A TWO-STAGE SYSTEM FOR SPOKEN LANGUAGE UNDERSTANDING

ShileiMiao, PeijiaQian，GaoshengZhang, LinghuiTang

Shenzhen Transsion Holdings Co., Ltd, Shanghai Branch, China
{gaosheng.zhang, shilei.miao, linghui.tang, peijia.qian}@transsion.com

ABSTRACT

Spoken Language Understanding(SLU) typically refers to intent determination and slot filling of input speech utterance. In this paper, we propose a Two-Stage system for SLU, which consists of Automatic Speech Recognition(ASR) tasks and Natural Language Understanding(NLU) tasks. In the first stage, we use a model based on the encoder-decoder structure to recognize the speech utterance into text. In the second stage, we combine Bidirectional Encoder Representations for Transformers(BERT) and Conditional Random Field(CRF) to train intent determination and slot filling jointly. In addition, we compress the data using Byte Pair Encoding(BPE) and construct the BERT word list by pre-training to save a large number of parameters, thus extending the model depth and obtaining an Exact Match boost. The final experiments demonstrate the effectiveness of our proposed model.

Index Terms— Spoken language understanding, automatic speech recognition, BERT, conditional random field

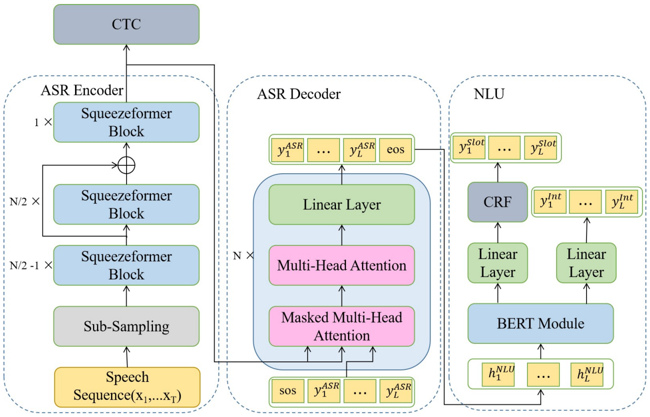
CTC
ASR Encoder
1 ×
Squeezeformer Block
N/2 ×
Squeezeformer Block
N/2 -1 ×
Squeezeformer Block
Sub-Sampling
Speech Sequence(x_1,...x_T)
ASR Decoder
y_1^{ASR} … y_L^{ASR} eos
Linear Layer
N ×
Multi-Head Attention
Masked Multi-Head Attention
sos y_1^{ASR} … y_L^{ASR}
NLU
y_1^{Slot} … y_L^{Slot}
CRF
y_1^{Int} … y_L^{Int}
Linear Layer
Linear Layer
BERT Module
h_1^{NLU} … h_L^{NLU}

Fig. 1. Two-Stage SLU system. The first stage contains the encoder and decoder for ASR. The second phase contains the intent determination and slot filling for the NLU task and the main method is based on BERT and CRF. N denotes the number of blocks in the modules.

1. INTRODUCTION

We propose a Two-Stage Spoken Langauge Understanding (SLU) system that separates the Automatic Speech Recognition(ASR) tasks in SLU from the Natural Language Understanding(NLU) tasks. The ASR and NLU tasks in our two-stage SLU system can perform separate pre-training tasks, allowing performance monitoring of individual components.

2. MODEL

The framework of our Two-Stage SLU system is shown in Fig.1. The first stage is the ASR module, which mainly consists of two network structures, encoding and decoding. The second stage is the NLU module with two tasks of intent determination and slot filling.

For the first stage, an encoder-decoder-based ASR is constructed to recognize speech utterance. The encoder part extracts features from the input speech sequences, and the attention-based decoder is an autoregressive model on the target text sequences, while the output encoding information of the encoder is obtained by attention during the autoregressive calculation, so that the information of the input sequences can be utilized. Connectionist Temporal Classification(CTC) [1] is another decoder network that performs feed-forward computation directly on the encoder output.

For the second stage, given the sequence of utterance output by the ASR stage, the NLU task is used to determine intent and to fill corresponding slots for different intents. The Bidirectional Encoder Representations for Transformers(BERT) [2] module and Conditional Random Field(CRF) [3] are the main components of this part of the model. After encoding the sequence information by BERT, the outputs are used for slot filling learning and intent determination respectively. On the one hand, the hidden state of the last layer output is used as input to CRF to learn label dependencies, and on the other hand, the semantic feature vector representation of the output is used for intent determination.

2.1. ASR task

2.1.1. Encoder

The encoder part of ASR is used to extract information from the input speech sequence. We first use the 1-D Convolutional Neural Networks to downsample the input speech se-

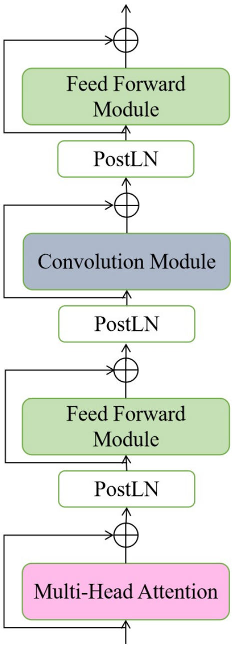

Fig. 2. The architecture of Squeezeformer Block ,which only uses Post-Layer Normalization and adjusts the module order.

quence, merging consecutive frame information. Then we use the Squeezeformer proposed by Kim et al.[4] as the encoder block. The details of Squeezeformer Block are shown in Fig.2. The squeezeformr model makes some improvements based on the conformer structure. The main contributions are: 1) the U-Net structure is used to compress the middle layer of the network and reduce the cost of Multi-Head Attention Module on long sequences; 2) the module structure is adjusted by removing redundant Pre-Layer Normalization layers and replacing the order of Multi-Head Attention and Feed Forward Module.

2.1.2. Decoder

Both CTC and attention-based decoders are combined in the decoding process. Firstly, decoding is performed using CTC, which gives output for each input frame, but this approach cannot construct relationships between texts. Secondly, the N-best result output from CTC is then re-scored using attention, which does not require output for every input frame, but the next output based on the whole input sequence information and the already output information.

2.1.3. Loss functions

The combination of CTC loss and Attention loss constitutes the loss for the ASR task. CTC loss is calculated based on the output of the encoder, and attention loss is based on the output of the decoder to calculate the cross-entropy loss of the model output probability and sample labels at each location. We denote the CTC loss as L_{ctc} and the attention loss as $L_{attention}$. Thus, the final loss is defined as:

$$L_{ASR}(x,y) = \lambda L_{CTC}(x,y) + (1-\lambda)L_{attention}(x,y) \quad (1)$$

2.2. NLU task

2.2.1. Intent determination

Intent determination is essentially a classification task. In this paper, we first pre-train the dataset using BERT, i.e., masked language model and next sentence prediction. Second, we implement the intent classification task by fine-tuning. Based on the hidden state h_1 of the first special token [CLS] output by the last layer of BERT, we can predict the intent as:

$$y^i = softmax(W^i h_1 + b^i) \quad (2)$$

2.2.2. Slot filling

The CRF module utilizes the hidden state of the BERT's last layer output for the slot filling task. CRF treats each point on the sequence as a whole when performing sequence annotation, instead of individual points, and the annotation results of each point are somewhat dependent on the path as a unit for training. For the input sequence$x = (x_1, x_2, ..., x_n)$, the corresponding tag labels are $y = (y_1, y_2, ..., y_m)$, where there are m tag categories and specify $y_0 = START\_TAG$, $y_{n+1} = STOP\_TAG$. The purpose of the conditional random field is to compute the posterior of the label sequence y given the input x:

$$P(y|x) = \frac{exp(score(x,y))}{\sum_{all\ possible\tilde{y}} exp(score(x,\tilde{y}))} \quad (3)$$

where the score is defined as follows:

$$score(x,y) = \sum_{i=0}^{n} transitions_{y_{i+1}y_i} + \sum_{i=1}^{n} feats_{i,y_i} \quad (4)$$

where feats is the emission matrix output by model and transitions is the matrix with initialized size (m, m) that follows the model for iteration.

2.2.3. Loss function

In the training process of NLU task, the loss function consists of intent determination and slot filling together. We denote y^{int} as the distribution over the intent labels and l^{int} as the ground truth label of intent. Cross-entropy loss is used for the intent:

$$L_{intent} = logy^{int}(l^{int}) \quad (5)$$

The loss function for slot is calculated based on the output of CRF, which is defined as:

$$L_{slot} = -log\frac{exp(score(x,y))}{\sum_{all\ possible\tilde{y}} exp(score(x,\tilde{y}))} \quad (6)$$

Thus, the total loss for NLU is defined as:

$$L_{NLU} = \alpha_1 L_{intent} + \alpha_2 L_{slot} \tag{7}$$

where α_1 and α_2 are the custom coefficients of L_{intent} and L_{slot}, respectively.

2.2.4. Joint inference

We take joint inferences of Intent Determination and Slot Filling on text. Intent Determination classifies the primary intent of the text, and Slot Filling identifies the slots and the intents nested within them. For example, for the text ”Are there sales events at the local mall”, the label ”[IN:GET EVENT Are there [SL:CATEGORY_EVENT sales events] at [SL:LOCATION [IN:GET_LOCATION the [SL:LOCATION_MODIFIER local] [SL:CATEGORY_LOCATION mall]]]]”, the Intent Determination classifies it as ”GET_EVENT”, and the identification result of Slot Filling is
”['O',
'O',
'CATEGORY_EVENT',
'CATEGORY_EVENT',
'O',
'LOCATION-GET_LOCATION',
'LOCATION-GET_LOCATION-LOCATION_MODIFIER',
'LOCATION-GET_LOCATION-CATEGORY_LOCATION']”

3. EXPERIMENT

3.1. Dataset

In addition to the STOP(Spoken Task Oriented Parsing) dataset [5] specified for the competition, we added the additional publicly available speech dataset LibriSpeech [6] when pre-training the ASR model.

STOP Dataset The dataset covers 8 different domains: alarm, event, messaging, music, navigation, reminder, timer and weather. The training set contains 120929 utterances, the evaluation set contains 33387 utterances, and the test set contains 75640 utterances. We set each nested slot relationship to a label. The number of slot labels is 316 and the intent has 69 different types.

Librispeech Dataset This dataset is a large corpus containing 960 hours of English speech. The data comes from the audiobooks of the LibriVox project.

3.2. Evaluation Metrics

The Exact Match(EM) between the predicted output and the annotated parse is used to evaluate model effects.

3.3. Training Details

Both the ASR and NLU pre-training tasks use Byte Pair Encoding(BPE) [7] to compress the text data. In the ASR-trained BPE dictionary, there are a total of 5000 subwords. The word list size after pre-training by BERT using BPE is 3155. Audio features are 80-dim log-mel filterbank features computed over a 25ms window, with 10ms shifts.

In the first stage, the ASR task has a total of 9.0M parameters, of which the ecoder part has 5.3M parameters and a total of 12 blocks, the decoder part has 2.9M parameters and a total of 4 blocks, and the CTC part has 0.8M parameters. In the encoder part, the encoder dimension is 128, the output dimension is 164, and there are 4 attention heads. In the decoder part, the hidden units number of position-wise feed-forward is 256, and there are 4 attention heads in total. The weights of ASR loss and CTC loss are both 0.5 during the training process.

In the second stage, the NLU task has a total of 4.4M parameters, of which the BERT module has 4.2M parameters and the CRF module has 0.2M parameters. The BERT module has a 4-layer encoder with an input embedding size of 3155, and the hidden layer unit as well as the ouput size is 256. In the training process, the coefficient of intent loss is 2 and the coefficient of CRF loss is 1.

3.4. Results

The results are shown in Table 1. When using the trained model for inference, we take a standardized approach to text formatting, such as removing extra spaces. The experiments empirically demonstrate the effectiveness of our proposed Two-Stage SLU Model.

Table 1. Results presenting the EM performance of our proposed Two-Stage SLU models.

Model	Parameters(M)	EM(%)
Two-Stage SLU Model	13.38	71.97

4. CONCLUSIONS

In this paper, we propose a two-stage SLU model combining ASR and NLU to better monitor the model performance in each stage and to optimize. Each stage can perform independent training tasks, and in the NLU stage, we combine intent determination and slot filling for joint training. In addition, we use the BPE splitting method with a small word list, which saves a large number of parameters for extending the model depth and achieves better model performance.

In future works, we plan to integrate ASR and NLU tasks into an end-to-end model, which can avoid iterative errors due to multi-stage tasks and save more parameters.

5. REFERENCES

[1] A. Graves, S. Fernández, and F. Gomez, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *International Conference on Machine Learning*, 2006.

[2] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[3] J. Lafferty, A. McCallum, and F.C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *proceedings of icml*, 2002.

[4] S. Kim, A. Gholami, A. Shaw, N. Lee, K. Mangalam, J. Malik, M.W. Mahoney, and K. Keutzer, "Squeezeformer: An efficient transformer for automatic speech recognition," 2022.

[5] P. Tomasello, A. Shrivastava, D. Lazar, P. Chun Hsu, D. Le, A. Sagar, A. M. Elkahky, J. Copet, W.-N. Hsu, Y. Mordechay, R. Algayres, T. Nguyen, E. Dupoux, L. Zettlemoyer, and A. rahman Mohamed, "Stop: A dataset for spoken task oriented semantic parsing," in *arXiv preprint arXiv:2206.00888*, 2022.

[6] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," 2015, p. 5206–5210.

[7] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1715–1725, 2016.