Interspeech 2024
1-5 September 2024, Kos, Greece

VN-SLU: A Vietnamese Spoken Language Understanding Dataset

Tuyen Tran[1], *Khanh Le*[1], *Ngoc Dang Nguyen*[1], *Minh Vu*[1], *Huyen Ngo*[1], *Woomyoung Park*[2], *Thi Thu Trang Nguyen*[1]

[1]Hanoi University of Science and Technology, Vietnam
[2]Naver Corporation, Korea

tuyencbt@gmail.com, khanh.lt2669@gmail.com, nndang2701@gmail.com,
minhducvu1311@gmail.com, huyenthu432002@gmail.com, max.park@navercorp.com,
trangntt@soict.hust.edu.vn

Abstract

Spoken Language Understanding (SLU) is a crucial task in spoken language processing. Despite the availability of numerous English datasets for research, there is a scarcity of resources for low-resource languages like Vietnamese. This paper introduces VN-SLU, the first dataset explicitly designed for Vietnamese SLU. VN-SLU includes 17,321 utterances from 240 Vietnamese speakers, obtained through novel crowd-sourcing methods. We propose a web tool for scenario generation and label validation, ensuring dataset quality and diversity. This tool prompts participants to confirm intents and slot values in smart home and virtual assistant dialogues, ensuring precise alignment. Experimental results highlight the challenging nature of the chosen test set sampling strategy in intent accuracy, SLU-F1, and utterance accuracy. Additionally, we explore the integration of pitch information into the Vietnamese SLU system. Results show improved performance compared to the baseline model.

Index Terms: Spoken language understanding, Vietnamese dataset, smart home

1. Introduction

Spoken Language Understanding (SLU) is the task of enabling machines to understand and interpret spoken language. Specifically, it involves the development of models that can extract meaning and intent from spoken language, enabling computers to interact with humans through voice commands or speech input. In the last decade, the research field of SLU has witnessed remarkable results and development [1, 2, 3, 4].

Throughout the years, there have been various datasets come into existence, setting up the benchmark for researchers to develop and compare SLU models covering many scenarios. While the research of SLU systems is currently advancing rapidly on these benchmarks, there has been very little development on SLU datasets of low-resource languages, such as Vietnamese. In response to this gap, we present the VN-SLU dataset, a resource aimed at innovating Vietnamese spoken language understanding. Our dataset contains two fields which can be used for SmartHome and VirtualAssistant. Inspired by Nguyen et al. [5], VN-SLU stands out by extending the focus to speech data, building on the structure of an NLU dataset, which focuses on text data, to form an SLU dataset. VN-SLU follows both the domain of SmartHome and VirtualAssistant speech commands and is constructed with an efficient crowd-sourcing web tool and a carefully designed recording and annotation strategy. Unlike previous NLU research [5], this strategy ensures authentic and natural data quality, aspects that were previously overlooked. Additionally, our crowdsourcing web tool includes a recording feature to collect speech for SLU models.

The dataset consists of 17,321 utterances from 240 Vietnamese speakers and 20.7 audio hours recording for both SmartHome and VirtualAssistant tasks. With coverage of various challenging recording environments and speaking styles, VN-SLU is expected to be a reliable benchmark for the development of Vietnamese spoken language understanding. Furthermore, we conduct a series of experiments using a state-of-the-art SLU model to create a test set that encompasses a wide range of complexities and challenges from the acquired data.

The rest of the paper is organized as follows. In Section 2, we explain our process of data construction. Section 3 describes the details of our datasets and Section 4 provides our experiments and comparisons of SLU models on our datasets. Lastly, we draw conclusions in Section 5.

2. Dataset Construction

2.1. Pipeline

We propose a construction pipeline consisting of four main steps to build a spoken language understanding dataset. Figure 1 illustrates the steps in the pipeline. The propose pipeline follows two main domain in SLU tasks SmartHome and VirtualAssistant, with adaptability to other domains. The following sections will discuss in detail the methods carried out in each step.

Figure 1: *Propose data construction pipeline.*

2.2. Data preparation

In order to construct a VirtualAssistant dataset of highest richness and precision, the initial focus is placed on the preparation of linguistic data. This foundational step involves a thorough examination not only of the ordinary interactions between users and their devices but also, significantly, of the nuanced exchanges that occur between users and their virtual assistants. Through a process of analysis and augmentation, a comprehensive array of 28 distinct intents and 57 entity values has been identified and incorporated within the dataset.

These intents and entity values represent a diverse spectrum of user-assistant interactions, encompassing a wide range of functionalities and services. From essential tasks like 'weather checking' and 'food ordering' to more specialized actions such as 'seat reserving,' each intent delineates a specific user objective, while each entity value provides pertinent parameters or attributes associated with these goals. We aim to equip virtual

10.21437/Interspeech.2024-1976

assistants with the depth and breadth of knowledge necessary to effectively address user inquiries and cater to their diverse needs.

2.3. Scenario Generation

After preparing the linguistic data, we use a strategy designed to generate slot combinations for every intent. It is required that at the end of the process, the distribution of values within each slot of combinations is balanced. We divide this stage into 2 phases: mapping and combining.

Algorithm 1 Scenario Generation Algorithm

Require:
$min\_device\_occurence$, r_name_d, r_name_l, r_choose_l, $D \leftarrow [d_1, ..., d_n]$, $C \leftarrow [c_1, ..., c_n]$, $L \leftarrow [l_1, ..., l_n]$
Ensure: $combinations$
$occ_D \leftarrow [0, \ldots, 0]$ ▷ *Length of occ_D equal length of D*
$occ_L \leftarrow [0, \ldots, 0]$
$combinations \leftarrow []$
while min of $occ_D \leq min\_device\_occurence$ **do**
  $command \leftarrow random(C)$
  $device \leftarrow$ Choosing random 1 in 3 devices having least occ_D
  if $random() \leq r\_name\_d$ **then**
    $device \leftarrow$ Add name to $device$
  end if
  $comb \leftarrow (command, device)$
  if $random() \leq r\_choose\_l$ **then**
    $location \leftarrow$ Choosing random 1 in 3 locations having least occ_L
    if $random() \leq r\_name\_l$ **then**
      $location \leftarrow$ Add name to $location$
    end if
    $comb \leftarrow (command, device, location)$
  end if
  if $comb \notin combinations$ **then**
    $combinations \leftarrow$ Add $comb$ to $combinations$
    $occ_D[device] \leftarrow occ_D[device] + 1$
    $occ_L[location] \leftarrow occ_L[location] + 1$
  end if
end while
$combinations \leftarrow$ Generate System Number
$combinations \leftarrow$ Generate Temporal Value
return $combinations$

2.3.1. Mapping Phase

During the mapping phase, we identify natural associations between values in different slots. For instance, pairing 'close' in the 'command' slot with 'radio' in the 'device' slot is deemed unnatural, whereas 'close' in 'command' and 'door' in 'device' is more natural. To achieve this, we first specify the pairs of slots to consider, such as "command" and "device", "device" and "location", etc. Second, for each value in the first slot, we determine which specific value in the second slot would naturally complement it. We repeat this process for other slot pairs in order to establish naturally compatible sets of values. To further align with reality scenarios, we introduce system number entities ("target number" and "changing value") and temporal slots ("time at" and "duration") for available commands (e.g., "increase", "decrease") and devices. The mappings of devices to specific slots are summarized as lists for system number values or temporal values.

2.3.2. Combining Phase

In the combining phase, the objective is to generate combinations based on the outcomes of the mapping phase starting with three core parameters: "command", "device", and "location". The Algorithm 1 demonstrates the steps to generate the result combinations where locations and devices are generated first, following with other entities. A percentage of devices and locations, denoted as initial parameters r_name_d and r_name_l, receive added descriptors, like "East light" or "bedroom of Minh,". Additionally, a certain proportion of combinations feature "target number" and "changing value" as system number entities and "time at", "duration" as temporal slots, providing context for more natural requests, closely resembling real-world usage. A part of combinations with one or both temporal slots will be selected according to their associated device values. The algorithm terminates when no device has an occurrence count less than a specified threshold, set to 6 in our dataset.

Furthermore, there are two specific intents for scenes that trigger commands, encompassing a total of 10 scenes (e.g., "party", "coming home") for simplifying device control. These combinations, which feature a primary slot called "scene", have no command value. They include time and location details, meaning they include location and temporal slots in their combinations. These combinations will be generated after the previous algorithm has completed.

2.4. Audio Recording

2.4.1. Crowdsourcing Web Tool

Following the prior work established in [5], we have created a web tool with a structure similar to the previous version, including a requester and a doer side, simulating homeowner-assistant conversations. A key enhancement is the ability for requester to use speech utterances in addition to text, expanding on the text-only functionality of the previous version.

2.4.2. Recording Process

Requester side. Similar to what is done in [5], requests are initiated for the control of devices within a smart home setting. In a conversation, the intent and slot values are selected randomly from generated combinations and presented to the requester. The requester then formulates requests, either (i) with the main intent, lacking some slot values, or (ii) with all slot values in one utterance. The first option is recommended for natural dialogues, but the second remains available. It is essential to explicitly specify the intent first (e.g., "close the device") and the associated slots (i.e., command, device, living_space) in the request before transmitting it. The conversation can be concluded once the requester has supplied all the requisite slot values.

Doer side. The doer, acting as the assistant, is responsible for gathering necessary slot values from the requester to execute commands effectively. First, the requester's intent must be clarified, and then the relevant slot values in the text are selected. During intent determination, the doer ensures the transcript matches the speech utterance, correcting errors as needed (see fig. 2). To confirm a successful turn, both sides must share the same intent and matching slot values in the text. The doer's role also involves asking for missing slot information until all slots are filled.

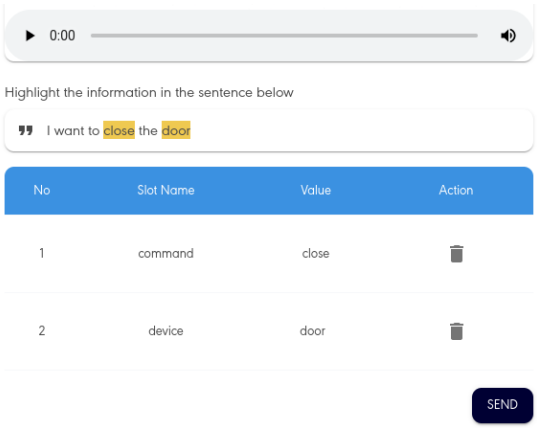

Figure 2: *Doer tags slot values (the applied language version is Vietnamese).*

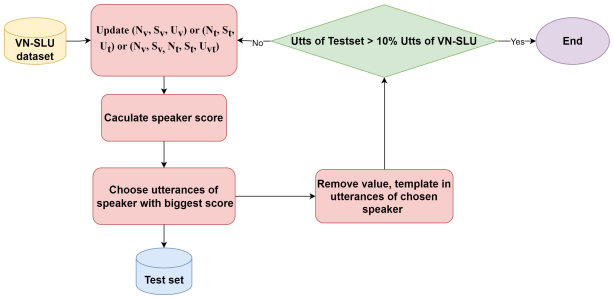

Figure 3: *Our proposal for sampling strategy.*

3. Our VN-SLU Dataset

3.1. Dataset Sampling

$$score_v = \frac{N_v}{U_v}, score_t = \frac{N_t}{U_t}, score_{vt} = \frac{N_v + N_t}{U_{vt}} \quad (1)$$

1. N_v: The number of distinct entity value in utterances of speaker not present in test set
2. N_t: The number of distinct sentence template in utterances of speaker not present in test set
3. U_v, U_t, U_{vt}: The order is the number of utterances for each speaker in algorithm calculated according to $score_v$, $score_t$, $score_{vt}$

Our algorithm choice test set is described in fig. 3.

From the input dataset, we have sets S_v consisting of distinct values not present in the test set, S_t consisting of templates of utterances not present in the test set, and S_{vt} consisting of both distinct values and distinct templates not present in the test set. The algorithm proceeds as follows:

Step 1: Compute the speaker score for each speaker using one of three formulas: $score_v$, $score_t$, $score_{vt}$.

Step 2: Select the speaker with the highest score and move all their utterances from the dataset to the test set.

Step 3: Remove all values and/or templates, which are in the moved utterances, from the sets S_t, S_v, and S_{vt}.

Step 4: Check the size of the test set. If it is greater than 10% of the dataset, terminate the algorithm. Otherwise, proceed to Step 5.

Step 5: Update N_v, N_t, U_v, U_t, U_{vt}, then go back to Step 1.

This algorithm aims to ensure that the test sets contain a diverse set of speakers not seen during training, as well as a broad range of unique entity and template values, maximizing the coverage and representativeness of the evaluation data.

3.2. Experiment on Sampling Strategies

We use Speechbrain-HuBERT[6] model and Nemo-Conformer-Transformers[7] to training datasets of each sampling strategy. The experimental results are described in Table 1.

Table 1: *Experimental results on different sampling strategies*

Test set	Intent Acc	SLU-F1	Utt Acc
Speechbrain - HuBERT Model			
Speaker - Value (1802 Utts)	**90.46**	**62.09**	**34.4**
Speaker - Template (1818 Utts)	92.65	93.49	83.16
Speaker-Value-Template (1832 Utts)	92.25	89.69	79.46
Nemo - Conformer - Transformer Model			
Speaker - Value (1802 Utts)	**94.06**	**70.36**	**42.11**
Speaker - Template (1818 Utts)	95.77	96.61	89.71
Speaker-Value-Template (1832 Utts)	95.44	94.03	87.42

Table 2: *Dataset statistics*

Property	Training Set	Test Set
# of Utterances	14,867	2,454
# of Hours	17.36	3.36
# of Speakers	172	68

3.3. Experimental Result

The experimental results in Table 1 show that the Speaker-Value method yields the lowest results for both participating models, indicating that this test set is the most challenging and effectively highlights model performance. Additionally, since during the experimentation of the test set selection methods, the common training set used was the remaining part of the dataset after removing the utterances in the 3 test sets, we will resample the dataset using the Speaker-Value method to optimize the dataset. The statistics regarding the train and test sets are summarized in Table 2.

3.4. VN-SLU Dataset

Following the data construction phase, we unveil the VN-SLU dataset, a comprehensive Spoken Language Understanding (SLU) corpus tailored specifically for the Vietnamese language within the domain of VirtualAssistant technology. The VN-SLU repository comprises an impressive assemblage of 17,321 distinct utterances sourced from a cohort of 240 proficient Vietnamese speakers, collectively representing an extensive corpus of 20.07 hours of speech data. This dataset encapsulates a diverse array of linguistic expressions and user interactions, organized into 28 distinct VirtualAssistant intents, each annotated with 57 corresponding entity values. VN-SLU is publicly available for researchers under a GitHub repository[1].

Furthermore, despite being a dataset for a low-resource language such as Vietnamese, VN-SLU shows commendable quality and quantity when compared with other widely-used datasets like SLURP [8], Snips [9] and FSC [10]. The detailed statistics are illustrated in Table 3.

[1]https://github.com/tuyen-tran1/VN-SLU

Table 3: *Comparison with several existing SLU datasets*

Name	Lang.	# Spkrs.	# Utts.	# Hrs.	Avg. Len [s]	Avg. # Utts/Intent	Avg. # Entity Values/Utt
SLURP [8]	Eng.	177	72,277	58.0	2.9	778	1.07
Snips [9]	Eng.	69	5,886	5.5	3.4	321	1.17
FSC [10]	Eng.	97	30,043	19.0	2.3	969	2.34
VN-SLU	**Vie.**	**240**	**17,321**	**20.1**	**4.2**	**618**	**2.00**

Table 4: *Experimental result on VN-SLU dataset*

Model	Pretrained	Intent Acc (%)	SLU-f1 (%)	Utt Accuracy (%)
Speechbrain HuBERT[6]	hubert-base-ls960	86.47	56.28	30.28
Speechbrain Wav2vec2[6]	Vietnamese-ASR	90.95	65.91	35.78
Nemo-Conformer-Transformer-large[7]	**NeMo ASR-Set**	**91.81**	**67.82**	**37.57**

Table 5: *Comparison results of the propose model with those of the baseline model*

Model	Intent Acc	SLU-F1	Utt Acc
Speechbrain HuBERT[6]	86.47	56.28	**30.28**
Our Propose Model	**88.14**	**58.47**	28.65

4. Experiment

4.1. Model Architecture

We incorporate a RNN-based attention decoder [11] to directly extract semantic information from the Hubert encoder [12]. The model undergoes training on a base one with 96 million parameters. The pretrained model with semi-supervised learning (SSL) hubert-large-ls960 datasets, respectively. The second architecture we employed is an end-to-end model within the NVIDIA NeMo Framework specifically designed for the Speech Intent Classification and Slot Filling tasks. This architecture is also built based on an Encoder-Decoder framework, consist of a Conformer-large module [13] serving as the encoder to extract features, and a Transformer Decoder [14] utilized for predicting semantic information. In both two approaches, both intents and slots are considered as a sequence-to-sequence automatic speech recognition (ASR) task and are decoded using the attentional decoder. The models's performance is assessed by computing the Negative Log-Likelihood (NLL) loss during character-level token generation.

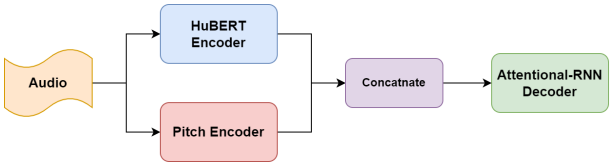

Figure 4: *Propose model with pitch extraction*

In tonal languages, pitch characteristics significantly impact the meaning of sentences. This observation drives our efforts to enhance model performance through the integration of pitch information. Therefore, we aim to explore the potential of integrating pitch extraction into the Speechbrain-HuBERT model[6] using the VN-SLU dataset. Our integrated experiment has yielded noteworthy improvements, emphasizing the importance of incorporating pitch details. The model architecture is described in fig. 4.

We employ the normalized cross-correlation function and median smoothing (NCCF) to extract pitch features from audio. Subsequently, these pitch features are encoded using an encoding block comprising two CNN stacks and two RNN stacks. The outputs of the Hubert Encoder and the Pitch Encoder are concatenated and passed to the decoder for generating the intent and entities of the utterance.

4.2. Metrics and Experimental Result

Following the work of [8], we use SLU-F1 to evaluate entity accuracy. Moreover, to evaluate the overall accuracy of intent and entity recognition, we derive a new metric - Utterance Accuracy, which computes the final accuracy by calculating the number of utterances with correct prediction in both intents and entities.

The experimental results of the models are summarized in Table 4. The Nemo Conformer - Transformer (End-to-End SOTA)[7] model achieved the best results with an intent accuracy of 91.81% and an SLU-F1 score of 67.82%. However, the utterance accuracy score remains relatively low at 37.57%. Moreover, the use of a pretrained model in the Vietnamese language context also demonstrates the significant impact of language in the SLU task. When utilizing the Speechbrain-Wav2vec2[6] model pretrained on Vietnamese data, we achieved impressive results compared to the state-of-the-art model using English pretrained models.

Additionally, from Table 5, we can observe the effectiveness of integrating pitch information into the model, as indicated by the increase in intent accuracy and SLU-F1 scores of 1.67% and 2.19%, respectively, compared to the baseline model[6]. This demonstrates the impact of pitch on the performance of SLU models for Vietnamese.

5. Conclusion

This paper releases VN-SLU, the first dataset explicitly designed for Vietnamese spoken language understanding. VN-SLU comprises 17,321 utterances from 240 Vietnamese speakers in smart home conversations. Constructed with an advanced crowd-sourcing pipeline, VN-SLU features an effective scenario generation algorithm and a dedicated web tool for precise audio recording and labeling alignment. Experimental results highlight the challenging nature of the chosen test set sampling strategy in terms of intent, SLU-F1, and utterance accuracy. Additionally, our propose model shows improved results, demonstrating the influence of pitch on Vietnamese SLU models.

6. Acknowledgements

This work was supported by the NAVER Corporation within the framework of collaboration with the International Research Center for Artificial Intelligence (BKAI), School of Information and Communication Technology, Hanoi University of Science and Technology under Project NAVER.2022.DA08.

7. References

[1] L. Qin, T. Xie, W. Che, and T. Liu, "A survey on spoken language understanding: Recent advances and new frontiers," *arXiv preprint arXiv:2103.03095*, 2021.

[2] N. Tomashenko, A. Caubrière, and Y. Estève, "Investigating adaptation and transfer learning for end-to-end spoken language understanding from speech," in *Interspeech 2019*. ISCA, 2019, pp. 824–828.

[3] S. Rongali, B. Liu, L. Cai, K. Arkoudas, C. Su, and W. Hamza, "Exploring transfer learning for end-to-end spoken language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 754–13 761.

[4] S. Thomas, H.-K. J. Kuo, B. Kingsbury, and G. Saon, "Towards reducing the need for speech training data to build spoken language understanding systems," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7932–7936.

[5] T. T. T. Nguyen, T. D. A. Dang, Q. V. Vu, and W. Park, "Building vietnamese conversational smart home dataset and natural language understanding model," *Interspeech 2022*, pp. 5180–5184, 2022.

[6] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *ArXiv*, vol. abs/2111.02735, 2021.

[7] H. Huang, J. Balam, and B. Ginsburg, "Leveraging pretrained asr encoders for effective and efficient end-to-end speech intent classification and slot filling," *ArXiv*, vol. abs/2307.07057, 2023.

[8] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "SLURP: A spoken language understanding resource package," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 7252–7262. [Online]. Available: https://aclanthology.org/2020.emnlp-main.588

[9] A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril *et al.*, "Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces," *arXiv preprint arXiv:1805.10190*, 2018.

[10] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *ArXiv*, vol. abs/1904.03670, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:102352396

[11] A. K. Das, A. Al Asif, A. Paul, and M. N. Hossain, "Bangla hate speech detection on social media using attention-based recurrent neural network," *Journal of Intelligent Systems*, vol. 30, no. 1, pp. 578–591, 2021.

[12] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[13] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.