Building Vietnamese Conversational Smart Home Dataset and Natural Language Understanding Model

Nguyen Thi Thu Trang\textsuperscript{1}, Dang Trung Duc Anh\textsuperscript{1}, Vu Quoc Viet\textsuperscript{1}, Park Woomyoung\textsuperscript{2}

\textsuperscript{1}Hanoi University of Science and Technology
\textsuperscript{2}Naver Corporation

trangntt@soict.hust.edu.vn, viet.vq151101@gmail.com, ducanhhbtt@gmail.com, max.park@navercorp.com

Abstract

Natural Language Understanding (NLU), which includes intent detection and slot tagging, plays an important role in any dialog system. This paper aims at building a first-ever conversational smart home dataset SmartNLU and NLU model for Vietnamese. Raw data were collected by asking participants provide or confirm the intents and slot values in the user utterances that they sent or received in a smart home conversation until all were matched, using a Wizard-of-Oz set-up of a web tool. The dataset has been released for a challenge carried out in AIHub\textsuperscript{1}, and published for the community. Several state-of-the-art joint NLU models were experimented on the released dataset. The proposed NLU model, which added PhoBERT to the DIET architecture of Rasa framework, gave the best results. The sentence accuracy of the DIET+PhoBERT was considerably higher than (i.e. 4.3\% to 11.7\%) the one of others. 

Index Terms: dialog system, natural language understanding, slot tagging, entity recognition, smart home

1. Introduction

Dialog systems have been becoming more and more popular due to their promising real-life applications. It is undeniable that Natural Language Understanding (NLU) plays a massive role in the success of these kinds of systems. In particular, the two main tasks in NLU are intent classification and slot tagging. The goal of intent classification is to categorize users' intent, while that of slot filling is to extract from that utterance correct entities for slots of the intent.

Despite being the 17th most widely spoken language, Vietnamese is still considered a low-resource language when it comes to NLU research. To the best of our knowledge, apart from the recently published PhoATIS \textsuperscript{1} [1], a translation of ATIS \textsuperscript{2} [2] provided by VinAI research, there is only one more Vietnamese dataset that are relevant to NLU \textsuperscript{3}. However, it is not publicly available for research.

Even with the publication of new dataset as well as progress in model architecture, there are still several remaining problems for Vietnamese NLU. Firstly, due to the limitations of the translation process as well as the imbalanced nature of PhoATIS \textsuperscript{1}, it is difficult to evaluate the effectiveness of NLU models using only this dataset. Secondly, the performance increase of recently published models over existing ones is not significant.

As a result, in this paper, we have built a conversational dataset in smart home domain as well as proposed a NLU model for Vietnamese language. The dataset has been released for a challenge carried out in AIHub\textsuperscript{1}, and published for the community\textsuperscript{2}.

The rest of the paper is organized as follows. Sections 2 and 3 present the data collection and cleaning process as well as the NLU models used to convey the experiments. The next section goes in depth into the data generating and sampling techniques that we conducted to generate the dataset. In section 5, the experimental results and evaluation are reported and discussed. Finally, we draw some conclusions of this work in section 6.

2. Data Collection & Cleaning

Utilizing the Wizard-of-Oz set-up \textsuperscript{4}, large multi-turn dialogue corpora can be created with the contribution of the participants. The scenario that we chose was a conversation between a smart home system and its owner, but our web tool can be altered to fit any scheme. There will be one user on each side of the conversation, one acts as the smart home system (doer) and the other acts as the house owner (requester).

2.1. Requester side

The goal of the requester (i.e. the house owner) is to request to control devices in a smart home. In a conversation, the intent of the user and slot values are assigned randomly and displayed to the requester (Figure 1). The requester can provide an utterance to ask for controlling devices in any option: (i) Provide the main intent lacking of some of slot values, e.g. “I want to change the light to blue!”, (ii) Provide all the slot values in one utterance, e.g. “Please change the light in the 2nd-floor bedroom to blue for me!”. To create natural and flexible dialogues, first option is recommended most of the time but still use the second one whenever they see fit. The requester needs to specify the intent (e.g. set color) and slots (i.e. command, device, color) in the user request before sending it. The conversation can be terminated if the requester provides all required slot values.

2.2. Doer side

The doer side (i.e. the smart home system) has to ask the user on the other side about the necessary information (i.e. slot values) for doing the command, e.g. which device to turn on/off, the room and floor where the device is at. The doer side cannot see slot values but can see which slot need to be collected. When receiving an utterance from the requester, the doer need to select

\textsuperscript{1}https://aihub.vn/competitions/207
\textsuperscript{2}https://github.com/lab914hust/SmartHomeNLU
the intent of the requester and provide slot values in the text (Figure 2). If both sides give the same intent and slot values in the text, the turn is confirmed. The doer continues to ask for lacking slots until all required slots are filled.

2.3. Data Cleaning

The first step of our data cleaning process is manually removing unnecessary sentences from the dialogue corpora collected. For example, occasionally, users send a message with typo and then send another message to correct it. In this case, the sentence with typo has to be removed. Utterances that do not convey enough meaning for intent detection, i.e., requiring a few more turns in the dialogue for sufficient information, are also removed. This leaves us with 644 utterances. Slot values of all sentences (including ones removed in the previous step) are extracted, and only sentences reserved are used as sentence templates, thus creating a slot value set and an intent template set.

To create user requests for training and testing, we need to fill these slot values into the template, similar to what was done in [5], as well as conduct data sampling to ensure the diversity as well as the balance of the dataset. Details on how we do so will be provided in Section 3, however, we will also publish the dataset before these steps so that researchers are able to explore these techniques as they please.

To create train, validation and test set, we divide the slot value set and sentence template set into three parts. To be specific, the sentence templates are partitioned with ratio of 6/2/2 for train/validation/test, respectively. With regards to slot values, the train and validation sets contain 60% of the total number each, while test set includes all of the them. However, we ensure that 20% of all slot values will not appear in either the train or the validation set. As a consequence, the test set requires generalization of the model to able to detect correctly classified novel words. The statistics for the numbers of slot values and intent templates are provided in Table 2 and Table 1 respectively.

3. Data Generating and Sampling Strategies

In this part, we discuss the generating and sampling techniques conducted on train, validation and test sets. We want these techniques to ensure the variety of the dataset with regards to slot value as well as the balance with regards to intent.

The first technique we think of is to fill slot values into intent templates, which we call “Full Filling”. All values of the same slot type were filled into slots in a template. However, this creates too many data samples, i.e. 80,000 generated sentences from 644 utterance templates. Not only that, the generated dataset was also heavily imbalanced. To deal with these problems, different data sampling strategies are conducted. These include Down Sampling and Over Sampling [6]. In Down Sampling we randomly remove sentences from the majority class while in Over Sampling, not only we do so but also the opposite, randomly duplicate data samples from minority class.

Another method we come up with to handle the redundancy and imbalance of the dataset is what we call Slot Filling and...
Sampling. Specifically, each template will appear at most 10 times. For each chosen intent template, values for all slot types are chosen in a round robin manner. This will ensure the frequency of appearance of all slot values to be roughly equal.

All of the methods mentioned above are carried out on utterances of intents that have fewer than 300 data samples. Because of the limited size of validation and test sets, we decided that it’s best to conduct filling and sampling these two sets. In later sections, we will conduct experiments to choose which out of the two data generation techniques to apply to the validation and test set.

The statistics of the dataset before and after using the generating and sampling techniques are provided in Table 4.

### 4. Vietnamese NLU Model

In this section, we present various NLU models, including state-of-the-art NLU models for Vietnamese and other languages. We also propose to use PhoBERT [7] as a language model for Vietnamese in a Transformer-CRF model [8].

#### 4.1. Finetuned BERT

BERT [9] has proven to be a strong baseline in many types of Natural Language Processing tasks. Its architecture contains multiple layers of bidirectional Transformer [10] Encoder. Input tokens are presented using a concatenation of WordPiece [11] embeddings, positional embedding and segment embedding. A special token specifically reserved for sentence classification tasks, denoted [CLS], is added at the beginning of the input sequence and another special token, denoted as [SEP], is appended to the end.

In our experiment, we finetuned PhoBERT [7], a large pre-trained word embedding exclusively trained on Vietnamese text using RoBERTa [12] architecture, for intent detection and slot filling independently. The output of the [CLS] token is used for intent detection, and those of other tokens are used for slot filling.

#### 4.2. JointBERT-CRF

The recent success of BERT, which breaks state-of-the-art in many NLP tasks simply by fine-tuned, leads to [13] experimenting the capability of this model on intent detection and slot filling. To the best of our knowledge, their research was one of the first to explore BERT [9] for NLU.

The JointBERT-CRF model [13] solved both tasks jointly. Input sequence is fed to a pre-trained BERT, output of the [CLS] token is fed to a fully connected linear layer to predict the intent of input sentence while those of other tokens are fed to a linear-chain CRF for predicting their slot types.

#### 4.3. JointIDSF

Extending the work of [13], JointIDSF [1] incorporated the intent context information into the slot tagging process to enhance the performance via an Intent-Slot attention layer. The intent prediction vector is fed to the Intent-Slot attention, the output of which is then concatenated with BERT-embedded vectors of word tokens. Those vectors are then fed to the linear-chain CRF for slot prediction, similarly to what we saw in JointBERT-CRF [13].

#### 4.4. DIET

DIET [8] is the joint model that Rasa proposed to solve both intent detection and slot filling in one model and achieved great results despite being lightweight and inexpensive to train. DIET uses hard parameter sharing to train two tasks simultaneously. Together with pre-trained word embedding, input tokens are featureized with what called Sparse Features, containing token level one-hot encodings and multi-hot encodings of characters’ n-grams. The addition of Sparse Feature is what allows DIET [8] to be used for a wide variety of languages, including low-resource ones that do not have any large pre-trained word embeddings.

For the main architecture, DIET [8] uses two Transformer [10] layers to extract semantic information from the input sequence. The output is then fed to a CRF layer to make predictions on slot types of tokens as well as intent of the sentence.

#### 4.5. DIET+PhoBERT

In this paper, we propose to use PhoBERT [7] as a pretrained word embedding model in DIET [8] for Vietnamese. The output of PhoBERT was combined with sparse features to leverage the quality of the model. The architecture of DIET combined with PhoBERT is illustrated in Figure 3.

We have conducted two training strategies: (i) DIET+PhoBERT and DIET+PhoBERT Finetuning. With DIET+PhoBERT, we used the trained PhoBERT directly when training DIET. With ‘DIET+PhoBERT Finetuning’, based on the idea of JointIDSF [1], we used the parameters learned through JointBERT-CRF [13] to continue training DIET-PhoBERT. We expected that this training strategy will further improve the performance of this model in the two NLU tasks.

### 5. Experiments

#### 5.1. Sampling Strategies for Training Set

As mentioned in Section 3, we conduct experiment to determine the data generation technique to apply to Validation and Test set. The results provided in Table 5 suggest that Slot Sampling creates a more challenging dataset. Out of the four sampling strategies applied to training set, three of them achieved lower sentence accuracy (a sentence is said to be correctly classified if both the intent as well as all of the slots are correctly predicted) on Slot Sampling Validation set than on Full Sampling Validation set. Thus, we choose this method to apply to both Test and Validation set.

To further evaluate the effectiveness of the data generating and sampling methods, we compare the results of JointBERT-CRF model on validation set. The metrics we use are intent accuracy, slot F1 score and sentence accuracy. The results provided in Table 4 suggests that Slot Filling and Sampling is superior to all other methods. Although the size of training set

<table>
<thead>
<tr>
<th>Dataset</th>
<th># utterances</th>
<th># slots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
<td>1,198</td>
<td>3,004</td>
</tr>
<tr>
<td>Validation</td>
<td>1,176</td>
<td>2,744</td>
</tr>
<tr>
<td>Full Filling Train</td>
<td>85,708</td>
<td>121,575</td>
</tr>
<tr>
<td>Full Filling + Over Sampling</td>
<td>3,169</td>
<td>7,673</td>
</tr>
<tr>
<td>Full Filling + Down Sampling</td>
<td>2,899</td>
<td>7,673</td>
</tr>
<tr>
<td>Slot Filling &amp; Sampling Train</td>
<td>3,492</td>
<td>9,090</td>
</tr>
</tbody>
</table>

**Table 3:** Statistics of training set after Filling and Sampling procedures and validation and test sets
Figure 3: Architecture of the proposed DIET+PhoBERT model

Table 4: Result of JointBERT-CRF model when various data generation and sampling strategies are applied to Training set (row) and Validation set (column)

<table>
<thead>
<tr>
<th>Sampling strategy</th>
<th>Slot Sent acc</th>
<th>Full Sent acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Filling</td>
<td>61.05</td>
<td>64.29</td>
</tr>
<tr>
<td>Full Filling + Over Sampling</td>
<td>58.93</td>
<td>61.73</td>
</tr>
<tr>
<td>Full Filling + Down Sampling</td>
<td>62.33</td>
<td>61.99</td>
</tr>
<tr>
<td>Slot Filling &amp; Sampling (*)</td>
<td>63.43</td>
<td>64.46</td>
</tr>
</tbody>
</table>

Table 5: Result on the validation set of JointBERT-CRF model with training set generated by different strategies

<table>
<thead>
<tr>
<th>Sampling strategy</th>
<th>Intent acc</th>
<th>Slot F1</th>
<th>Sent acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Filling</td>
<td>74.57</td>
<td>92.98</td>
<td>61.05</td>
</tr>
<tr>
<td>Full Filling + Over Sampling</td>
<td>72.53</td>
<td>93.77</td>
<td>58.93</td>
</tr>
<tr>
<td>Full Filling + Down Sampling</td>
<td>76.19</td>
<td>93.10</td>
<td>63.33</td>
</tr>
<tr>
<td>Slot Filling &amp; Sampling (*)</td>
<td>76.36</td>
<td>94.26</td>
<td>63.43</td>
</tr>
</tbody>
</table>

is much larger than that of other sampling strategies, the result that Full filling achieved was not great, almost 2% lower than the best strategy in all three evaluation metrics. Down sampling was able to reduce the impact of imbalance and redundant, low-quality data. On the contrary, Over Sampling decreased the model’s effectiveness, leading to the worst result of all 4 data filling and sampling strategies.

5.2. NLU Model Comparison

From the result on test set in Table 6, we can see that all of the models used achieved roughly 87% Intent accuracy, 95% Slot F1 and 76% Sentence accuracy. DIET without the aid of pretrained Language Model achieved fairly low result on all 3 metrics compared the other models, however, when it combines PhoBERT and PhoBERT finetuned on JointBERT-CRF[13] with Sparse feature, the figures for all of these metrics improved significantly. Specifically, adding PhoBERT finetuned increased Intent accuracy by more than 4%, Slot F1 by roughly 2.5% and the Sentence accuracy rose by more than 10%, which demonstrates the effectiveness of pretrained Language Models. With regards to PhoBERT finetuned, this model achieved the highest result on Intent accuracy but performed the worst for Slot Filling task. This further backs up the finding by

<table>
<thead>
<tr>
<th>Model</th>
<th>Intent acc</th>
<th>Slot F1</th>
<th>Sent acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIET-PhoBERT-Finetune (*)</td>
<td>89.98</td>
<td>97.33</td>
<td>82.72</td>
</tr>
<tr>
<td>DIET-PhoBERT (*)</td>
<td>86.73</td>
<td>97.56</td>
<td>78.46</td>
</tr>
<tr>
<td>DIET</td>
<td>85.73</td>
<td>94.83</td>
<td>71.03</td>
</tr>
<tr>
<td>JointIDSF</td>
<td>88.90</td>
<td>94.91</td>
<td>75.54</td>
</tr>
<tr>
<td>JointBERT-CRF</td>
<td>88.90</td>
<td>95.27</td>
<td>76.63</td>
</tr>
<tr>
<td>PhoBERT-Finetune (Separate)</td>
<td>91.24</td>
<td>94.50</td>
<td>76.04</td>
</tr>
</tbody>
</table>

[14] that utilizing the relationship between these two tasks helps to improve the prediction of models.

6. Conclusions

We have proposed a pipeline to build a conversational dataset for Vietnamese. We also release SmartNLU, a first-ever conversational smart home dataset, for a challenge carried out in AIHub, and for the community. Raw data were collected by asking participants to play as a requester or a doer in a smart home conversation of 10 intents with 7 slots using a Wizard-of-Oz set-up of a web tool. The data were then cleaned and processed to build templates of user utterances with empty slots. The Slot Filling Sampling strategy, which filled all slots by the round-robin algorithm to templates, was empirically chosen to generate user utterances from collected templates. This strategy helped to remove the redundancy and imbalance of the dataset but remained its’ diversity and quality. The final dataset includes 3,492/1,176/1,198 sentences correspondingly for the training/validation/test sets. We did experiments on several state-of-the-art joint NLU models with the released dataset. The proposed NLU model, which added PhoBERT to the DIET architecture of Rasa framework, outperformed the joint others with improvements from 4.3% to 11.7% in term of sentence accuracy.

Acknowledgments

This work is supported by Naver Corporation and School of Information and Communication Technology, Hanoi University of Science and Technology.
7. References


