models was used in study [3, 6, 7], CRF in [8], SVM in [2, 4] and maximum entropy model in [5]. For now, deep learning model have proved as strong method for extracted entity, the study in [9] showing this overwhelming. To increase the accuracy of extracting entity from speech, some studies not only using text output from the ASR module but also consider the speech features. The author in [10] propose a 2-step approach for extracting named entity values. Thereby, first step the 1-best hypothesis produced by the ASR system will go through NER module to tag entity. Then

the second step will extract the entity values from the word lattice on the areas selected by the named entity module. Another method for this strategy is in [11] that adding loss of the NER model to loss of the ASR model so they can optimize the ASR model directly for the NER task. The research in this way needs to gather more data that have to include both text transcription and tagged entity in there. This makes collecting enough data to train models effectively more difficult, especially in rare languages.

The ASR output (hypothesis output) sometimes different from the reference text (due to ASR error) that makes hard to evaluate directly the accuracy of NER output. To fix the mismatch between these, many approaches was proposed. In [3], the authors make a module to align between hypothesis text and reference text before calculate the F1-score of NER result. The research in [12] define other structured NER dataset (including hierarchical and compositional) for effective benchmark. But [12] require re-label the NER dataset. In [13] study, they upgraded of MUC score with phonetic alignment and content comparison. This approach, semantically, make the score closer to the actual accuracy.

In summary, we gain state-of-the-art in the NER VLSP 2018 dataset by using our own language model trained using the large Vietnamese corpus dataset. We also proposed a pipeline for extracting entities in the speech dataset, building a new dataset to evaluate this proposed pipeline and proves the effectiveness of the proposition. We release our pretraining code and model implemented in PyTorch.

2. Related works

2.1. Automatic Speech Recognition

The goal of Automatic Speech Recognition (ASR) is to address the transcription correlates with audio signals. Normally, an ASR system contains two main components, which are acoustic model and language model.

In order to figure out the correlation between phonemes and signals, basically a Hidden Markov Model (HMM) model is used. Recently, TDNN-based acoustic model [14] has proved the better performance compare with conventional HMM-GMM (Gaussian mixture model) system. However, training TDNN still requires the alignment trained by GMM.

The main idea of language model (LM) comes from the need of predicting probability of next word. The n-gram model is frequently integrated into the ASR system to assign the conditional sequence of words. The LM in this paper is trained on large amount of text with various domain but mostly news. After training stage, the acoustic model finally composes with language model using a mapping dictionary of words and their pronunciation called lexicon.

2.2. Capitalization and Punctuation Recovery

Normally, the ASR output is lower case without any sentence mark, Capitalization and Punctuation Recovery (CaPu) is a task to regain that information. This is immensely important for other downstream tasks that use the output of ASR for example Machine Translation, NER. These tasks require correct text formatings, such as with proper punctuation and capitalization as one of the indispensable things.

Many studies on CaPu task for ASR processed in recently year. For example, [15] uses a phrase-based machine translation model to insert punctuation. The research in [16, 17] combine information from speech signal with text transcript to improve the performance. Conditional Random Field model was applied in [18, 19] to insert punctuation that considered as a part of speech tagging problem. Some of the newest works are [20, 21] that using the Transformer model applies with the chunk merging technique to do the CaPu task. This technique showing very effective because it processes with the output of ASR without boundary. In our work, we use this approach to handle CaPu task.

2.3. Named Entity Recognition

Aggarwal and Zhai state the definition of an entity’s identification problem [22]: the Name Entity Recognition function, also referred to as NER, is to recognise and classify named entities from free text into a collection of predefined categories, such as person, organisation, and location.

Highly effective entity recognition studies are also focused on statistics machine learning. One of the first machine learning methods applied to NER is the Hidden Markov model (HMM). Nymble is the first HMM-based NER system developed by Bikel et al [23]. Later built machine learning methods for NER have shifted from the HMM model to the discrete models to overcome the HMM disadvantages. A common discrete model used in entity recognition is the Maximum Entropy model (MaxEnt). In [24]. The results of the system were evaluated to be not inferior to those of other contemporaries but gained the advantage of model building costs. In [25], the authors proposed the maximum Markov Entropy model for the problem of entity identification. The typical studies for NER using this model [26, 27] have also achieved positive results. Conditional Random Fields (CRF) is another typical discrete model for entity identification problem [28]. In [29], the authors proposed a semi-Markov Conditional Random Fields model and applied to NER, the results were assessed to be better than conventional CRFs.

Recently, studies have focused on improving the efficiency of entity identification by adding information generated from large amounts of unlabeled data. This approach is the application of semi-supervised machine learning and deep learning machine learning techniques [30]. Since then, the deep learning method has been extensively studied for the NER problem. Many new models have been applied, improved, combined and achieved important results in many different languages [31, 32].

3. Proposed method

3.1. System overview

Figure 1 show an overview of the system extract NER in Speech. This system has 3 distinct submodules respectively are ASR, CaPu, and NER. ASR model will transcript speech signal to text, this text in raw format that doesn’t have punctuation and all lowercase. This text will be reformatted using the CaPu model and then information of entities is taken out from that text by using the NER model.

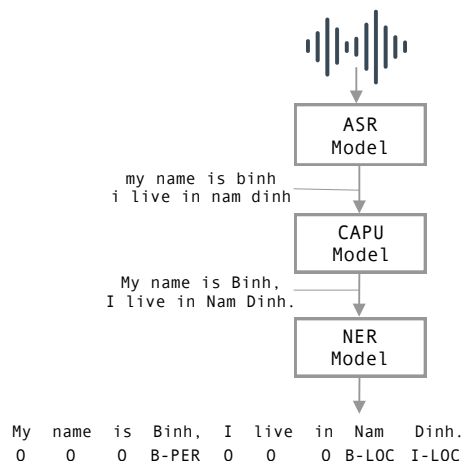


Figure 1: NER in speech system overview

3.2. Capitalization and Punctuation Model

Upper case and sentence mark take an important role that provides the meaning of a sentence, to be one of the indispensable information need to feed in a NER model. Unfortunately, this information was ignored in ASR model. Figure 1 reveals our proposal to attach a CaPu model to recovery this kind of information.

The study in [20] shows very suitably for our task because it can handle endless string that output from the ASR model and having a good performance. In [20], the authors proposed to handle the CaPu problem as a translation task that translate a lower case string without punctuation to a normal string with uppercase and sentence mark. The model was based using the Transformer sequence to sequence model and apply technique chunk merging. Example in figure 2 shows the idea how the CaPu model working with chunk merging technique. The input text is split into three overlap chunks then transformed in parallel using CaPu model. The idea of merging overlap chunks is to replace the boundary words by the same words but in the middle of other sentences. Example ‘tai chính’ words in the first sentence are replaced by ‘tai, Chính’ words in the second sentence to produce the final output.

3.3. Named Entity Recognition Model

The main difference in our NER module in this system is that instead of training from scratch or using other pre-trained word embeddings, we train our own word embedding. Pre-trained language models, especially the Bidirectional Encoder Representations (BERT) recently become really popular and helped to produce significant improvement gains for various NLP tasks.

In our work, we use RoBERTa architecture [33] (an improved recipe for training BERT models) and train it on Vietnamese corpus to make pre-trained language model. Because of the limitation of computing resources, we reduce the number of hidden layers and attention head and embedding dimension from the RoBERTa base model architecture (detail in section 5.3). The trained model we named it ViBERT. The pre-trained model we public in Github¹. Figure 3 depicts our NER model design. ViBERT was used to embed the input sentence. The bidirectional GRU models and CRF layer attached to the top of ViBERT to classify the entity-tag of each input word.

¹<https://github.com/nguyenvulebinh/vietnamese-roberta>

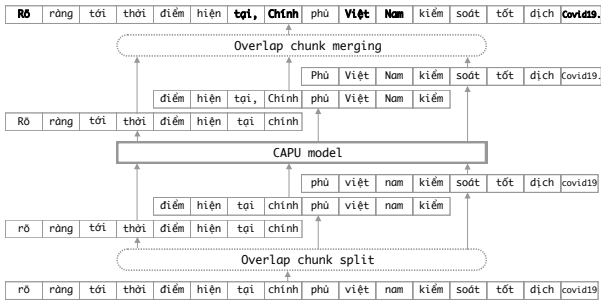


Figure 2: CAPU model apply chunk merging in Vietnamese [20]. Step 1: Split input sentence to overlap chunks. Step 2: Producing formatted text using CaPu model. Step 3: Concatenation the overlap chunks.

4. Speech NER dataset

NER on speech is one of the problems with rare official data even in the most popular language like English. In Vietnamese, this is the first attempt to build a dataset for this task. Two approaches that we consider are creating speech data from the NER dataset or vice versa. Recording audio from reading the NER text reference is much easier than tagging NER on the ASR reference transcript.

NER VLSP 2018 [34] is the best dataset, for now, to benchmark an NER system in the Vietnamese language. Because of the limit of resources, we only consider recording audio reading the test set in the dataset that includes 4272 samples with total 242k words. The speech data was created by 4 people reading in different environments that produce a total of more than 26 hours of audio.

5. Experiments

In this section, we evaluate the effectiveness of the the proposed NER model with RoBERTa language models, and how good the CaPu models can contribute to the NER accuracy on input speech data.

With regards to the ASR system, we utilized time-delayed neural network models (TDNN) trained with the Kaldi toolkit [35]. The model has 22 layers and is trained with 3000 hours of speech data.

5.1. Data preparation

NER dataset for the Vietnamese language was prepared in the Vietnamese Language and Speech Processing Conference (VLSP). It contains total 32.000 samples for the training set and 4272 samples for testing purpose. The entities need to be extracted are a person name (PER), name of an organizer (ORG) and location name (LOC). This dataset was detailed in [34]. The original data in XML format and contains entities at nested levels. To make it facilitate comparing with the public the result in [34], data was converted to CoNLL NER format and just detecting the entity at the first level.

CaPu and ViBERT need a large corpus to train. We use 50GB of text with approximate 7.7 billion words that crawl from many domains on the internet including news, law, entertainment, wikipedia and so on. Because of the variety of typing encode in Vietnamese language on the internet, we use Visen² library to unify encode method.

²<https://github.com/nguyenvulebinh/visen>

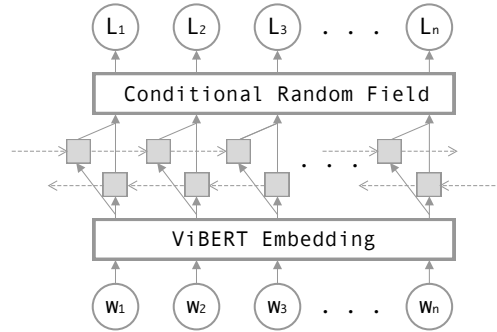


Figure 3: Named Entity Recognition Model

In the CaPu problem, punctuations were handle include ‘.’, ‘.’. We split the corpus into segments of random range from 4 to 60 words. The total data that use to train the CaPu model is more than 300 million samples. All data are prepared in the type of encoded as the description in [20]. To evaluate CaPu model, we use the reference text in VLSP 2018 test set.

ViBERT model was trained using corpus data processed by the byte-pair-encoding (BPE) algorithm. BPE was set up to output vocabulary size 50k. We don’t use Vietnamese word segmentation because it makes the size of vocabulary bigger and it doesn’t make better representation due to the transformer architecture already make good use context of the sentence with multi heads attention.

To training the ASR model, we prepare Vietnamese speech corpus consists of approximately 3000 hours of speech data including various domains and speaking styles. The data is augmented with noise and room impulse respond to increase the quantity and prevent over-fitting.

The dataset prepared in section 4 was used to evaluate the performance of the whole system. This dataset includes speech data to evaluate the ASR system and entity-tag to evaluate the accuracy of extracted entity from the ASR output by using the NER model.

5.2. Evaluation metric

In the normal NER system, the input is text and the output is a label for each word in that text. However, in our system (extracting NER in Speech), the input is speech and we need extract every entities in that audio. As system description in section 3.1, NER module will extract entities from the output of the ASR module. The problem here is the ASR output can have some types of errors like insertions, deletions, substitutions that make the length of hypothesis output labels may be different from the ground truth labels, that make it impossible to calculate the *F1* score like in the normal NER system.

To bypass this mismatch, the ASR output will be compared with the reference text in the NER dataset. If ASR output is right (**T**), the hypothesis entity tag was remained. If the error type is deletions (**D**) or substitutions (**S**), the hypothesis output of this word will become tag **O**. Else if error type is insertions (**I**), the label will be removed. That way will make the size of reference labels equal with the size of hypothesis labels. For example in figure 4 ASR output have some error. After alignment, the precision (*P*) is 100% and recall (*R*) is 33.33% because only 1 of 3 tags is correct (Việt Nam). $F1 = 2 * (P * R) / (P + R) = 50\%$

5.3. Model setups

We reduce the size of the *RoBERTa_{base}* model that implementation in fairseq [36] to produce ViBERT. This

Reference text	Ông	Tedros	thuộc	WHO	đánh	giá	cao	Việt	Nam	
Reference tag	O	B-PER	O	B-ORG	O	O	O	B-LOC	I-LOC	
ASR + CAPU output	Ông	Dros	thuộc	_	đánh	giá	cao	tại	Việt	Nam
Hypothesis tag	O	B-PER	O	O	O	O	O	O	B-LOC	I-LOC
ASR error	T	S	T	D	T	T	T	I	T	T
Alignment text	Ông	_	thuộc	_	đánh	giá	cao	Việt	Nam	
Alignment tag	O	O	O	O	O	O	O	B-LOC	I-LOC	

Figure 4: Evaluation example

model contains 4 encoder layers compare with 4 layers in *RoBERTa_{base}*. The number of heads also reduce from 12 to 4 and the hidden dimension size is down from 768 to 512. Each training sample contains at most 512 subwords tokens. Like [33], we optimize the models using Adam. ViBERT was trained using a batch size of 512 and a peak learning rate of 0.0003 with 3000 warming update steps. The total updating steps is 800k. We use 2 Nvidia 2080Ti GPUs (12GB each) for 5 weeks.

NER model setting use ViBERT for word representation and has 4 bidirectional GRU layers. The hidden size of GRU cell is 512. CRF was used in the output layer to produce 9 labels (**B-X**, **I-X**, **O** where **X** in set {**ORG**, **PER**, **LOC**}). This model also was optimized by using adam, the batch size is 64 and the training process converges after 30 epochs.

5.4. Benchmark result

Our NER model is the pre-trained word embedding (ViBERT) combine with GRU and CRF layer shows its effecton making the SOTA result (90.18%) when compared with the previous public result (table 1). This is the result evaluated directly using the reference text that is case sensitive and having full punctuation.

Table 1: Evaluation NER models on the NER VLSP 2018 dataset

Model type	F1 score
Vi Tokenizer + Bidirectional Inference [34]	88.78%
VNER [37]	77.52%
Multi layers LSTM [34]	83.8%
CRF/MEM+BS [34]	84.08%
ViBERT + GRU + CRF (our result)	90.18%

The word error rate of our ASR system is 6.57%. Table 2 shows that if the ASR output put directly to the NER model, the entity identity results are critically reduced from 90.18% to 63.89%. The importance of the uppercase letter and punctuation also be observed in the experiment of running NER model on uncased reference text (without case sensitive and punctuation). In this case, F1 score drop vastly from 90.18% to 75.35%.

Table 2: Evaluation NER models on the different text input type

Input type	F1 score
Reference text	90.18%
ASR output	63.19%
ASR output + CAPU	67.13%
Uncased reference text	75.35%
Uncased reference text + CAPU	81.41%

The main different between the ASR output and the reference text is the case sensitive and punctuation (ignore the error of the ASR model). In the figure 5 demonstrate the result of the

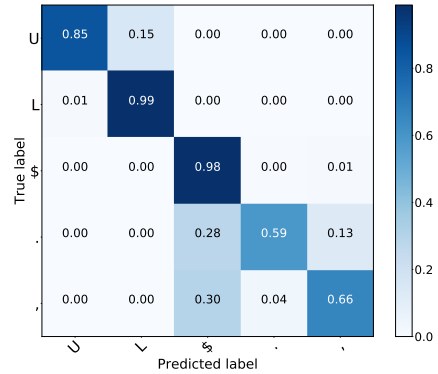


Figure 5: Evaluating the CAPU model on the uncased reference text (without case sensitive and punctuation)

CaPu model. The accurate of recovery upper case letter is 85%. Recuperation of punctuation is harder when the accurate is remain in nearly 60% for the dot mark (‘.’) and 66% for comma mark ‘,’. The error of recovering punctuation happened when the CaPu model does not understand the meaning of the input sentence and put a blank mark (\$) after these words.

Table 2 elucidate the effectiveness of the CaPu model in improving the accuracy of the NER model working on ASR output. F1 score of the NER model raises nearly 4% from 63.19% to 67.13% when applying the CAPU model on the ASR output. The CaPu model also shows its value, improving more than 6% F1 score of the NER model when apply to the uncased reference text.

6. Conclusions

In this study, we proposed to apply the capitalization and punctuation recovery model to improve named entity recognition from Vietnamese speech. The experimentation demonstrated the effectiveness of this combination. Through the research, we presented the first speech dataset that lay the groundwork for doing study extracted entity in speech for Vietnamese language. Besides that, we presented the effective impact of pre-trained language model for Vietnamese language apply for NER task that achieved new state-of-the-art on VLSP 2018 dataset. We public this pre-trained model to other researchers can further experiment. This is very useful for minority languages like Vietnamese.

7. References

- [1] R. Grishman and B. M. Sundheim, “Message understanding conference-6: A brief history,” in *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.
- [2] M. Surdeanu, J. Turmo, and E. Comelles, “Named entity recognition from spontaneous open-domain speech,” in *Ninth European Conference on Speech Communication and Technology*, 2005.
- [3] F. Kubala, R. Schwartz, R. Stone, and R. Weischedel, “Named entity extraction from speech,” in *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*. Citeseer, 1998, pp. 287–292.
- [4] K. Sudoh, H. Tsukada, and H. Isozaki, “Incorporating speech recognition confidence into discriminative named entity recognition of speech data,” in *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2006, pp. 617–624.

- [5] L. Zhai, P. Fung, R. Schwartz, M. Carpuat, and D. Wu, "Using n-best lists for named entity recognition from Chinese speech," in *Proceedings of HLT-NAACL 2004: Short Papers*. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 37–40. [Online]. Available: <https://www.aclweb.org/anthology/N04-4010>
- [6] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel, "Named entity extraction from noisy input: Speech and OCR," in *Sixth Applied Natural Language Processing Conference*. Seattle, Washington, USA: Association for Computational Linguistics, Apr. 2000, pp. 316–324. [Online]. Available: <https://www.aclweb.org/anthology/A00-1044>
- [7] J.-H. Kim and P. C. Woodland, "A rule-based named entity recognition system for speech input," in *Sixth International Conference on Spoken Language Processing*, 2000.
- [8] C. Parada, M. Dredze, and F. Jelinek, "Oov sensitive named-entity recognition in speech," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [9] V. Yadav and S. Bethard, "A survey on recent advances in named entity recognition from deep learning models," in *COLING*, 2018.
- [10] F. Béchet, A. Gorin, J. Wright, and D. H. Tur, "Named entity extraction from spontaneous speech in how may i help you?" in *Seventh International Conference on Spoken Language Processing*, 2002.
- [11] V. Goel and W. Byrne, "Task dependent loss functions in speech recognition: Application to named entity extraction," in *ESCA Tutorial and Research Workshop (ETRW) on Accessing Information in Spoken Audio*, 1999.
- [12] O. Galibert, S. Rosset, C. Grouin, P. Zweigenbaum, and L. Quintard, "Structured and extended named entity evaluation in automatic speech transcriptions," in *Proceedings of 5th International Joint Conference on Natural Language Processing*, 2011, pp. 518–526.
- [13] J. D. Burger, D. Palmer, and L. Hirschman, "Named entity scoring for speech input," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 1998, pp. 201–205.
- [14] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH*, 2015.
- [15] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *International Workshop on Spoken Language Translation (IWSLT) 2012*, 2012.
- [16] P. Żelasko, P. Szymański, J. Mizgajski, A. Szymczak, Y. Carmiel, and N. Dehak, "Punctuation prediction model for conversational speech," *arXiv preprint arXiv:1807.00543*, 2018.
- [17] O. Tilk and T. Alumäe, "Lstm for punctuation restoration in speech transcripts," in *Sixteenth annual conference of the international speech communication association*, 2015.
- [18] W. Lu and H. T. Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 177–186.
- [19] N. Ueffing, M. Bisani, and P. Vozila, "Improved models for automatic punctuation prediction for spoken and written text." in *Interspeech*, 2013, pp. 3097–3101.
- [20] B. Nguyen, V. B. H. Nguyen, H. Nguyen, P. N. Phuong, T. Nguyen, Q. T. Do, and L. C. Mai, "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)*, 2019, pp. 1–5.
- [21] H. N. Thi Thu, B. N. Thai, H. N. Vu Bao, T. Do Quoc, M. L. Chi, and H. N. Thi 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)*, 2019, pp. 1–5.
- [22] C. C. Aggarwal and C. Zhai, *Mining text data*. Springer Science & Business Media, 2012.
- [23] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," *arXiv preprint cmp-lg/9803003*, 1998.
- [24] H. L. Chieu and H. T. Ng, "Named entity recognition: a maximum entropy approach using global information," in *Proceedings of the 19th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 2002, pp. 1–7.
- [25] A. McCallum, D. Freitag, and F. C. Pereira, "Maximum entropy markov models for information extraction and segmentation." in *Icml*, vol. 17, no. 2000, 2000, pp. 591–598.
- [26] O. Bender, F. J. Och, and H. Ney, "Maximum entropy models for named entity recognition," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 148–151.
- [27] J. R. Curran and S. Clark, "Language independent ner using a maximum entropy tagger," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003, pp. 164–167.
- [28] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003, pp. 188–191.
- [29] S. Sarawagi and W. W. Cohen, "Semi-markov conditional random fields for information extraction," in *Advances in neural information processing systems*, 2005, pp. 1185–1192.
- [30] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuska, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
- [31] Y. Ji, C. Tong, J. Liang, X. Yang, Z. Zhao, and X. Wang, "A deep learning method for named entity recognition in bidding document," in *Journal of Physics: Conference Series*, vol. 1168, no. 3. IOP Publishing, 2019, p. 032076.
- [32] E. Trandafili, E. K. Meçe, and E. Duka, "A named entity recognition approach for albanian using deep learning," in *Complex Pattern Mining*. Springer, 2020, pp. 85–101.
- [33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.
- [34] H. Nguyen, Q. Ngo, L. Vu, V. Tran, and H. Nguyen, "Vlsp shared task: Named entity recognition," *Journal of Computer Science and Cybernetics*, vol. 34, no. 4, pp. 283–294, 2019. [Online]. Available: <http://vjs.ac.vn/index.php/jcc/article/view/13161>
- [35] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *ASRU*, 2011.
- [36] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- [37] K. A. Nguyen, N. Dong, and C.-T. Nguyen, "Attentive neural network for named entity recognition in vietnamese," in *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*. IEEE, 2019, pp. 1–6.